International Journal of Digital Multimedia Broadcasting

Video Analysis, Abstraction, and Retrieval: Techniques and Applications

Guest Editors: Jungong Han, Ling Shao, Peter H. N. de With, and Ling Guan

Video Analysis, Abstraction, and Retrieval: Techniques and Applications

International Journal of Digital Multimedia Broadcasting

Video Analysis, Abstraction, and Retrieval: Techniques and Applications

Guest Editors: Jungong Han, Ling Shao, Peter H. N. de With, and Ling Guan

Contents

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 348914, 2 pages
doi:10.1155/2010/348914

Editorial

Video Analysis, Abstraction, and Retrieval: Techniques and Applications

Jungong Han,[1] Ling Shao,[2] Peter H. N. de With,[1,3] and Ling Guan[4]

[1] *Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands*
[2] *The University of Sheffield, Sheffield S1 3JD, UK*
[3] *CycloMedia Technology, 418 0BB Waardenburg, The Netherlands*
[4] *Ryerson University, Toronto, ON, Canada M5B 1Z2*

Correspondence should be addressed to Jungong Han, jg.han@tue.nl

Received 9 May 2010; Accepted 9 May 2010

The proliferation of TV programs and personal DV cameras has led to an explosion of digital video content, which enriches the personal entertainment of users. However, the rapidly increasing availability of video data has not yet been accompanied by an increase in its accessibility. This is due to the fact that video data are naturally different to traditional forms of data, which can be easily accessed and searched using textual queries. Therefore, the problem of efficiently organizing video, such as TV news and sports, into more compact forms and extracting semantically meaningful information becomes more and more important. In the past ten years, video analysis and retrieval techniques have received significant attention from both industry and academia. The research has gradually converged to three fundamental areas, namely, video *analysis*, video *abstraction*, and video *retrieval*. Video analysis is utilized to extract both the general and the domain-specific visual features, such as colour, texture, shape, human faces, and human motion. Video abstraction generates a representation of visual information, similar to the extraction of keywords or summaries in text-document processing. Basically, video abstraction is associated with key-frame detection, shot clustering, and the extraction of domain knowledge in a video source. The content attributes found in video analysis and abstraction processes are often referred to as metadata. Video retrieval based on the extracted metadata is a fast and interactive tool that allows users to query, search, and browse large video databases. Although a lot of efforts have been devoted into these three areas, both the computational cost and the accuracy of the existing systems are still far from satisfactory. For example, a typical problem is how to optimally use unlabeled data in case that the number of available training samples is very small. Another problem is that in a general setting, the low-level features do not have a direct link to high-level concepts. This raises the question how this semantic gap can be bridged.

This special issue highlights the most recent advances and promising results of the research community working in video analysis, abstraction, and retrieval. After two rounds of careful reviews, nine papers were selected for publication in this special issue. We group the selected papers into three thematic sections corresponding to three research areas mentioned above, though we notice that some papers cover two or even three areas.

The first section deals with *feature extraction* issues in video analysis. The first paper entitled "*ipProjector: designs and techniques for geometry-based interactive applications using a portable*" presents an interactive projection system for virtual studio setup using a single self-contained and portable projection device. The projection allows special effects of a virtual studio to be seen by live audiences in real time. The techniques discussed in this paper, such as colour wheel analysis and motion-based camera calibration, are highly related to and widely used in the video analysis area. The second paper, entitled "*Flexible human behaviour analysis framework for video surveillance applications*," investigates the human motion based on a two-camera setup, which is a key clue for analyzing the surveillance video. The main contribution is the effective combination of trajectory estimation and human-body modelling, facilitating

the semantic analysis of human activities in video sequences. Moreover, an automatic camera-calibration technique is employed to establish the correspondence between the two video channels. By doing so, the system can make decision based on fusing the information from both cameras, thereby resulting in a robust detection.

The second section discusses the *video structuring, video segmenting, and shot detection,* which are all hot topics in the video abstraction area. This section includes four papers. The first paper entitled "*Statistical skimming of feature films*" presents a statistical framework based on Hidden Markov Models (HMMs) for skimming feature films. This work combines the information derived from the story structure with the characterization of the shots in terms of salient features. The structure of the video is captured by HMMs, which model semantic scenes and produce the shot sequence of the final skim. The second paper entitled "*An optimized dynamic scene change detection algorithm for H.264/AVC encoded video sequence*" concentrates on scene change detection for *compressed* video sequences. The scene detector employs a dynamic threshold that adaptively tracks different features of the video sequence, thereby increasing the accuracy in correctly locating true scene changes. The work has been successfully applied within an error-concealment framework for H.264 decoding. The third paper entitled "*Automatic TV broadcast structuring*" proposes a fully automatic system to detect the start and end times of each program in TV broadcasts. The algorithm is based on the detection of repeated sequences, in order to extract *long* useful programs, such as movies, news, TV series, and TV shows. The last paper in this section entitled "*Unsupervised segmentation methods for TV contents*" also deals with the shot boundary detection problem. This paper analyzes both the audio and the video signal of a sequence, rather than relying on the video signal only. The basic system is built upon the hypothesis that it is possible to segment any audiovisual document into homogeneous segments at the adequate scale. The system has been evaluated for two different applications: TV program boundary detection and speaker diarization. For both cases, the system achieves high accuracy.

The third section addresses the *video indexing, browsing, and retrieval*. The first paper entitled "*A video browsing tool for content management in postproduction*" implements an interactive video browsing tool for supporting content management and selection in postproduction. Many visual features, like camera motion, visual activity, face occurrence, global colour similarity, and repeated takes, are extracted and used in this system. The second paper entitled "*Personalized sports video customization using content and context analysis*" focuses on sports videos and addresses three research issues: semantic video annotation, personalized video retrieval, and system adaptation. The system is designed for users to watch refined video segments containing their favourite semantics instead of lengthy sports matches. Moreover, both subjective content preference and objective environment constraints are well balanced so that the optimal visual experience can be brought to the particular viewer. The last paper "*Multimodal indexing of multilingual news video*" deals with the analysis of multilingual news telecasts in India. The basic approach is to index the news stories with relevant keywords discovered in speech and in form of "ticker text" on the visuals. They also create a multilingual keywords-list in English and Indian languages, to enable keyword spotting in different TV channels, both in spoken and visual forms. The evaluation shows that restricting the keyword list to a manageable size results in drastic improvement in indexing performance.

We would like to thank all the authors for sharing with us their nice and innovative work. We also express our sincere gratitude to all the reviewers for their timely and insightful comments on selecting papers. Finally, we are particularly grateful to the Editor-in-Chief, Dr. Fa-long Luo, for his invitation and guidance throughout the entire process. Without his support, we could not make this special issue possible.

Jungong Han
Ling Shao
Peter H. N. de With
Ling Guan

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 352060, 12 pages
doi:10.1155/2010/352060

Research Article

ipProjector: Designs and Techniques for Geometry-Based Interactive Applications Using a Portable Projector

Thitirat Siriborvornratanakul and Masanori Sugimoto

Interaction Technology Laboratory, Department of Electrical Engineering and Information Systems, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Correspondence should be addressed to Thitirat Siriborvornratanakul, thitirat@itl.t.u-tokyo.ac.jp

Received 1 September 2009; Revised 23 November 2009; Accepted 14 December 2009

Academic Editor: Jungong Han

We propose an interactive projection system for a virtual studio setup using a single self-contained and portable projection device. The system is named ipProjector, which stands for Interactive Portable Projector. Projection allows special effects of a virtual studio to be seen by live audiences in real time. The portable device supports 360-degree shooting and projecting angles and is easy to be integrated with an existing studio setup. We focus on two fundamental requirements of the system and their implementations. First, nonintrusive projection is performed to ensure that the special effect projections and the environment analysis (for locating the target actors or objects) can be performed simultaneously in real time. Our approach uses Digital Light Processing technology, color wheel analysis, and nearest-neighbor search algorithm. Second, a paired projector-camera system is geometrically calibrated with two alternative setups. The first uses a motion sensor for real-time geometric calibration, and the second uses a beam splitter for scene-independent geometric calibration. Based on a small-scale laboratory setting, experiments were conducted to evaluate the geometric accuracy of the proposed approaches, and an application was built to demonstrate the proposed ipProjector concept. Techniques of special effect rendering are not concerned in this paper.

1. Introduction

Recently, virtual studio setups have become popular for modern studio productions. Techniques such as studio camera tracking and 3D graphic rendering have been integrated into a conventional studio setup, so professional special effects can be created at lower cost. The main problem with current virtual studio setups is that these special effects are invisible during film recording. Therefore, the special effects cannot be shown to live audiences in live broadcasting. In addition to actors and moderators, it is difficult to respond correctly to invisible content.

One solution based on existing technologies uses a projector. By combining a projector and a camera, any visible or invisible special effect can be directly superimposed onto a studio surface. The visible special effects support live broadcasting, provide realism during virtual studio recording, and allow direct interaction with actors. The invisible special effect, which can only be seen by a specific camera, works very well as a supporting system for virtual studio production. Moderators can embed hidden scripts for verification. In an advanced virtual studio system like that proposed in [1], the hidden information is used to accurately track studio cameras and to render real-time 3D graphics in an arbitrary studio environment.

Using a projector and a camera for live broadcasting with special effects is one type of real-time interactive projection application. By using a projector-camera paired system (a *pro-cam* system), we propose a system that includes hardware designs and software implementations. The system is self-contained, portable, and performs real-time projection and real-time environment analysis simultaneously.

The self-contained portable device allows easy setups in any arbitrary studio environment. Furthermore, the shooting and projecting angles of the camera and projector can be determined freely based on the director's consideration. This is important in actual film shooting where the highest priority is to get the best images from the desirable

angle shot. The simultaneous real-time projection and real-time environment analysis allows advanced virtual studio techniques to be performed while special effects are being continuously projected. Environment analysis in this context refers to the analysis of three targets: the studio scenery and objects, and inside actors. For example, captured images are analyzed to locate an inside actor and his postures so that special effects can be created and projected in response to his actions. Note that the concept proposed in this paper could also be applied to other portable interactive projection systems.

This paper focuses on two problems that are important in implementing real-time interactive projection applications. The first problem is projecting real-time special effects so that they do not interfere with normal camera capturing. The second problem is calibrating a pro-cam system precisely to ensure correct transformations between a projector and a camera in a portable setup, which implies use in an unknown environment.

The first problem is also called the nonintrusive projection problem. In normal projection, projected images can be captured by a camera; discriminating between projected contents and real contents in captured images is still difficult. Nonintrusive projection guarantees that projected special effects will be visible to humans and normal studio cameras but invisible to the calibrated camera. This is necessary to avoid projective interference (as seen by the camera) that may lead to an incorrect environment analysis. The technique proposed in this paper involves Digital Light Processing (DLP) projector, RGB color space sampling and nearest-neighbor search algorithm.

With respect to the second problem, there are two related calibrations that are often concerned by researchers using a pro-cam system: geometric and radiometric calibrations. Both calibrations require completely different calibration approaches. In this paper, we focus on the first calibration to achieve a geometrically calibrated pro-cam system. With the calibrated system, we are able to project special effects onto desired locations. A motion-sensor-based calibration technique and a beam-splitter-based calibration technique are investigated in the following sections.

In this paper, we show two complete designs of the self-contained portable projection device. Based on the developed designs, solutions of the nonintrusive projection problem and the geometric calibration problem are proposed. Experiments were conducted to evaluate the accuracy of the proposed approaches in a small-scale laboratory setting; a white board and color magnets were used to represent the studio projection surface and the studio target objects, respectively. Techniques of special effect rendering are not concerned in this paper; therefore, the projected special effects are the simple "+" pattern. The proposed devices and approaches can be applied in a full-scale virtual studio environment by implementing additional target detection algorithms for the desired studio target objects or actors.

The rest of this paper is laid out as follows. Section 2 explains recent advances in interactive projector systems and then discusses related research in the two problem areas as mentioned earlier. Section 3 shows the techniques using an off-the-shelf DLP projector for nonintrusive projection; the initial setup for pro-cam synchronization, necessary camera settings and detailed analysis results regarding our DLP projector model are written in the section. Section 4 presents two alternative system configurations (in combination with the initial setup shown in Section 3) for real-time pro-cam geometric calibration based on perspective transformation model. On one hand, Section 4.1 applies an additional motion sensor for calibration on a planar or slanted surface. On the other hand, Section 4.2 uses a beam splitter and introduces a complete portable design consisting of a projector, camera and beam splitter; the precise geometric calibration is achieved on both planar and nonplanar surface and suitability of the design regarding the nonintrusive projection is observed in the section. In Section 5, performances of the proposed nonintrusive projection and geometric calibration approaches are experimentally confirmed and an application is built to demonstrate the overall concept of the ipProjector. Finally, Section 6 concludes this article along with a plan for future works.

2. Related Work

2.1. Interactive Projector Systems. Cotting and Gross [2] introduced an environment-aware display system that automatically avoids projections onto nonsurface objects. Their system performs real-time interactions, but is limited to fixed projectors and fixed cameras mounted on a ceiling. In addition, surfaces are restricted to flat table surfaces whose distance to the ceiling is unchanged. In Cao et al. [3, 4], interactive mobile projector systems were developed. While their projection device is self-contained, it requires a camera mounted separately in a workspace for 3D positioning purpose. In CoGAME [5], images projected from a handheld projector control the movement of a robot. A camera with an IR filter sees only three IR LEDs attached to the robot, and so other visual information about the environment is disregarded completely. The latest SixthSense prototype [6] is a mobile pro-cam device that offers meaningful interactions with different objects found in the real world. However, the projector and camera are not calibrated in their system and environment analysis is performed even though there is projective interference. Consequently, geometric accuracy is limited and color markers are used to help locate target objects like fingertips and desired projection areas. Unlike these systems, our proposed interactive system is truly self-contained and geometrically calibrated. It can perform real-time projection and real-time environment analysis simultaneously without any projective interference.

2.2. Nonintrusive Projection. This topic is a subset of the embedded imperceptible-pattern projection problem. Prototypes of an infrared projector were proposed in [7, 8] to project infrared and visible light simultaneously. An infrared pattern is fixed by using an internal mask inside a projector in [7], but is variable in [8]. Unfortunately, the work of Lee et al. [8] requires many internal changes inside a DLP projector that can be accomplished only by

a commercial manufacturer. While an infrared projector is under investigation, there are existing solutions proposed for this problem. For the office of the future [9], structured light can be embedded into a DLP projector by making significant changes to the projection hardware. However, this implementation is impossible unless it is incorporated into the design of the projector or full access to the projection hardware is available. In [1, 10, 11], a code image is projected at high speed with its neutralized image, which integrates the coded patterns invisibly due to limitations of the human visual system. According to these papers, projecting and capturing at 120 Hz can guarantee a hidden code. Commonly available projectors usually perform projections at a maximum rate of 87 Hz.

For this paper, we apply an approach based on the DLP characteristics. Using the camera classification approach proposed in [12] and the nearest-neighbor search algorithm, we are able to perform nonintrusive projection using an off-the-shelf DLP projector.

2.3. Real-Time Pro-Cam Geometric Calibration. When a projector and a camera are rigidly fixed to each other, some have assumed that the geometric registration between them is roughly constant [13]. However, as the angle of the projector moves from the perpendicular or as a surface becomes nonplanar, this approach will no longer guarantee good geometric registration. Projecting a known pattern onto a surface is a classical approach to solve this problem that gives precise calibrations for both planar surfaces [14–16] and irregular surfaces [17–19]. However, the computational cost is high for complex surfaces, and patterns must be re-projected when a component of the system (e.g., a projector, camera or surface) moves. Similar approach is applied in the catadioptric projectors [20] whose projected light is refracted/reflected by refractors/reflectors; geometric registration between the two devices is obtained by projecting a series of known patterns and allowing the camera to sense them. A real-time approach that does not interrupt normal projection was proposed in [21] by attaching four laser-pens to a pro-cam system. Although detecting bright laser points sounds easier than detecting points projected by a projector, locating small laser points in a messy camera image is still difficult. In [22], Johnson and Fuchs proposed a real-time approach that does not interrupt normal projection, requires no fixed marker and can be applied to a complex surface. By matching feature points found in the projected image and the predicted captured image, the pose of the projector is tracked and the calibration is achieved in real time. However, the camera is stationary and separated from the projector in their system.

This paper involves two calibration approaches that can be implemented as a single self-contained device. The first approach uses one additional motion sensor, and geometric calibration is achieved in real time on a planar or slanted surface. The second approach uses a beam splitter to coaxialize the projector and camera. Geometric calibration is independent of the scene, so both planar and irregular surfaces can be used as projection surfaces.

3. Nonintrusive Projection

As mentioned in the introduction, it is important that a real-time environment analysis has no interference from any projected contents. If it does, the system might consider the projected content as a real target and generate special effects in response to that false detection.

Recently, internal characteristics of a DLP projector have received lot of attentions from research communities. On one hand, in [23, 24], the dithered illumination pattern corresponding to the DMD chip (which operates at 10 000 Hz) is observed and utilized using a very high speed camera (whose maximum speed is 3000 fps). On the other hand, characteristics of the color wheels (which rotate at 120 Hz) can be investigated and used by a camera with slower capturing speed. Our nonintrusive projection is based on the latter one; characteristics of the color wheels are utilized here for nonintrusive projection purpose. The proposed approach has three main advantages: it requires no internal change to the projector or the camera, it can be applied to any off-the-shelf DLP projector, and it supports embedded variable light patterns in the future without further hardware modifications.

In the following sections, we explain in detail how to analyze the characteristics of the color wheels inside the DLP projector and how to use these characteristics for nonintrusive projection. Note that a beam splitter (as described in Section 4.2) has not yet been applied in these sections.

3.1. DLP Projector Analysis. Because each DLP projector model owns unique characteristics of the inside color wheels, DLP projector analysis has to be performed before using an unknown DLP projector model for the proposed nonintrusive projection approach. To understand the overall characteristics of the color wheels without full access to a DLP chip and its controller, we applied the camera-based classification method proposed in [12]. In this section, we briefly explain the classification steps and show the classification results of our DLP projector.

First, this classification method requires a camera with an external trigger feature to synchronize it with a DLP projector. Synchronization between the projector and the camera is performed by tapping the vertical sync signal (5 V, 60 Hz) from the computer to the projector. By using the tapped signal as a trigger, our camera remains synchronized to the projector. In addition, the shutter of the camera must be set to open for a very short period in order to sense the fast characteristics of the color wheels. In our setup, the camera is set to expose for only 0.55 ms.

The following devices were used for our pro-cam synchronization: a HP MP2225 DLP projector (XGA 1024 $\times$ 768 projection resolution) with a D-sub 15 pin connector, a Dragonfly Express camera (VGA 640 $\times$ 480 captured resolution) connected through a FireWire 800 (IEEE1394B) port, and an ELECOM VSP-A2 VGA splitter. The camera is equipped with a Tamron 13VM308AS lens. The synchronization setup is illustrated in Figure 1.

Second, we analyzed the overall sequences of the color wheels inside our DLP projector by projecting single-color

images (corresponding to the colors of each available color wheel of the projector) at maximum intensity through all possible starting exposure times. Figure 2 was created by allowing the synchronized camera to sense these projected colors with different starting exposure times.

Third, we analyzed detailed mirror flip sequences for all 256 values in the selected color channel (i.e., red, green or blue) within a narrow starting exposure period. From Figure 2, the selected color channel is the red channel, which is the first channel appearing in the sequences. Mirror flip sequences were then obtained by projecting uniform red images with intensity values ranging from 0 to 255, and with the starting exposure times ranging from 0 to 2 ms. Figure 3 was created by allowing the camera to sense these red projections. A starting exposure time of 1.4 ms was finally chosen because it provides the best distributed red ramps, as shown in Figure 3.

Following the explained steps, we are able to synchronize the camera with the projector at the appropriate starting exposure time. For our selected starting exposure time, the camera can only see the red light of the DLP projector (correspondences between projected red intensities and red intensities seen by the camera are shown in Figure 3). If red intensities of the projected special effects are similar to those of the projected background color (as seen by the camera), the system cannot differentiate the projected special effects from the background in captured images. Hence, further environment analysis is not interfered by the real-time special effect projection.

3.2. Environment Illumination. To perform the DLP analysis and use it for nonintrusive projection (as mentioned in Section 3.1), the shutter of the camera is set to expose for only 0.55 ms, which is too short for the camera to sense the environment properly (as shown in Figure 4(b)) unless there is light emitting from the projector (as shown in Figure 4(c)). In [12], the 256 red intensities in the selected timeslot were classified into three sets: *white*, *black* and *grey*. *White* refers to colors whose projection fully turns mirrors inside the DLP projector and transmits lights toward a surface. *Black* refers to colors whose projection does not flip the mirrors and transmits no light toward a surface. *Grey* refers to unreliable states between *white* and *black*.

For environment analysis purpose, we need to illuminate the environment while projecting nonintrusive special effects. Thus, only colors whose red value contained in the *white* set should be projected. Figure 4(c) depicts an image seen by the camera when Figure 4(d), whose red intensity is maximal for all pixels, was projected. Note that the intensity of Figure 4(b) was enhanced here to allow the environment to be seen.

3.3. Color Space Sampling and Color Conversion. The method of projecting only colors whose red intensity contained in the *white* set (as mentioned in Section 3.2) works only for a DLP projector model that does not have interdependent color channels in mirror flipping. Instead of depending on the red channel and risking effects from the other color channels,

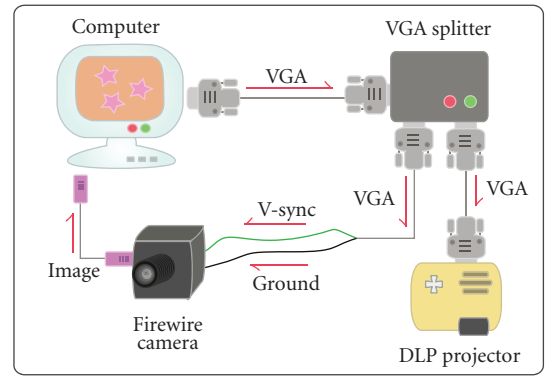


Figure 1: Setup for projector-camera synchronization.

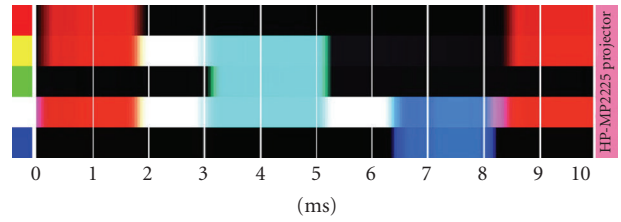


Figure 2: Color-wheel sequences of the HP MP2225 as seen by the synchronized camera with starting exposure times ranging from 0 to 10 ms.

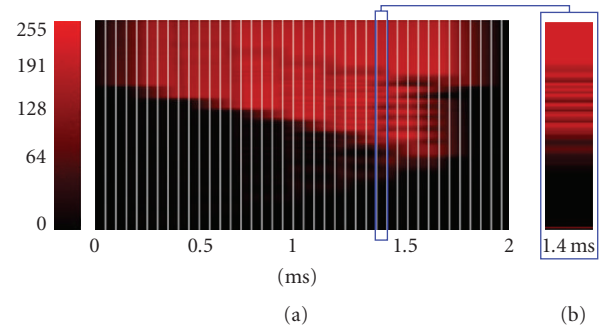


Figure 3: (a) Detailed mirror flip sequences for all 256 values in the red channel with the starting exposure times ranging from 0 to 2 ms. (b) Mirror flip sequences at the selected starting exposure time.

we propose a DLP-model-independent classification that involves all color channels (i.e., red, green and blue) of projected images. Each color in the RGB color space was projected onto the white surface and sensed by the camera. Using the camera classification method, these projected colors were classified into the three sets previously discussed. Based on this proposed classification, we are also able to determine whether our DLP projector has interdependent color channels in mirror flipping.

From the entire RGB color space containing 256 × 256 × 256 colors, we selected only 11 × 11 × 11 colors and obtained 115 colors belonging to the *white* set. Because the amount of light passing through the camera is decreased by

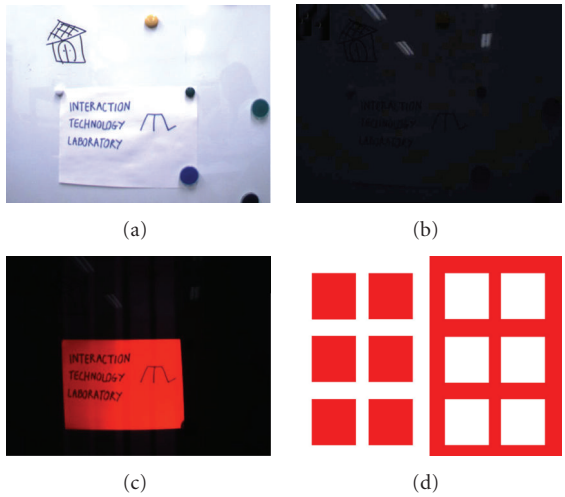


Figure 4: (a) is an environment seen by a camera with normal shutter settings. (b) is the environment (a) as seen by the synchronized camera. (c) is the environment (a) seen by the synchronized camera when image (d) is being projected from the DLP projector.

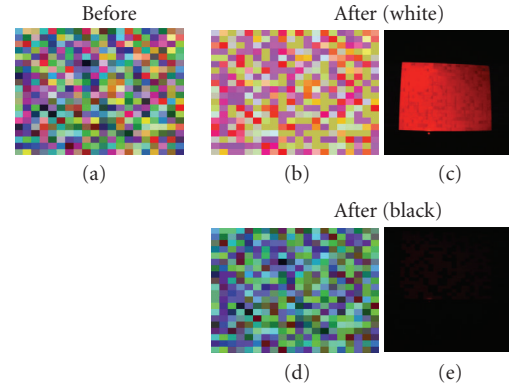


Figure 5: Color conversion using BBF algorithm. (a) is the randomly generated 400 colors before conversion. (b) and (d) are the (a) image after converted to the *white* and *black* set, respectively. (c) and (e) are the blank surface seen by the synchronized camera when image (b) and (d) are being projected, respectively.

the beam splitter setup (as detailed in Section 4.2), we set a high threshold value for the *white* set. Thus, there are few colors categorized into this set. In detail, we projected single-color images (corresponding to the colors in the selected color space) and allowed the camera to sense the projected colors. For one projected color, values of all pixels inside the projection area (as seen by the camera) are averaged before that color is classified into the appropriate set. Because of this, we can ensure that the different resolutions between the projector and camera will not affect our processes of color space sampling and color classification.

For converting an arbitrary color into the most similar color in the *white* set, we used an approximate nearest-neighbor search algorithm called Best Bin First (BBF) [25]. Using this, we can convert an arbitrary color into the *white* set, and the working environment is then always illuminated. In Figure 5, we randomly generated 400 colors and converted them into the *white* set using the BBF algorithm. The converted image was then projected and sensed by the synchronized camera. Conversion of the same image to the *black* set is also shown for comparison. From images captured by the camera shown in Figure 5, it is clear that our classification and conversion approach can perform efficiently, and the surface remains well illuminated when projecting colors in the *white* set.

4. Real-Time Pro-Cam Geometric Calibration

Geometric calibration is the first problem encountered by most pro-cam systems. Geometric mapping between camera and projector coordinates is necessary to find corresponding positions between the two coordinate systems, and to project images back to desired locations on an actual surface. The three objects affecting this calibration are the projector, the camera and the surface. Any relative movement between any pair of the three objects causes changes in the pro-cam geometric mapping. By attaching the camera rigidly to the projector, there is no relative movement between the projector and the camera but this cannot prevent relative movements with the surface. On a planar surface, when the angle of a projector moves away from perpendicular, the geometric mapping changes. Nonoverlapping fields of view of the projector and the camera may make geometric mapping impossible in some positions. On an irregular surface, there is an additional serious problem as 3D shapes of the surface create parallax effects between projector and camera coordinates. This problem is difficult to recover from unless the 3D geometries of the surface are known.

In this paper, geometric mapping between camera and projector coordinates is computed by perspective transformation [26] whose computation effort is lighter than Euclidean calculation does. Based on the fact that all points seen by the camera lay on some unknown plane, the perspective transformation between the two coordinates can be established by a 3×3 homography matrix. Suppose that (X, Y) is a pixel in projector coordinates whose corresponding pixel in camera coordinates is (x, y), the perspective transformation from (x, y) to (X, Y) can be expressed with eight degrees of freedom in

$$(X, Y) = \left(\frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + h_9}, \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + h_9} \right), \quad (1)$$

where $\vec{h} = (h_1 \cdots h_9)^T$ is constrained by condition $|\vec{h}| = 1$. The same transformation as written in (1) can be expressed in homogeneous coordinates as

$$\begin{pmatrix} Xw \\ Yw \\ w \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (2)$$

$\vec{h}$ can be computed from four corresponding pixels between the two coordinates (four correspondences ensure that no

three points is collinear). When there are more than four corresponding pixels found between the two coordinates ($\delta > 4$ in (3)), the RANSAC method is applied for estimating the values of $\vec{h}$ in the following equation:

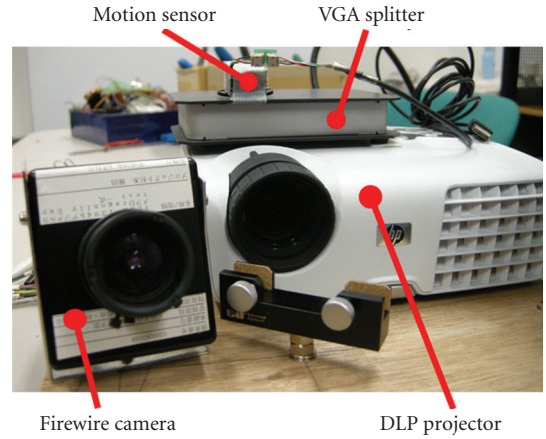
$$\begin{pmatrix} X_1 w & X_2 w & \cdots & X_\delta w \\ Y_1 w & Y_2 w & \cdots & Y_\delta w \\ w & w & \cdots & w \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 & \cdots & x_\delta \\ y_1 & y_2 & \cdots & y_\delta \\ 1 & 1 & \cdots & 1 \end{pmatrix}. \tag{3}$$

The purpose of the following sections (i.e., Sections 4.1 and 4.2) is to find at least four corresponding pixels between camera and projector coordinates and use them to compute the updated $\vec{h}$ values in real time. After that, geometric mapping from any (x, y) to (X, Y) and vice versa is achieved using $\vec{h}$ and $[\vec{h}]^{-1}$, respectively. Sections 4.1 and 4.2 explain two alternative approaches for finding a set of corresponding pixels between the two coordinates based on two different setups. Both approaches are implemented as a single self-contained device and support portable use. The first approach uses an additional motion sensor and can perform real-time geometric calibration on a planar or slanted surface. The second approach uses a beam splitter so that the geometric mapping is independent of the surface.

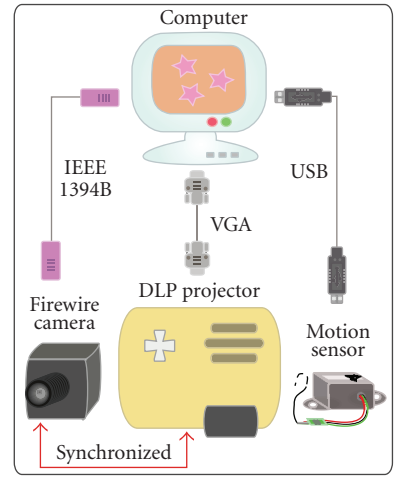
By the way, considering the frame buffer architecture of the projector, there is one frame delay time before an image sent to the projector will be projected out. Any movement regarding the projector, camera or surface during this delay time causes geometric errors between the calculated geometries and the actual geometries (of projected images) appearing on the surface. In our system, frequency of the projection cycle is 60 Hz (as mentioned in Section 3.1) which equals to the maximum 16.67 ms projection delay time. For the application proposed in Section 5, this delay time is relatively short compared to other processing times. Therefore, we decided to neglect the effect of this delay from our calculation. However, for an interactive system that requires the millisecond precision in projection, additional techniques such as motion estimation should be applied.

4.1. Motion-Sensor-Based Approach. This section describes the approach using a motion sensor to find a set of corresponding pixels between camera and projector coordinates on a planar or slanted surface. Two tilt sensors fixed to a projector were first proposed in [16]. Acquiring the tilt angles from both sensors in real time allows the correct estimation of the world's horizontal and vertical directions without using markers. Dao et al. [27] extended the sensor-based idea to an accelerometer combined with a digital compass. Their system measures the inclined angle of a projector directly in both vertical and horizontal axes, and then creates an interactive application by using real-time keystone correction.

A sensor eliminates the need for fiducial markers but still allows a single self-contained device. Our configuration for this calibration approach is shown in Figure 6; a projector, camera and motion sensor are fixed together on a wooden base so that their relative positions and orientations cannot be changed. The purpose is to obtain pairs of corresponding (X, Y) and (x, y) in real time. Unlike previous researches that use the sensors to compute the rotation matrix of a projector or to correct the keystone distortion, we directly correlate the sensor values with camera coordinates in order to find updated geometric mapping between camera and projector coordinates in real time. Keystone correction is not concerned in our system.



(a)



(b)

Figure 6: Projector-camera-sensor configuration.

In this paper, a NEC/TOKIN MDP-A3U9S 3D motion sensor is used with a data update rate of 125 Hz. The relative pitch and roll angles are calculated from three acceleration values, *AccX*, *AccY* and *AccZ* (acceleration values along X, Y, and Z axis, resp.), read from the accelerometer embedded in our motion sensor. Setting the reference angles is simple—the projector is moved until the images appear rectangular on a surface and then a key or button is pressed. The reference can be reset whenever a user prefers or feels significant geometric errors (between the calculated geometries and the actual geometries appearing on the surface) in the calibration. Five consecutive samples of the 3D acceleration

values acquired from the sensor are averaged in real time before being used to compute the relative pitch and roll angles. Averaging adds a delay but is recommended for a smoother calibration. The calculation of the relative pitch and roll angles can be summarized as follows:

$$\begin{aligned}\text{pitch}_{\text{rel}} &= \arccos\left(\frac{AccX_{\text{avg}}}{\sqrt{{AccX_{\text{avg}}}^2 + {AccY_{\text{avg}}}^2 + {AccZ_{\text{avg}}}^2}}\right) \\ &\quad - \text{pitch}_{\text{ref}}, \\ \text{roll}_{\text{rel}} &= \arctan\left(\frac{AccY_{\text{avg}}}{AccZ_{\text{avg}}}\right) - \text{roll}_{\text{ref}},\end{aligned} \tag{4}$$

where $\text{pitch}_{\text{ref}}$ and roll_{ref} refer to the reference pitch and roll angles. $AccX_{\text{avg}}$, $AccY_{\text{avg}}$ and $AccZ_{\text{avg}}$ are the average of five consecutive $AccX$, $AccY$ and $AccZ$ values read from the sensor, respectively.

This approach requires offline calibration. However, the calibration data are compatible with the system if there is no change in the relative positions or orientations of the three devices. Suppose that the offline calibration is achieved using N sample images captured from the camera, and all sample images share a set of n points to be calibrated. A set of calibration data provided by one sample image can be written as $(p, r, x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$. Let p and r refer to the relative pitch and roll angles, and (x_i, y_i) represents the 2D camera coordinate of an ith observed point in the sample image. For N sample images captured from different angles and orientations, we have

$$\begin{aligned}A &= \begin{bmatrix} p_{(1)} & r_{(1)} & 1 \\ p_{(2)} & r_{(2)} & 1 \\ \vdots & \vdots & \vdots \\ p_{(N)} & r_{(N)} & 1 \end{bmatrix}, \\ B &= \begin{bmatrix} x_{1(1)} & y_{1(1)} & x_{2(1)} & y_{2(1)} & \cdots & x_{n(1)} & y_{n(1)} \\ x_{1(2)} & y_{1(2)} & x_{2(2)} & y_{2(2)} & \cdots & x_{n(2)} & y_{n(2)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{1(N)} & y_{1(N)} & x_{2(N)} & y_{2(N)} & \cdots & x_{n(N)} & y_{n(N)} \end{bmatrix}.\end{aligned} \tag{5}$$

The offline calibration is finished when an adjustment matrix (β) is obtained by using linear least squares to solve

$$\left(A^T A\right)\beta = A^T B. \tag{6}$$

At any time τ during the online calibration, (x_i, y_i) is updated from the relative pitch and roll angles (calculated as explained earlier) by

$$\left\lfloor x_{1(\tau)}, y_{1(\tau)}, x_{2(\tau)}, y_{2(\tau)}, \ldots, x_{n(\tau)}, y_{n(\tau)} \right\rfloor = \left\lfloor p_{(\tau)}, r_{(\tau)}, 1 \right\rfloor \beta. \tag{7}$$

Following the above explanation, even though the position and orientation of the pro-cam system (relative to the surface) are not known, the system is able to obtain camera coordinates $(x_{i(\tau)}, y_{i(\tau)})$ of the n pre-defined points (whose projector coordinates $(X_{i(\tau)}, Y_{i(\tau)})$ are known). Using the corresponding $(x_{i(\tau)}, y_{i(\tau)})$ and $(X_{i(\tau)}, Y_{i(\tau)})$ to compute $\vec{h}$ as shown in (3) (when $\delta = n$), real-time geometric mapping between camera and projector coordinates is achieved.

4.2. Beam-Splitter-Based Approach. A beam splitter is an optical device that reflects half of the incoming light and transmits the other half. There are few researches concerning pro-cam systems using a beam splitter. In [7], two cube beam splitters are used to construct an IR projector and a multi-band camera. However, the beam splitters are used for internal hardware architecture purposes not calibration purposes. Fujii et al. [28] briefly described the idea of scene-independent geometric calibration using a plate beam splitter attached to an off-the-shelf projector. The calibration technique proposed in this section was inspired by this work. Both their research and ours operate in the visible light spectrum; however, the camera settings are completely different. In their research, a camera uses a normal shutter speed and works independently to a projector. In our research, as described in Section 3, the camera is accurately synchronized with the DLP projector and its shutter is opened for only 0.55 ms. In this section, we investigate a concrete portable design and suitability of the beam splitter regarding our nonintrusive projection approach. Many related factors are introduced and observed. The beam splitter used in our configuration is a TechSpec plate beam splitter 48904-J, whose dimensions are 75×75 mm.

Using a beam splitter to coaxialize the two devices ensures that any surface visible to a camera can also be projected upon. The shapes of the surface do not affect the geometric mapping or cause parallax between the two coordinates. This means that geometric mapping ($\vec{h}$) needs to be computed only once using (3); recomputation is not necessary if there is no change in the relative positions or orientations among the projector, camera and beam splitter. The coaxial concept is illustrated in Figure 7(a) and our beam splitter configuration is shown in Figure 7(d). The distance from the front edge of the wooden base to the projector lens is 13 cm. In addition to the design proposed in [28], we added a curtain made from a light absorbing black-out material to achieve the more practical portable design. This curtain not only eliminates reflections of the environment at the left side of the projector (as shown in Figure 7(c)) but also prevents the projector's reflected light from interfering with the environment (as shown in Figure 7(b)). Note that the camera was not in the high shutter speed mode when capturing Figure 7(c).

As explained in Section 3, the nonintrusive projection shortens the exposure time of the camera significantly. Furthermore, the beam splitter setup allows only half of the projected light to be transmitted to the surface. As a result, the amount of light passing through the camera lens in this configuration is quite limited and may result in inaccurate environment analyses. Therefore, we conducted three experiments using the camera setting explained in Section 3 and investigated factors related to the practicality of

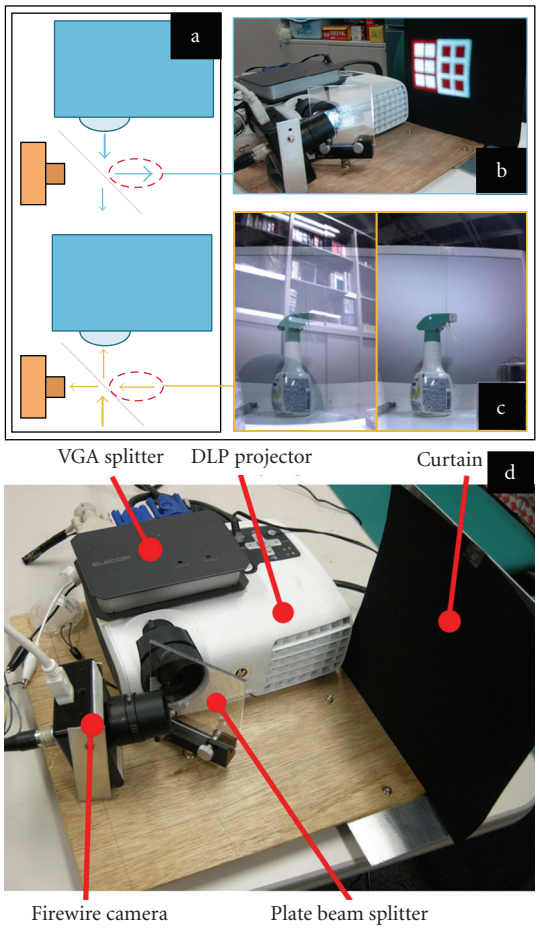


FIGURE 7: (a) The concept of pro-cam coaxialization using a beam splitter. (b) The use of a curtain to block projector's reflected light. (c) Captured image without and with the curtain (left and right). (d) Our beam splitter configuration.

this beam splitter configuration. Only red value is concerned in our experiments because the projector projects the red light at the selected timeslot.

In the first experiment, we investigated how much the beam splitter reduces the amount of light seen by the camera. The three experimental setups are (1) no beam splitter, and camera directly sees a surface, (2) a beam splitter in front of the projector lens, but the camera still sees a surface directly, and (3) our configuration as shown in Figure 7(d). With a distance of 50 cm from the wooden base to the surface, we projected uniform red images with intensity values ranging from 0 to 255 onto a whiteboard, and allowed the camera to sense these red projections. Figure 8 shows the red intensities seen by the camera in the three setups. Comparing the first and second setup, the red intensities seen by the camera are reduced by 53.78% when the beam splitter is placed in front of the projector lens. Comparing the second and third setup, the red intensities are reduced by 38.39% when the camera sees the surface through the beam splitter. In total, by comparing the first and third setup, our configuration reduces the red intensities seen by the camera to about 71.52% of its original intensities compared with the conventional pro-cam setup. This means that amount of light seen by the camera will be quite limited with this configuration. Therefore, a projector model whose brightness is not strong enough might be difficult to be used in our proposed system.

We investigated the distance in the second experiment. Because the exposure time of the camera is very short, the farther away the surface is, the less light the camera will see. If an environment is insufficiently illuminated, it will be difficult to use any image processing techniques. In this experiment, we constantly projected a red image (whose red intensity value was 255) onto a whiteboard located at different distances. Figure 9 shows the experimental results. At a distance of 20 cm, the environment was illuminated with a very bright red light. At a distance of 130 cm, the environment was too dark for the camera to see properly. Note that the brightness of our DLP projector is 1400 ANSI lumens and the distance written in this context refers to a distance from the surface to the front edge of the wooden base.

Finally, we performed an experiment to measure the maximum distance over which our configuration can perform environment analyses correctly. We projected a pure red image (with red intensity values equal to 255) at different distances from 20 to 130 cm. The surface was a whiteboard containing five color magnets inside the projection area. We applied 2D Gabor filters [29] to each captured image to evaluate the accuracy of the object detection at each distance. The Gabor filters failed to detect all objects at a distance of 130 cm. Because the projected color used here is the brightest color possible to be seen by the camera, we conclude that the maximum distance at which this beam splitter configuration can be used in the nonintrusive projection mode is 120 cm.

5. Experimental Results and Evaluations

In this section, we discuss the experiments conducted to evaluate the proposed approaches. All experiments were performed using a Dell Inspiron 1150 Mobile Intel Pentium 4 laptop with a processor running at 2.80 GHz. The projector's focus was adjusted manually in all experiments.

For the nonintrusive projection, we evaluated whether our sampled colors (from the *white* set) can illuminate an environment enough for an environment analysis. At a specific distance, we projected each color from the *white* set onto the whiteboard holding five color magnets, and allowed the camera to sense the whiteboard. After applying 2D Gabor filters to the captured images, the projected colors causing incorrect detection were counted. Table 1 shows the experimental results. Most of the misdetection was caused by noises added in the dark environment. Overall, five magnets were detected satisfactorily until the distance reached 110 cm.

To determine the accuracy of the pro-cam geometric calibration on planar surfaces, we measured the geometric error of the three approaches compared with a ground truth. Apart from the two proposed approaches (see Section 4), we added the single calibrated approach for comparison purposes; this approach lets the camera sense a surface

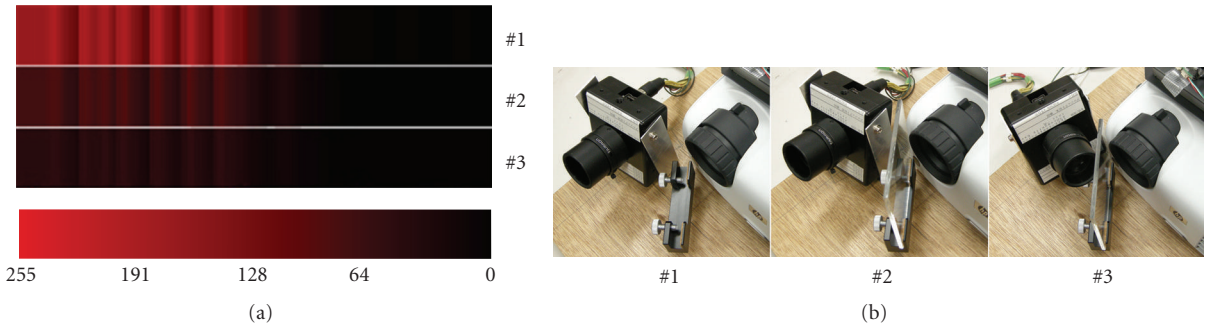


Figure 8: Red intensities seen by the camera in three different setups.

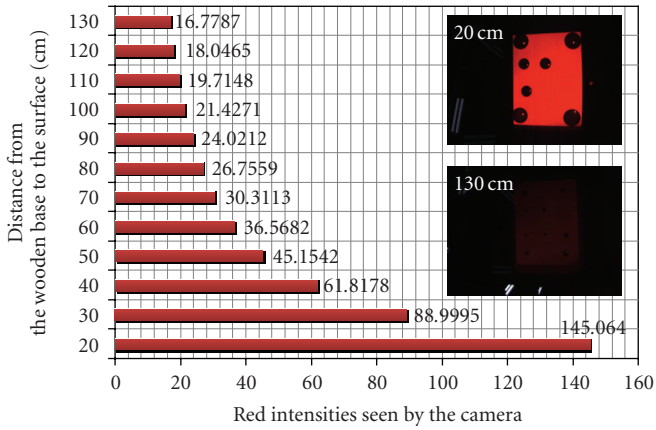


Figure 9: Red intensities seen by the camera at different distances to the surface using the beam splitter setup.

Table 1: Accuracy evaluation of the nonintrusive projection regarding the distance to the surface.

Distance (cm)	Number of misdetected colors	Distance (cm)	Number of misdetected colors
20	0	80	19
30	0	90	29
40	1	100	39
50	1	110	48
60	4	120	80
70	19	130	111

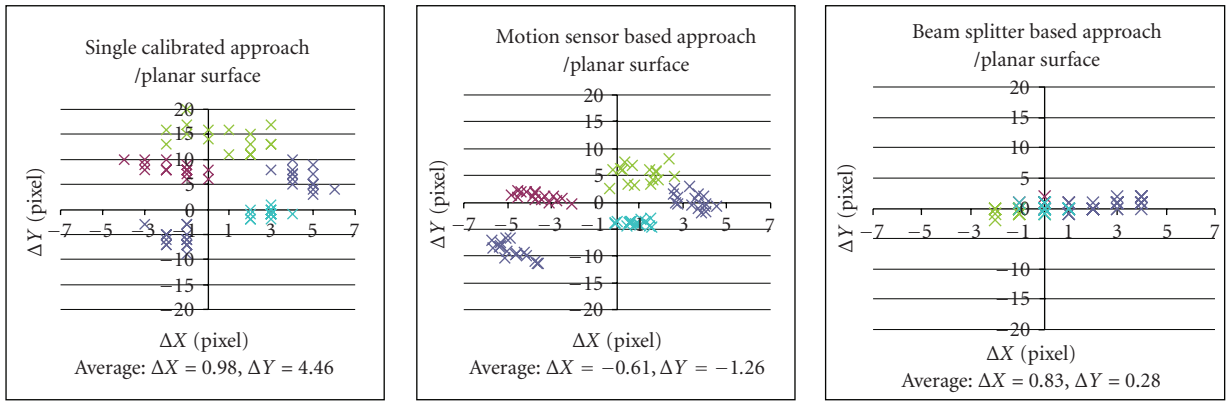
Note: There are 115 colors in the white set.

directly and assumes a static geometric mapping between projector and camera coordinates. The motion sensor offline calibration was performed using 10 sample images containing 16 calibrated points ($N = 10$ and $n = 16$ according to Section 4.1). In the experiments, camera coordinates generated by the three approaches were compared with actual camera coordinates. Five experiments were conducted with different orientations of the projector for each approach and each experiment was performed using 16 tested points (the number of tested point written here is not the n value used in the motion sensor offline calibration). Except for the beam splitter approach, whose device setup is different, the experiments were conducted simultaneously. Note that the resolution of the camera coordinates was 640×480 pixels.

According to the experimental results shown in Figure 10, the beam-splitter-based approach is the most accurate and provides the narrowest range of geometric errors in both axes. For the other two approaches, error values derived from the same experiment cluster together. The clustering locations in the five experiments are similar in both approaches because they were generated from the same projector orientations. However, the distribution of error values in the single calibrated approach is wider than that for the motion-sensor-based approach. Besides, the accuracy of the two proposed calibration approaches does not fall over time. The accuracy of the motion-sensor-based approach may decrease when the current projector orientation differs significantly from the reference orientation.

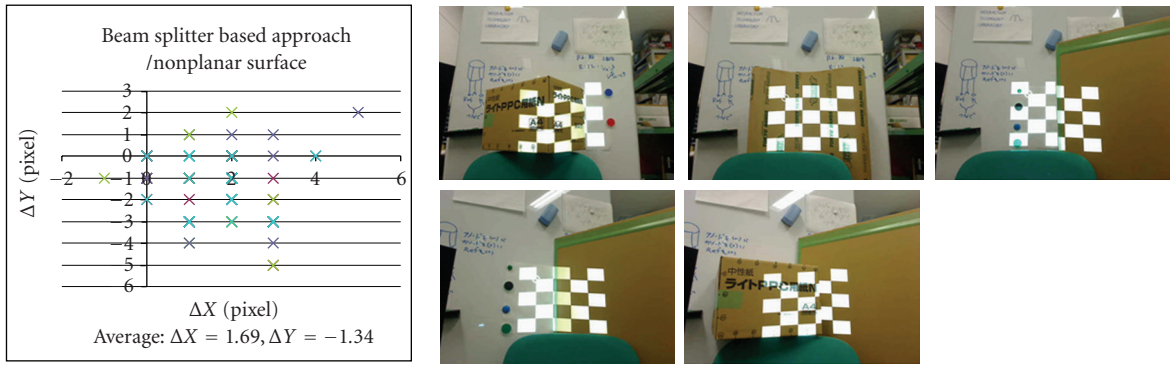
In addition to the beam-splitter-based calibration approach, we conducted the same experiment with 16 tested points on five nonplanar surfaces that cannot be calibrated using the other two calibration approaches. Figure 11 shows the experimental results and the experimental surfaces captured by a separate camera. Using the beam splitter setup, geometric errors are small even for these difficult surfaces.

Comparing the motion sensor and the beam splitter setups, the latter provides more precise geometric calibration and allows calibration on nonplanar surfaces. This is good for a portable system where the geometry of a surface is



(a) (b) (c)

Figure 10: Geometric error (camera coordinates) of the single calibrated approach, the motion-sensor-based approach and the beam-splitter-based approach on planar surfaces.



(a) (b)

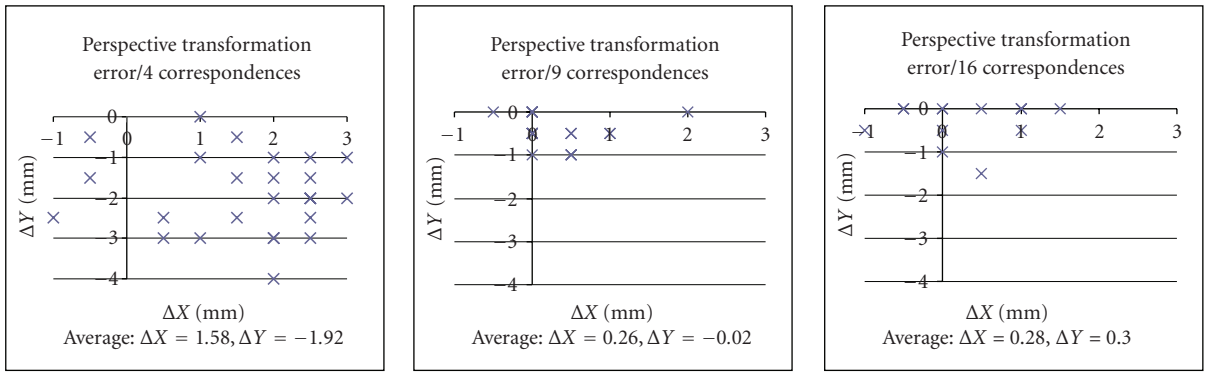
Figure 11: Geometric error (camera coordinates) of the beam-splitter-based approach on nonplanar surfaces.

not known. However, considering the quality of captured images, direct capturing can sense the environment more effectively than indirect capturing. Therefore, by using the motion sensor setup, image processing can be performed more efficiently.

We measured the geometric error of the perspective transformation by reprojecting known points onto an actual surface. Using a planar surface, three experiments were conducted using the geometric mapping created from 4, 9, and 16 correspondences, respectively. During each experiment, a chessboard pattern with 25 inner corners was projected. We located each inner corner in camera coordinates, applied the perspective transformation to map those camera coordinates to projector coordinates using (3), and drew the transformed projector coordinates back on the projected images. In this way, we can measure the geometric error between the actual projector coordinates and the transformed projector coordinates of each tested point on the actual surface. Figure 12 shows the experimental results. While experimenting, we moved neither the surface nor the devices, and the projection area appearing on the surface had the dimensions 294×222 mm.

Finally, we built a small-scale application using the beam splitter setup calibrated with 16 correspondences. The purpose was to demonstrate the entire concept of our interactive portable projector. Any objects are detected using 2D Gabor filters; nonintrusive projection is used to draw text or special effects upon the detected objects or other areas of the surface. Large-scale programs using the same procedure can be built for live broadcasting of virtual studio productions. Instead of the Gabor detector, human detector, gesture recognition or motion analysis algorithms might be used to detect an actor and interpret his actions accurately in real time; robustness of the system regarding moving targets will depend on the selected detection algorithm. The rendering technique might be used to virtually paint the studio, and animation techniques could be utilized to draw attractive characters interacting with the detected actor.

Figure 13 shows the application in action, including snapshots of the surface (taken with a separate camera)



(a) (b) (c)

Figure 12: Geometric error of perspective transformation on an actual surface (world coordinates) using the number of calibrated correspondences of 4, 9, and 16.

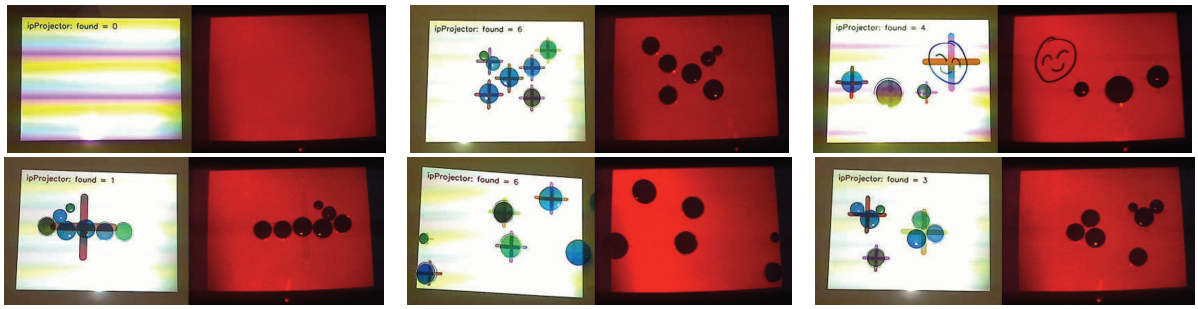


Figure 13: Demonstration of the ipProjector concept. Left images are snapshots of the surface taken by a separated camera. Right images are images captured by the system camera.

and images captured by the system camera. The number of detected objects is indicated in red while the other detected objects are marked with a "+" sign at the detected centroid. The "+" signs are drawn in different colors but our separate camera cannot sense them properly. The reader is encouraged to look closely at the images on the right. It can be seen that the system camera sensed traces of the projected contents in some images. This is due to the range of the threshold set during the color classification (Section 3.3). However, these traces are not clear enough to intrude on any environment analysis. Note that the captured images were enhanced here for better visualization.

6. Conclusion and Future Work

In this paper, we investigated hardware setups and software techniques for creating a real-time interactive projection application, including live broadcasting from an advanced virtual studio. The projection device developed in this paper is self-contained and portable, and can be installed or used easily in an existing studio environment. The nonintrusive projection problem is solved by using an off-the-shelf DLP projector together with DLP color wheel analysis, color space sampling and approximate nearest-neighbor search. The proposed solution ensures that the environment is always illuminated, while projected content does not intrude on any environment analysis. For the real-time geometric calibration problem, two approaches with different setups were proposed. On a planar surface, the motion-sensor-based approach can update the geometric mapping in real time. For a more accurate calibration approach that can be applied to planar and nonplanar surfaces, we proposed the beam-splitter-based setup. Related factors of this setup regarding the nonintrusive projection and further image processing were also investigated.

The device and techniques proposed in this paper will help the creation of special effects that appear in response to actors or other studio objects in real time. By integrating the proposed concepts with appropriate target detection algorithms and special effect rendering techniques, an existing virtual studio setup will then support live broadcasting in an actual studio production. For example, in the weather forecast virtual studio, positions of the actor and his hands should be detected so that the weather map can be rendered and projected in response to those detected positions. In the future, we plan to create a robust interactive projection application that benefits from the proposed device and techniques. Furthermore, we are interested in adding human

detection or hand gesture recognition to the system to incorporate the system with a real human.

References

[1] A. Grundhöfer, M. Seeger, F. Häntsch, and O. Bimber, "Dynamic adaptation of projected imperceptible codes," in *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '07)*, 2007.

[2] D. Cotting and M. Gross, "Interactive environment-aware display bubbles," in *Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology (UIST '06)*, pp. 245–254, 2006.

[3] X. Cao and R. Balakrishnan, "Interacting with dynamically defined information spaces using a handheld projector and a pen," in *Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology (UIST '06)*, pp. 225–234, 2006.

[4] X. Cao, C. Forlines, and R. Balakrishnan, "Multi-user interaction using handheld projectors," in *Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology (UIST '07)*, pp. 43–52, 2007.

[5] K. Hosoi, V. N. Dao, and M. Sugimoto, "CoGAME: manipulation by projection," in *Proceedings of the International Conference on Computer Graphics and Interactive Techniques, emerging technologies (SIGGRAPH '07)*, 2007.

[6] P. Mistry, P. Maes, and L. Chang, "WUW—Wear ur world—a wearable gestural interface," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI '09)*, pp. 4111–4116, 2009.

[7] K. Akasaka, R. Sagawa, and Y. Yagi, "A sensor for simultaneously capturing texture and shape by projecting structured infrared light," in *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling (3DIM '07)*, pp. 375–381, 2007.

[8] J. C. Lee, S. Hudson, and P. Dietz, "Hybrid infrared and visible light projection for location tracking," in *Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology (UIST '07)*, pp. 57–60, 2007.

[9] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: a unified approach to image-based modeling and spatially immersive displays," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, 1998.

[10] H. Park, M. H. Lee, B. K. Seo, Y. Jin, and J. I. Park, "Content adaptive embedding of complementary patterns for non-intrusive direct-projected augmented reality," in *Proceedings of the 12th International Conference on Human-Computer Interaction (HCI '07)*, 2007.

[11] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, and M. Gross, "Scalable 3D video of dynamic scenes," *Visual Computer*, vol. 21, no. 8–10, pp. 629–638, 2005.

[12] D. Cotting, M. Naef, M. Gross, and H. Fuchs, "Embedding imperceptible patterns into projected images for simultaneous acquisition and display," in *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*, pp. 100–109, 2004.

[13] S. Borkowski, O. Riff, and J. L. Crowley, "Projecting rectified images in an augmented environment," in *Proceedings of the IEEE International Workshop on Projector-Camera Systems (PROCAMS '03)*, 2003.

[14] M. Fiala, "Automatic projector calibration using self-identifying patterns," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, 2005.

[15] R. Raskar, J. van Baar, and J. X. Chai, "A low-cost projector mosaic with fast registration," in *Proceedings of the IEEE Asian Conference on Computer Vision (ACCV '02)*, 2002.

[16] R. Raskar and P. Beardsley, "A self-correcting projector," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 2, pp. 504–508, 2001.

[17] J. van Baar, T. Willwacher, S. Rao, and R. Raskar, "Seamless multi-projector display on curved screens," in *Proceedings of the Workshop on Virtual Environments (EGVE '03)*, 2003.

[18] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines, "iLamps: geometrically aware and self-configuring projectors," in *Proceedings of the ACM Transactions on Graphics (SIGGRAPH '03)*, no. 3, pp. 809–818, July 2003.

[19] W. Sun, X. Yang, S. Xiao, and W. Hu, "Robust checkerboard recognition for efficient nonplanar geometry registration in Projector-Camera systems," in *Proceedings of the ACM/IEEE 5th International Workshop on Projector Camera Systems (PROCAMS '08)*, 2008.

[20] Y. Ding, J. Xiao, K.-H. Tan, and J. Yu, "Catadioptric projectors," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 2528–2535, 2009.

[21] A. Kushal, J. van Baar, R. Raskar, and P. Beardsley, "A handheld projector supported by computer vision," in *Proceedings of the 7th Asian Conference on Computer Vision (ACCV '06)*, pp. 183–192, 2006.

[22] T. Johnson and H. Fuchs, "Real-time projector tracking on complex geometry using ordinary imagery," in *Proceedings of the International Workshop on Projector-Camera Systems (PROCAMS '07)*, 2007.

[23] S. J. Koppal and S. G. Narasimhan, "Illustrating motion through DLP photography," in *Proceedings of the IEEE International Workshop on Projector-Camera Systems (PROCAMS '03)*, pp. 9–16, 2009.

[24] S. G. Narasimhan, S. J. Koppal, and S. Yamazaki, "Temporal dithering of illumination for fast active vision," in *Proceedings of the 10th European Conference on Computer Vision (ECCV '08)*, pp. 830–844, 2008.

[25] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 1000–1006, 1997.

[26] R. Sukthankar, R. G. Stockton, and M. D. Mullin, "Smarter presentations: exploiting homography in camera-projector systems," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 247–253, 2001.

[27] V. N. Dao, K. Hosoi, and M. Sugimoto, "A semi-automatic realtime calibration technique for a handheld projector," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '07)*, 2007.

[28] K. Fujii, M. D. Grossberg, and S. K. Nayar, "A projector-camera system with real-time photometric adaptation for dynamic environments," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 814–821, 2005.

[29] S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1160–1167, 2002.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 920121, 9 pages
doi:10.1155/2010/920121

Research Article

Flexible Human Behavior Analysis Framework for Video Surveillance Applications

Weilun Lao,[1,2] Jungong Han,[1] and Peter H. N. de With[1,3]

[1] *Eindhoven University of Technology, Den Dolech 2, 5600MB Eindhoven, The Netherlands*
[2] *Guangdong Power Grid Company, 510620 Guangzhou, China*
[3] *Cyclomedia, 4180BB Waardenburg, The Netherlands*

Correspondence should be addressed to Weilun Lao, w.lao@tue.nl

Received 5 October 2009; Accepted 9 January 2010

Academic Editor: Ling Shao

We study a flexible framework for semantic analysis of human motion from surveillance video. Successful trajectory estimation and human-body modeling facilitate the semantic analysis of human activities in video sequences. Although human motion is widely investigated, we have extended such research in three aspects. By adding a second camera, not only more reliable behavior analysis is possible, but it also enables to map the ongoing scene events onto a 3D setting to facilitate further semantic analysis. The second contribution is the introduction of a 3D reconstruction scheme for scene understanding. Thirdly, we perform a fast scheme to detect different body parts and generate a fitting skeleton model, without using the explicit assumption of upright body posture. The extension of multiple-view fusion improves the event-based semantic analysis by 15%–30%. Our proposed framework proves its effectiveness as it achieves a near real-time performance (13–15 frames/second and 6–8 frames/second) for monocular and two-view video sequences.

1. Introduction

Visual surveillance for human-behavior analysis has been investigated worldwide as an active research topic [1]. In order to have automatic surveillance accepted by a large community, it requires a sufficiently high accuracy and the computation complexity should enable a real-time performance. In the video-based surveillance application, even if the motion of persons is known, this is not sufficient to describe the posture of the person. The postures of the persons can provide important clues for understanding their activities. Therefore, accurate detection and recognition of various human postures both contribute to the scene understanding. The accuracy of the system is hampered by the use of a single camera, in case of complex situations and several people undertaking actions in the same scene. Often, the posture of people is occluded, so that the behavior cannot be realized in high accuracy. In this paper, we contribute to improve the analysis accuracy by exploiting the use of second camera and mapping the event into a 3D scene model, that enables analysis of the behavior in the 3D domain. Let us now discuss related work from the literature.

1.1. Related Work. Most surveillance systems have focused on understanding the events through the study of trajectories and positions of persons using *a priori* knowledge about the scene. The Pfinder [2] system was developed to describe a moving person in an indoor environment. It tracks a single nonoccluded person in complex scenes. The VSAM [3] system can monitor activities over various scenarios, using multiple cameras that are connected as a network. It can detect and track multiple persons and vehicles within cluttered scenes and manage their activities over a long period of time. The real-time visual surveillance system W4 [4] employs the combined techniques of shape analysis and body tracking, and models different appearances of a person. This single-camera system detects and tracks groups of people and monitors their behaviors, even in the presence of partial occlusion and in outdoor environments. However,

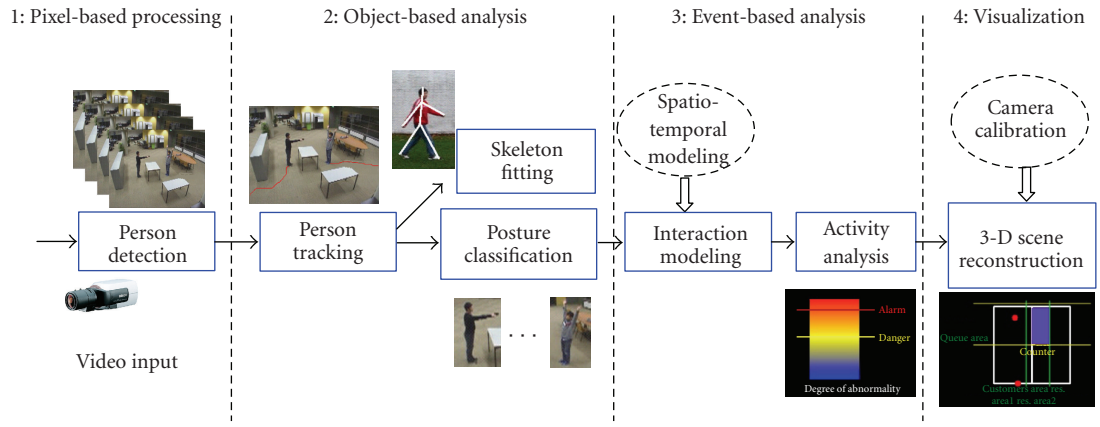


Figure 1: Block diagram of our four-level human motion analysis system.

the above systems generally suffer from the problem that they lack reliable continuous observation of people displacements. The monitoring performances of the above systems mainly rely on the detected trajectories of the concerned objects. Furthermore, the results are not sufficient for event analysis in some cases. As the local properties of the detected persons are missing, the developed systems lack the semantic recognition result of dynamic human activities. In this paper, we explore the *combination* of using trajectory, posture recognition, skeleton fitting, and 3D scene reconstruction in order to improve the semantic analysis of the human behavior. Furthermore, we apply the above techniques to a dual-camera system to improve the accuracy of event recognition.

1.2. 3-D Reconstruction. Scene reconstruction in 3D is a useful tool in semantic-event analysis, which is generally utilized in multimedia applications [5]. The accurate and realistic reconstruction in a virtual space can significantly contribute to the scene understanding, like crime-evidence collection and tactical analysis. Therefore, it is interesting to extend scene-reconstruction functionality in advanced surveillance applications, such as home-care monitoring and robbery-detection surveillance. The 3D scene reconstruction can be conducted to visualize the scene for further analysis. The 3D reconstruction is in fact a mapping of the 2D image data into a 3D real-world model. After the mapping from image to real world is performed, we can estimate the position and calculate the real speed of the persons involved in the video scene. Furthermore, the scene can be reconstructed by realistic 3D models by advanced modeling software to improve the reality. The above postprocessing step extends the framework with better visual presentation. In the application of a bank-robbery detection, for example, this extended processing plays a useful role in the crime-scene analysis, data retrieval, and evidence collection.

The principle of the mapping is basically a homography equation that describes the conversion of 2D point locations into 3D positions. For this purpose, we aim at finding a set of reference points that can be reliably detected. These reference points are used as an input for a multiparameter homography estimation. If sufficient reference points are available, the parameters of the homography can be computed. After this calibration, each input image can be mapped onto the 3D space using the computed homography. If we extend the system to a dual-camera setup, each of the cameras needs to be calibrated as described above. Both camera views are mapped onto the same 3D space. If one person is occluded in one camera view, his position can still be mostly determined by the second camera, so that a reliable 3D scene analysis can be conducted.

1.3. Research Objectives. To address the challenging problem of accurately analyzing human motion and summarizing events at high semantic level, we contribute in three aspects.

(i) A flexible framework is proposed to enable hierarchical human motion analysis. It can be utilized in surveillance applications with four-level analysis results using single or multiple cameras.

(ii) A 3D reconstruction scheme is introduced for scene understanding based on automatic camera calibration. The location and posture of persons are visualized in a 3D space after context knowledge is integrated. More specifically, the 2D-3D mapping provides a platform for a normalized motion configuration (i.e., location and speed) and scene visualization/analysis in the real world.

(iii) A fast scheme is proposed to detect different body parts in human motion. More specifically, for every individual person, features of body ratio, silhouette, and appearance are integrated into a hybrid model to detect body parts. The conventional assumption of upright body posture is not required.

In the sequel, we first present a system overview in Section 2 and then describe in detail the techniques for each level in Section 3. The experimental results on surveillance video are provided in Section 4. Finally, Section 5 draws conclusions.

2. Our Proposed System Framework

Our work aims at the object/scene analysis and behavior modeling of deformable objects. The framework captures the human motion, analyzes and demonstrates its gesture/activity, infers the semantic event exploiting interaction modeling, and performs the 3D scene reconstruction. The previous analysis is only possible if the scene and its objects are analyzed at various levels (e.g., background modeling, moving objects, event recognition, etc.). The block diagram of our multilevel event-analysis system is shown in Figure 1. The term multilevel refers to the four different conceptual levels: *pixel-based level, object-based level* (including trajectory estimation, posture classification, and skeleton fitting), *event-based level,* and *visualization level*.

(i) *Pixel-based level*. The background modeling and object detection are implemented. Each image within the video covering an individual human body is segmented to extract the "blobs" representing foreground objects. These detected blobs are refined afterwards to produce the human silhouette.

(ii) *Object-based level*. It performs trajectory estimation, posture classification, and skeleton fitting. We first track every moving person. Afterwards, a shape-based analysis is conducted to classify different posture types. Finally, a skeleton model is adaptively produced for every object.

(iii) *Event-based level*. Interaction relationships are modeled to infer a multiple-person event. This semantic analysis is thus responsible for the human activity recognition.

(iv) *Visualization level*. With the aim of 2D-3D mapping calibration, the 3D scene reconstruction is conducted to visualize the scene for further analysis. This level can be simple for home use, but advanced for professional use (e.g., after crime analysis in 3D).

The purpose of the framework is that it should be powerful and robust enough to facilitate a few different surveillance applications. To fulfill this objective, the semantic-level analysis should be of sufficiently high performance. In the sequel, home-care monitoring and the detection of a robbery for security surveillance are our key applications.

3. Techniques for Human Behavior Analysis

3.1. Trajectory Estimation. At the pixel-based level, the human silhouette is detected based on background subtraction. This general method can be used to segment moving objects in a scene, assuming that the camera is stationary and the lighting condition is fixed. To improve the blob segmentation, a shadow-removing approach [6] is used in our scheme. The false segmentation caused by shadows can be minimized by computing differences in a color space (RGB) that is less sensitive to intensity changes.

At the object-based level, person tracking (trajectory estimation) and posture classification are performed. In the trajectory-estimation step, we employ the broadly accepted mean-shift algorithm for tracking persons, based on their individual appearance model represented as a color histogram. When the mean-shift tracker is applied, we extract every new person entering the scene and calculate the corresponding histogram model in the image domain. In subsequent frames for tracking that person, we shift the person object to the location whose histogram is the closest to the previous frame. Afterwards, from our previous work [7], we have adopted the Double Exponential Smoothing (DES) operator to track moving persons. This filter runs approximately 135 times faster than the popular Kalman filter-based predictive tracking algorithm, with equivalent prediction performance. When the trajectory is obtained, we can estimate the position of the persons involved in the video scene. Therefore, we can conduct a body-based analysis at the location of the person in every frame.

Based on the results of trajectory estimation, the person action is classified into three types: running, walking, and standing. In the case of standing, the speed of the moving person is below a predefined threshold. Only in that case, posture classification is performed, which will be addressed in the next subsection.

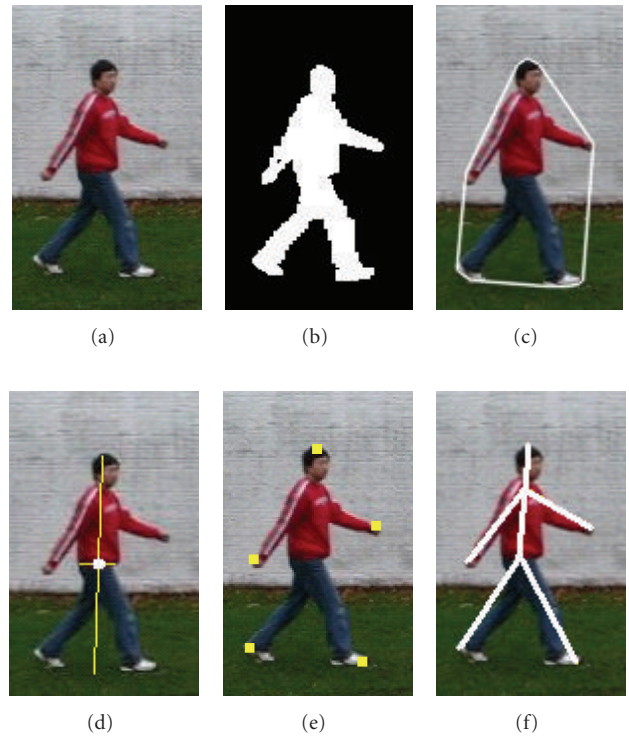


Figure 2: Procedure of skeleton-fitting processing: (a) original frame, (b) foreground segmentation (after shadow removal), (c) body modeling based on convex hull, (d) center-point estimation, (e) body-part location and, (f) skeleton construction in single-person motion.

3.2. Individual Posture Recognition with CHMM. We adopt a simple but effective shape descriptor to analyze the human

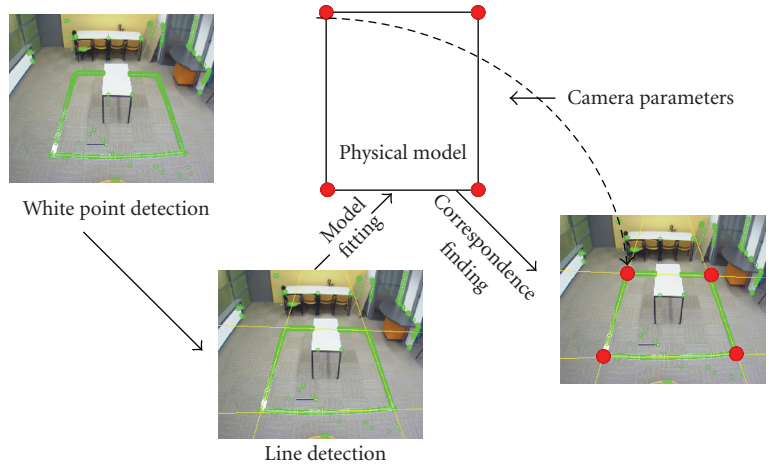


Figure 3: Example image of the corresponding homography based on camera calibration.

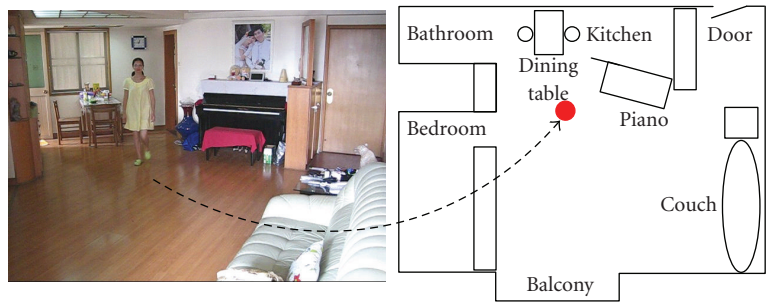


Figure 4: Example image of corresponding 2D/3D mapping in home-care monitoring.

silhouette prior to conducting the temporal modeling scheme of a Continuous Hidden Markov Model (CHMM) to recognize the posture type.

Individual posture classification is important for human-activity recognition. Firstly, we adopt a shape-based descriptor to analyze the human silhouette. Our posture classifier utilizes two features commonly used for object classification: area and the ratio of the bounding box attached to each detected object. This approach is simple but efficient, and it contributes significantly to the tracking and avoids a complex procedure for training data. The nonperson objects and image noise can be effectively removed. The disturbance generated from different person heights is also considered. We perform a training step regarding different heights in the scene before an adaptive threshold is applied. Finally, we can obtain the observed 2D feature vector of the silhouette.

Due to noise from segmentation errors, a single-frame recognition is not sufficiently accurate when we require general motion classification. The temporal consistency is required for a good posture recognition. Therefore, we adopt the HMM as our posture classifier, since it has proven to be an effective tool for sequential data processing. We use the Continuous Hidden Markov Model (CHMM) with left-right topology [8]. Suppose a CHMM has E states $Q = \{q_1, q_2, \ldots, q_E\}$ and F output symbols $V = \{v_1, v_2, \ldots, v_F\}$. It is fully specified by the triplet $\lambda = \{A, B, \pi\}$. Let the state at time step t be s_t; then the $E \times E$-state transition matrix $\mathbf{A}$ can be defined by

$$\mathbf{A} = \left\{a_{ij} \mid a_{ij} = P\left(s_{t+1} = q_j \mid s_t = q_i\right)\right\}, \quad 1 \le i,\ j \le E. \tag{1}$$

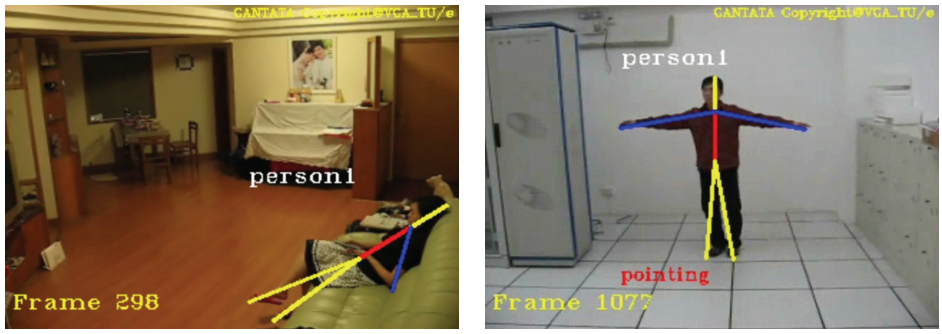
The $E \times F$-state output probability matrix $\mathbf{B}$ is defined as

$$\mathbf{B} = \left\{b_j(k) \mid b_j(k) = P\left(v_k \mid s_t = q_j\right)\right\}, \quad 1 \le j \le E, \quad 1 \le k \le F. \tag{2}$$

The initial state distribution vector π is specified as

$$\pi = \{\pi_i \mid \pi_i = P(s_1 = q_i)\}, \quad 1 \le i \le E. \tag{3}$$

We assign a CHMM model to each of the predefined posture types for the observed human body. Each CHMM is trained based on the Baum-Welch algorithm [8]. The learning process can calculate all parameters of the model using the training data. In other words, the triplet λ is obtained for each model. After having the models for each posture, we can proceed to implement the online testing. Given an observation sequence $Obv = \{Obv_1, Obv_2, \ldots, Obv_T\}$, we can calculate $P(Obv \mid \lambda_i)$, which is the probability of the observation sequence Obv given model i with λ_i. The probability $P(Obv \mid \lambda_i)$ can be obtained by using the forward algorithm [8]. After each model's output probability



(a) (b)

Figure 5: Example images of the skeleton-fitting result of human activity in two indoors events.

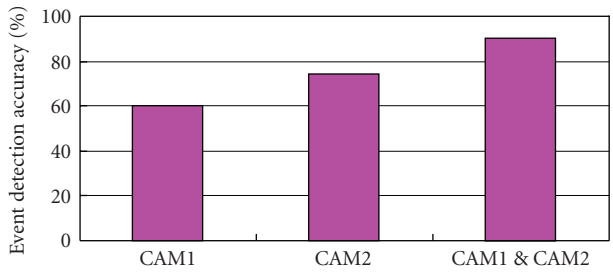


Figure 6: Event detection classification results based on single/multiple cameras.

is calculated, the model with maximum probability is chosen as the recognition result. We can therefore recognize the posture class C_T as being the one that is represented by the maximum probable model among K types, which is specified by

$$C_T = \arg\max_i P(Obv \mid \lambda_i), \quad 1 \le i \le K. \tag{4}$$

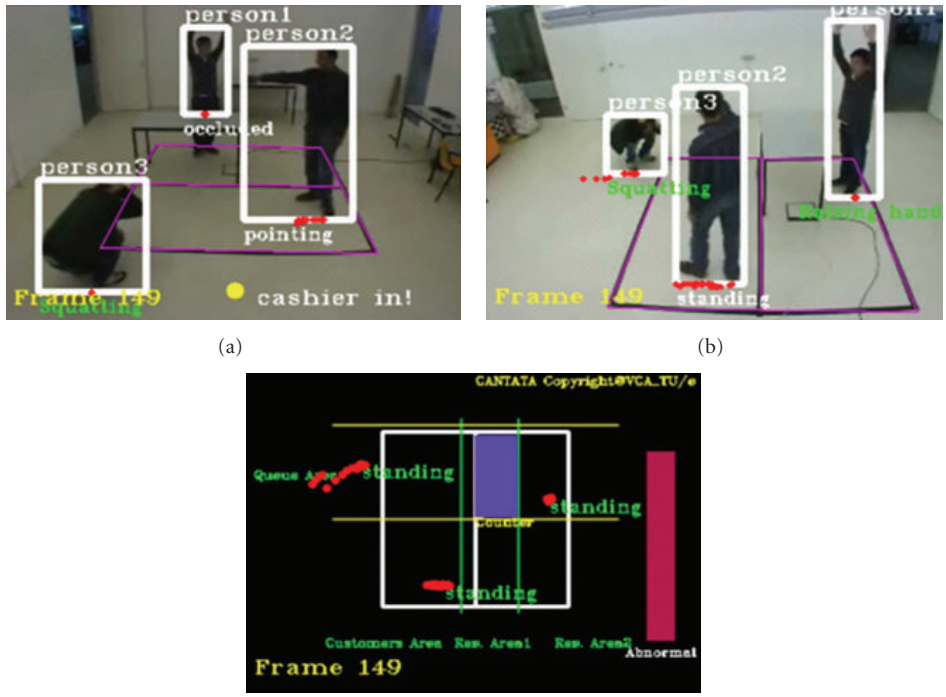
In our investigated case ($T = 20$, $K = 4$), every given posture is finally classified into one of the following types: *pointing, squatting, raising hands overhead,* and *lying*.

3.3. Skeleton Fitting. The purpose of skeleton fitting is to visualize the behavior of the person. To this end, we need to detect individual body parts at each frame. Since this has roots in earlier scientific research, we first briefly present an overview of this work below.

Accurate detection and efficient tracking of various body parts play an important role in presenting the human behavior. Existing fast techniques can be classified into two categories: appearance-based and silhouette-based methods. *Appearance-based* approaches [9, 10] utilize the intensity or color configuration within the whole body to infer specific body parts. They can simplify the estimation and collection of training data. However, they are significantly affected by the variances of body postures and clothing. For the *silhouette-based* approach [11–13], different body parts are located employing the external points detected along the contour, or internal points estimated from the shape analysis. The geometric configuration of each body part is modeled prior to performing the pose estimation of the whole human body. However, the highly accurate detection of body parts remains a difficult problem, due to the effectiveness of segmentation. Human limbs are often inaccurately detected because of the self-occlusion or occlusion by other objects/persons. Summarizing, both silhouette and appearance-based techniques do not offer a sufficiently high overall accuracy of body-part detection. Also, the assumption of upright posture is generally required. In our work, we do not need the assumption of an upright posture. We had to design a new algorithm because in the desired applications, persons are not always in an upright position. In the following, we summarize our algorithm that was reported first in [14].

We develop a fast scheme to detect different body parts in human motion. More specifically, for every individual person, features of body ratio, silhouette, and appearance are integrated into a hybrid model to detect body parts. The conventional assumption of upright body posture is not required. The skeleton-fitting processing step models the human motion by a skeleton model. The detailed procedure is illustrated in Figure 2. In the example of single-person motion, the input frame (Figure 2(a)) is first subject to shadow removal, and then segmented to produce a foreground blob (Figure 2(b)). Afterwards, the convex hull is implemented for the whole blob (Figure 2(c)). The dominant points along the convex hull are strong clues, in the case of single-person body-part detection. They infer the possible locations of the ending points of body parts, like head, hands, and feet. Here we employ a *content-aware* scheme to estimate the center point (Figure 2(d)), which is fundamentally used to position the human skeleton model. Meanwhile, dominant points along the convex hull are selected and refined to locate the head, hands and feet (Figure 2(e)). Finally, different body parts are connected to a predefined skeleton model involving a center point, where the skeleton is adapted to the actual situation of the person in the scene (Figure 2(f)).

3.4. Interaction Modeling. In multiperson events, the event analysis is achieved by understanding the interactions among



(a) (b)

(c)

Figure 7: Example of our simulated multicamera robbery-detection result.

people involved in the scene. The temporal constraints of two-person interactions are defined by two events in terms of causal and coinciding relations of the two persons' posture changes. The events are seldom instantaneous and often significantly rely on the temporal order and relationship of their subevents (the individual posture). We introduce appropriate spatial and temporal constraints for each of the various two-person interaction patterns as domain knowledge. The satisfaction of specific spatial/temporal constraints contributes to the semantic recognition of the interaction. Therefore, the event-level recognition is characterized by the integration of domain-specific knowledge, whereas the object-level recognition is more closely related to the pure motion of a human body.

In order to represent temporal relationships of subevents, we apply the temporal logic based on interval algebra, as used in [15]. Seven temporal relationships are indicated from the set of $TR = \{after, meets, during, finishes, overlaps, equal, starts\}$. These keywords can link different sub-events after the individual event is analyzed. In this way, the scene becomes a chain of sub-events which are linked by the previously mentioned key words. To describe the semantic meaning of the scene, we apply heuristic rules. For example, in the application of a bank-robbery detection, the heuristic rules are based on expert knowledge. In our investigated case of robbery detection [14], the posture "pointing" is a key reference posture. It can significantly infer the robbery event. Other postures are also estimated to improve the recognition accuracy based on specific temporal constraints. Suppose person A has an action labeled as "pointing", person B is detected to be "raising both hands" and person C is "squatting" during the sub-event from person A, we can infer that a robbery actually occurs. After performing the interaction modeling, we are able to calculate the degree of abnormality. If the degree value is above a predefined threshold, the surveillance system will trigger the alarm signal (e.g., when a detected robbery event happens). The advantage of using such a metric is that an abnormal situation can raise the degree value to alert security people without signaling an alarm. The degree value can thus be used as a preventive measure, rather than alarming when the actual robbery takes place.

3.5. 3D Scene Reconstruction. Based on the automatic or manual camera calibration, we can implement the 2D-3D mapping. In other words, the image contents can be described in a 3D world domain. Furthermore, the real scene is reconstructed in a virtual space. This 3D scene reconstruction is useful for after-crime analysis and it contributes to the crime-evidence collection.

The objective of the camera calibration is to provide a geometric transformation that maps the points in the image domain to the real-world coordinates. An example of the scene reconstruction is visualized in Figure 3. In our system, we analyze the human behavior based on the person's trajectory and/or speed on the ground, so that the height information of the human is not required. Since both the ground and the displayed images are planar, the mapping between them is a homography, which can be written as

Table 1: The detection results for human activity recognition in home-care monitoring.

	In kitchen	Sitting at dining table	Sitting on couch	Playing piano	To balcony	In bedroom	In bathroom	Enter/Leave by door
In kitchen	14/16	2/16	0	0	0	0	0	0
Sitting at dining table	0	8/8	0	0	0	0	0	0
Sitting on couch	0	0	10/10	0	0	0	0	0
Playing piano	0	1/8	0	7/8	0	0	0	0
To balcony	0	0	0	0	10/10	0	0	0
In bedroom	0	0	0	0	2/12	10/12	0	0
In bathroom	0	1/7	0	0	0	0	6/7	0
Enter/Leave by door	0	0	0	0	0	0	0	5/5

a 3×3 transformation matrix H, transforming a point $p = (x, y, z)^T$ in image coordinates to the real-world coordinates $p' = (X, Y, Z)^T$ with $p = Hp'$, which is equivalent to

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \quad (5)$$

The transformation matrix H can be calculated from four points whose positions are known both in the real world and in the image. In our previous work [7], we have developed an automatic algorithm to establish the homography mapping for analyzing a tennis video, where the court lines and their interaction points are identified in the image. Such lines and points are related to the lines and points in a standard tennis court. Therefore, the homography mapping described in (5) can be established after the correspondences are found. This approach has been adopted in our surveillance system. The basic idea is to manually put four white lines forming a rectangular on the ground (see Figure 3). We have measured the length of each line in the real world, thereby defining their coordinates in the real-world domain. Afterwards, the algorithm proposed in our previous work can be applied for calculating parameters of the homography mapping. The complete algorithm comprises four steps, which are white-pixel detection, line detection, finding intersection points and calculating the parameters. For more details, we refer to an earlier publication [7].

After performing the mapping from the image to real world, we can estimate the position and calculate the real speed of the persons involved in the video scene. The label of walking or standing can be therefore assigned to an individual person. Furthermore, the scene can be reconstructed in 3D space. The above post-processing step extends our framework with better visual presentation and scene understanding. For instance, the 3D location of every moving person infers his actual behavior in the application of home-care monitoring. In the application of a bank-robbery detection, this extended processing is useful in the crime-scene analysis, data retrieval and evidence collection.

4. Experimental Results

We have trained our framework using 10 video sequences of various single/multiperson motion (15 frames/s) in both home-care and robbery-event scenarios. Then we have used 15 similar sequences for testing. On the frame basis, we have obtained a 98% accuracy rate on person detection, 95% detection rate on person tracking (where the criterion is that at least 70% of the human body is included in the detection window), and 82% detection rate on posture classification (in the robbery-event scenario).

4.1. Single-Camera Experiment: Home-Care Monitoring. Based on the trajectory estimation, we can calculate the speed and estimate the location of each individual person. We have conducted the experiment in our first case study on home-care monitoring. The experimental videos were captured in an apartment involving 6 persons (with different gender, height, age and clothes). The length of video sequences is more than 2 hours. The layout of the apartment is demonstrated in Figure 4. Based on the detected location and speed, the human daily activity is classified into 8 types (in kitchen, sitting at dining table, sitting on couch, playing piano, to balcony, in bedroom, in bathroom, and enter/leave by door). The classification results of activity recognition (involving the detected sequence numbers and total test sequence numbers, zero means that no corresponding sequence is detected) are summarized in Table 1. In our experiments, the ground truth of body-part locations were manually obtained. The maximum tolerable errors in the evaluation is set to 20 pixels. The skeleton model is further reconstructed on the individual body. Two examples of such a modeled presentation are portrayed by Figure 5.

4.2. Dual-Camera Experiment: Robbery-Event Detection. To further combat the problem of occlusion, multiple cameras are employed for capturing the same scene from different angles. We have conducted a dual-camera experiment in our second case study on a robbery-event detection. We analyze both camera views and combined the semantic of both views into one degree of abnormality. Currently, an OR logic operator is applied to link the two viewpoints at the level of the abnormal-event detection. Two different event types (normal and abnormal) are defined based on

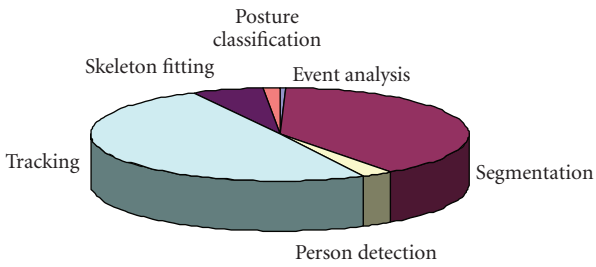


Figure 8: System performance: average time-consuming percentage of each module.

domain knowledge. The detection accuracy for each camera and the combination are shown in Figure 6. It proves that the dual-camera scheme effectively improves the event-based semantic analysis. Figure 7 shows a detection example of a simulated bank-robbery event. The position of every person is visualized after trajectory estimation. The postures are estimated and the semantic event is highlighted after interaction modeling from two different viewpoints (Figures 7(a) and 7(b)). The camera calibration is performed and the 2D-3D mapping is visualized. The degree of abnormality is also calculated and shown in Figure 7(c). Although the posture pointing is not recognized in one camera (see Figure 7(b)), it is correctly recognized in the other camera (see Figure 7(a)). The robbery event is successfully detected afterwards.

4.3. System Performance. Our system performance was tested by video sequences at 640 × 480 resolution (VGA), with a P-IV 3-GHz PC. Results show that our system fulfils the real-time requirement, as 13–15 frames/second and 6–8 frames/second are obtained for monocular and two-view video sequences, respectively. Figure 8 presents the average cycle-consumption percentage of every module. We have assumed that camera calibration is an off-line process taking place first. As can be noticed, foreground/background segmentation and tracking modules consume most computing cycles.

5. Conclusion

We have proposed a layered framework that enables multilevel human motion analysis, featuring layers at pixel, object, event and visualization level. The framework captures the human motion, classifies its posture, generates a fitting skeleton model after body-part detection, infers the semantic event exploiting interaction modeling, and performs the 3D scene reconstruction. We have applied this framework in a single-camera setup and a dual-camera setup. In the last case, it is possible to benefit from the extra view in case of occlusion and it may also add to after-crime analysis. This extension of multiple-view fusion improves the event-based semantic analysis by 15%–30%.

The framework was evaluated for two applications, a home-care monitoring case and a robbery-detection case. The practical results have shown that our framework can be used for various surveillance cases. The choice of using single or multiple cameras is basically independent on the type of surveillance applications and it is more ruled by the quality requirements or the occurrence of occlusions. Performance evaluations have shown that the framework is efficient and achieves a fast performance (13–15 frames/second and 6–8 frames/second) for monocular and two-view video sequences. Therefore, it can be used in an embedded system implementation.

We are improving the effective modeling of multi-person interaction, in order to obtain a probability-based inference engine. The self-occlusion problem is not yet thoroughly tackled at the current stage. Thus we intend to integrate an effective occlusion-handling module, which was reported in [16] to improve the motion-analysis robustness.

References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 34, no. 3, pp. 334–352, 2004.

[2] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: realtime tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780–785, 1997.

[3] R. T. Collins, A. J. Lipton, T. Kanade, et al., "A system for video surveillance and monitoring," Tech. Rep. CMU-RI-TR-00-12, CMU, Pittsburgh, Pa, USA, 2000.

[4] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[5] J. Han, D. Farin, and P. H. N. de With, "A real-time augmented-reality system for sports broadcast video enhancement," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 337–340, Augsburg, Germany, 2007.

[6] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.

[7] J. Han, D. Farin, P. H. N. de With, and W. Lao, "Real-time video content analysis tool for consumer media storage system," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 870–878, 2006.

[8] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[9] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 734–741, 2003.

[10] S. Park and J. K. Aggarwal, "Simultaneous tracking of multiple body parts of interacting persons," *Computer Vision and Image Understanding*, vol. 102, no. 1, pp. 1–21, 2006.

[11] H. Fujiyoshi, A. J. Lipton, and T. Kanade, "Real-time human motion analysis by image skeletonization," *IEICE Transactions on Information and Systems*, vol. E87, no. 1, pp. 113–120, 2004.

[12] C.-C. Yu, J.-N. Hwang, G.-F. Ho, and C.-H. Hsieh, "Automatic human body tracking and modeling from monocular video sequences," in *Proceedings of the IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP '07), vol. 1, pp. I917–I920, Honolulu, Hawaii, USA, 2007.

[13] P. Peursum, H. H. Bui, S. Venkatesh, and G. West, "Robust recognition and segmentation of human actions using HMMs with missing observations," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 13, pp. 2110–2126, 2005.

[14] W. Lao, J. Han, and P. H. N. de With, "Fast detection and modeling of human-body parts from monocular video," in *Articulated Motion and Deformable Objects*, vol. 5098 of *Lecture Notes in Computer Science*, pp. 380–389, Springer, Berlin, Germany, 2008.

[15] J. F. Allen and G. Ferguson, "Actions and events in interval temporal logic," *Journal of Logic Computation*, vol. 4, pp. 531–579, 1994.

[16] J. Han, M. Feng, and P. H. N. de With, "A real-time video surveillance system with human occlusion handling using nonlinear regression," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '08)*, pp. 305–308, Hannover, Germany, 2008.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 709161, 11 pages
doi:10.1155/2010/709161

Research Article

Statistical Skimming of Feature Films

Sergio Benini, Pierangelo Migliorati, and Riccardo Leonardi

Department of Information Engineering DII - SCL, Università di Brescia, via Branze 38, 25123 Brescia, Italy

Correspondence should be addressed to Sergio Benini, sergio.benini@ing.unibs.it

Received 29 August 2009; Accepted 21 December 2009

Academic Editor: Ling Shao

We present a statistical framework based on Hidden Markov Models *(HMMs)* for skimming feature films. A chain of *HMMs* is used to model subsequent story units: *HMM* states represent different visual-concepts, transitions model the temporal dependencies in each story unit, and stochastic observations are given by single shots. The skim is generated as an observation sequence, where, in order to privilege more informative segments for entering the skim, shots are assigned higher probability of observation if endowed with salient features related to specific film genres. The effectiveness of the method is demonstrated by skimming the first thirty minutes of a wide set of action and dramatic movies, in order to create previews for users useful for assessing whether they would like to see that movie or not, but without revealing the movie central part and plot details. Results are evaluated and compared through extensive user tests in terms of metrics that estimate the content representational value of the obtained video skims and their utility for assessing the user's interest in the observed movie.

"*I took a speed reading course and read "War and Peace" in 20 minutes. It involves Russia.*"

Woody Allen.

1. Introduction

In the last years, with the proliferation of digital TV broadcasting, dedicated internet websites, and private recording of home video, a large amount of video information has been made available to end-users. Nevertheless, this massive proliferation in the *availability* of digital video has not been accompanied by a parallel increase in its *accessibility*. In this scenario, video summarization techniques may represent a key component of a practical video-content management system. By watching a condensed video, a viewer may be able to assess the relevance of a programme before committing time, thus facilitating typical tasks such as browsing, organizing, and searching video-content.

For unscripted-content videos such as sports and home-videos, where the events happen spontaneously and not according to a given script, previous work on video summarisation mainly focused on the extraction of *highlights*. Regarding scripted-content videos—those videos which are produced according to a script, such as feature films (e.g., Hollywood movies), news and cartoons—two types of video abstracts have been investigated so far, namely, *video static summarization* and *video skimming*. The first one is a process that selects a set of salient key-frames to represent content in a compact form and present it to the user as a static programme preview. Video skimming instead, also known as *video dynamic summarization*, tries to condense the original video in the more appealing form of a shorter video clip. The generation of a skim can be viewed as the process of selecting and gluing together proper video segments under some user-defined *constraints* and according to given *criteria*. On the one hand, final user *constraints* are usually defined by the time committed by the user to watch the skim, which in the end determines the final skim ratio. On the other hand, skimming *criteria* used to select video segments range from the use of motion information [1], the exploitation of the hierarchical organization of video in scenes and shots as in [2], or the insertion of audio, visual, and text markers [3].

In this paper, in order to derive the skim, we propose to combine the information deriving from the story structure with the characterization of the shots in terms of salient features, that are motion dynamics for action movies, and the presence of human faces for dramas, respectively. These salient features inherently estimate the contribution

of each shot in terms of "content informativeness" and determines whether the shot will be included in the final skim. The "structure informativeness" of the video is instead captured by *HMMs*, which model semantic scenes and whose observations produce the shot sequence of the final skim.

The paper is organized as follows. Section 2 gives a brief overview on the current state-of-the-art related to video skimming and the other techniques here employed. Section 3 presents the criteria adopted to realize the skim. In Section 4, we characterize the content informativeness of shots by the use of salient features related to the movie genres. Section 5 describes how to model each story unit by an *HMM*. In Sections 6 and 7, the video skims are generated and evaluated, while in Section 8 conclusions are drawn.

2. Related Work

In the past, *HMM* has been successfully applied to different domains such as speech recognition [4], genome sequence analysis [5], and so forth. For video analysis, *HMMs* have been used to distinguish different genres [6], and to delineate high-level structures of soccer games [7]. In this work instead, *HMMs* are used as a unified statistical framework to represent visual-concepts and to model the temporal dependencies in story units with the aim of video skimming.

Even if the interest in effective techniques for dynamic video skimming is highly in demand, to date, there are relatively less works that address dynamic video skimming than works related to static video summarisation (see, e.g., [8] for a systematic classification of previous works on condensed representations of video content). In general, to process huge quantities of video frames is more difficult than to select a subset of relevant key-frames. It is also challenging to define which segments have to be highlighted and mapping the mechanisms of human perception into an automated abstraction process. For these reasons, at the moment most current video skimmings are intended as natural evolutions of the methods employed for generating the related static summaries. Therefore, many skimming methodologies rely on the same clustering algorithms which have been adopted to obtain static video summaries, such as those that have been extensively reviewed in [9]. For example, in one of the latest works [10], the authors propose an algorithm for video summarization which first constructs story boards and then it removes redundant video content using hierarchical agglomerative clustering at the key-frame level.

Since it is easy to understand how a tool that can automatically shorten the original video while preserving only the important content would be greatly useful to most users, alternative skimming methods have been developed in time. The oldest and most straightforward approach is to compress the original video by speeding up the playback without considerable distortion, as pointed out by Omoigui [11]. A similar approach is also described in [12], where an audio time-scale modification scheme is applied. However, these techniques only allow a maximum time compression of around 2.5 times, depending on the rate of speech; since once the compression factor goes beyond this range, the perceived speech quality becomes quite poor or annoying. A similar method is found in [13], where the skim generation is formulated as a rate-distortion optimisation problem.

A number of approaches use attention and saliency models to derive the video skim. In [14] summaries are generated by merging together those video segments that contain high-confidence scores in terms of motion-attention. In a further generalisation described in [1], the same authors take into account also the presence of human faces and the present audio information. One limitation of these two approaches, that we try to overcome in this work, is that the structural information such as the intershot relationship is not exploited for video skimming. As a result, the produced dynamic video summary is purely a collection of video highlights in terms of attention model and does not take into account the content coverage and relationships. In [15] a method for the detection of perceptually important video events, based on saliency models for the audio, visual and textual information, is also described.

Other techniques rely on the presence of textual information only. The Informedia project [3], for example, concatenates audio and video segments that contain preextracted text key-words to form the skim, for example from news.

Without relying on text cues, in [16] skims are generated by a dynamic sampling scheme. Videos are first decomposed into subshots, and each subshot is assigned a motion-intensity index. Key-frames are then sampled from each subshot based on an assigned rate which is derived by the motion index. During the skim playback, techniques of linear interpolation are adopted to provide users with a dynamic storyboard. Similar methods for skim generation based on precomputed key-frames are described in [17, 18].

Singular Value Decomposition and Principal Component Analysis have been also proposed in [19] as attractive models for video skimming. However, these techniques remain computationally intensive since they process all video frames, which cannot be practical for huge repositories.

More recently, research on generating skims for specific unscript-content video has been reported. For example, ad-hoc summarisation methods for rushes were designed within the RUSHES project [20] and have been a field of competition in the TrecVid 2008 [21] as in the related work in [22].

In [23], a skimming system for news is presented. By exploiting news content structure, commercials are removed by using audio cues, and then anchor persons are detected (using a Gaussian Mixture Model) and glued together to form the skim.

Home video skimming is addressed in many works, such as in [2, 24]. In this last work, video skimming is based on media aesthetics. Given a video and a background music, this system generates a music-video-style skimming video automatically, with consideration of video quality, music tempo, and the editing theory.

Some research efforts [25] have been investigating the generation of skims for sports videos based on the identification of exciting highlights such as soccer goals or football touchdowns, therefore, according to the significance of play scenes.

Finally, regarding movies and narrative video, the rules of cinematic production are exploited in [26–28] to produce a syntactical-based reduction scheme for skim generation. Based on both audio and visual information, some utility functions are modelled to maximise the content and coherence of the summaries.

3. Skimming Criteria

Since a skimming application should automatically shorten the original video while preserving the important and informative content, in this work it is proposed that the time allocation policy for realising a skim should fulfil the following criteria.

(i) "*Coverage*". The skim should include all the parts of the movie structure into the synopsis. Since in the movie cinematic syntax, the story structure constitutes a fundamental element for conveying the movie message, each *Logical Story Units (LSU)*, that is, each "sequence of contiguous and interconnected shots sharing a common semantic thread" [29] (which is the best computable approximation to a semantic scene), should participate the skim. Therefore, if V is the original video of total length $l(V)$, we consider it to be already segmented into n Logical Story Units Λ_i by previous analysis as in [30] or in [31], that is, $V = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_n\}$. The final skim v will contain the skimmed version of each *LSU*, that is, $v = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$.

(ii) "*Representativeness*". Each Logical Story Unit should be represented in the skim proportionally to its duration in the original video. Therefore, if r is the skimming ratio defined by the user (i.e., $r = l(v)/l(V)$), the length of each *LSU* in the synopsis should be $l(\lambda_i) = r \cdot l(\Lambda_i)$, for all $i = 1, \ldots, n$.

(iii) "*Structure Informativeness*". Since in the movie cinematic syntax, the information which is introduced by the film editing process, especially by the shot patterns inside story units (e.g., dialogues, progressive scenes, etc.), is relevant for the storytelling and for conveying the plot, this information should be preserved in the final skim. Therefore, if Λ_i is a *dialogue* in the movie V, the corresponding story unit $\lambda_i \in v$ should preserve the dialogue structure.

(iv) "*Content Informativeness*". To represent each story unit, the most "informative" video segments should be preferred. Of course, "informative" is a term that can assume multiple meanings depending on context. Since we are dealing with movies, a segment is intended as "informative" if it is effective in conveying the general concept presented in the film. One possibility of relating this to some physical properties of the video material is to link the concept of "informative" segment with the genre of the movie. If the film is an action movie, for example, an informative segment will be a high dynamic one, while in case of a dramatic film, informative segment will be those showing key dialogues between main characters. We can therefore quantify the "informativeness" of a video segment by assessing in the video the presence of one *salient feature* $\mathcal{F}$ which is related to the film genre.

4. Salient Features

In order to assess the *content informativeness* of each shot of the film, we introduce the concept of salient feature $\mathcal{F}$ related to the movie genre. The skimming procedure that follows this description is general and can be applied to any film provided that a salient feature for that movie genre is defined. The user can also choose to apply a salient feature which is not related to the movie genre, just because he/she is more interested in that, or as a leisure activity, for example, in the context of a video mash-up application. There is of course no limit to the set of salient features that can be defined.

In order to provide a couple of examples of possible salient features, we shortly describe in the following a measure of motion activity which can be useful to skim action movies, and a face detection procedure to assess the presence of human faces in dramatic movies, where most information is conveyed by dialogues between characters.

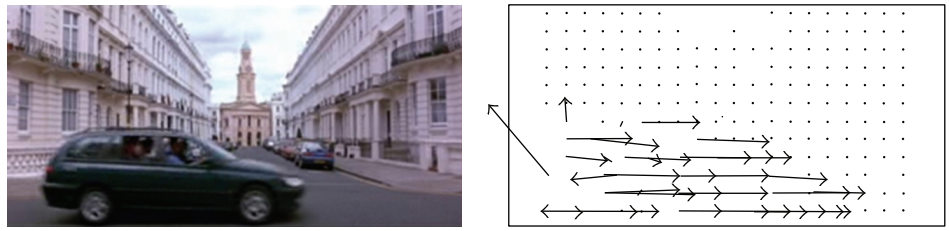
4.1. Motion Activity. The intensity of motion activity is a subjective measure of the perceived intensity of motion in a video segment. For instance, while an "*anchorman*" shot in a news program is perceived by most people as a "low-intensity" action, a "*car chasing*" sequence would be viewed by most viewers as a "high-intensity" sequence.

As stated in [32], the intensity of motion activity in a video segment is in fact a measure of "how much" the content of a video is changing. Motion activity can be therefore interpreted as a measure of the "entropy" (in a wide sense) of a video segment. We characterize the motion activity of video shots by extracting the motion vector *(MV)* field of *P*-frames (see Figure 1) directly from the compressed *MPEG* stream, thus allowing low computational cost.

For compression efficiency, *MPEG* uses a motion-compensated prediction scheme to exploit temporal redundancy inherent in an image sequence. In each *GOP* (Group of Pictures), *I*-frames are used as references for the prediction. *P*-frames are coded using motion-compensated prediction from a previous *P* or *I*-frame (forward prediction), while *B*-frames are coded by using past and/or future pictures as references. This means that, in order to reduce the bit-rate, macroblocks (*MBs*) in *P* and *B*-frames are coded using their differences with corresponding reference *MBs*, and a motion vector carries the displacement of the current *MB* with respect to a reference *MB*.

The raw *MV* field extracted turns out to be normally rough and erratic, and not suitable for tasks such as accurately segmenting moving objects. However, after being properly filtered, the *MVs* can be very useful to describe the general motion dynamics of a sequence, thus characterising the amount of visual information conveyed by the shot.

The filtering process applied includes first removing the *MVs* next to image borders which tend to be unreliable, then using a texture filter, followed by a median filter. The texture filter is needed since, in the case of low-textured uniform



(a) (b)

Figure 1: A decoded *P*-frame and its motion vector field.

areas, the correlation methods used to estimate motion often produce spurious *MVs*. After having filtered the motion vectors on texture criterion, a median filtering is used to straighten up single spurious vectors such as those that could still be present close to borders.

In general, the perceived motion activity in a video is higher when the objects in the scene move faster. In this case the magnitudes of the *MVs* of the macroblocks (*MBs*) that make up the objects are significant, and one simple measure of motion intensity can be extracted from the *P*-frame by computing the mean μ_P of the magnitudes of motion vectors belonging to intercoded *MBs* only.

However, most of the perceived intensity in a video is due to objects which do not move according to the uniform motion of the video camera. Thus, a good *P*-frame-based measure of motion intensity is given by the standard deviation σ_P of the magnitudes of motion vectors belonging to intercoded *MBs*.

The measure σ_P can be also extended to characterize the motion intensity $\mathcal{MI}(S)$ of a shot *S*, by averaging the measures obtained on all the *P*-frames belonging to that shot. *MPEG7 Motion Activity* descriptor [32] is also based on a quantized version of the standard deviation of *MVs* magnitudes. For our purposes, each shot *S* is assigned its motion intensity value $\mathcal{MI}(S)$ in its not-quantized version. This value $\mathcal{MI}(S)$ tries to capture the human perception of the "intensity of action" or the "pace" of a video segment, by considering the overall intensity of motion activity in the shot itself (without distinguishing between the camera motion and the motion of the objects present in the scene). Since this is in fact a measure of "how much" the content of a video segment is changing, it can be interpreted as a measure of the "entropy" of the video segment, and can be used as a salient feature $\mathcal{F}$ for summarization purposes.

4.2. Presence of Human Faces. Since a user can be interested in privileging in the final skim the presence of shots containing human faces, for example, for visualising excerpts from dramatic movies, it is possible to define, for each shot, the salient feature $\mathcal{F}$ as the percentage of frames in the shot which contain at least one human face, subjected to a minimal dimension. In the actual implementation, the work by Viola and Jones described in [33] has been preferred to other face detection methods, but no restriction to other procedures is imposed. A possible extension of this salient feature related to dramatic movies is the integration of the information related to the human presence with the information related to the type of audio, for example, by detecting speech during dialogues.

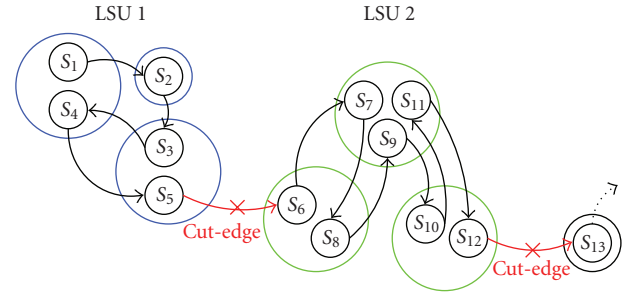


Figure 2: Detection of cut edges in a Scene Transition Graph. Since each cut edges is a sort of one-way transition from a highly connected group of clusters to another group, this can be considered as a reliable *LSU* boundary.

5. Modelling LSU with Hidden Markov Models

5.1. Logical Story Units Structure. In [30] it is shown how a video can be represented by a *Scene Transition Graph (STG)*, whose nodes are clusters of visually similar and temporally close shots, while edges between nodes stand for the transitions between subsequent shots. In the same work, the authors demonstrate that after the removal of cut-edges, that is, the edges which, if removed, lead to the decomposition of the *STG* into two disconnected subgraphs, each well connected subgraph represents a *Logical Story Unit (LSU)*, as shown in Figure 2.

In fact, since cut edges are one-way transitions from one set of clusters which are highly connected among each other (i.e., nodes connected by cycles in the corresponding indirect graph, see [30]) to another set of clusters characterized by a completely new visual-content, cut edges can be then considered as reliable *LSU* boundaries.

The *STG* has been computed on the base of an *LSU* segmentation obtained as in [31]. On the shot level, any existing technique for shot boundary detection can be employed, without loss of generality. However, the algorithm here employed adopts the classical twin comparison method, where the error signal used to detect transitions is based on statistical modeling [34].

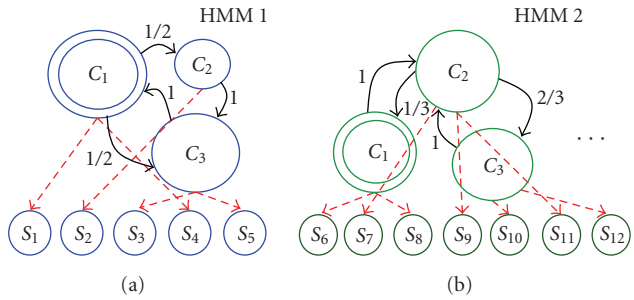


FIGURE 3: *LSUs* of Figure 2 are equivalently modeled by *HMMs*. States $\{C_i\}$ correspond to distinct nodes of the *STG* subgraph; transition probabilities are computed according to the relative frequency of the transitions between clusters, and shots $\{S_i\}$ are the possible observation set. The *HMM* initial states have been indicated with a double circle.

Starting from the *STG* representation, each *LSU* can be equivalently modeled by an *HMM*. This is a discrete state-space stochastic model which works well for temporally correlated data streams, where the observations are a probabilistic function of a hidden state [4]. Such a modelling choice is supported by the following considerations [7]:

(1) Video structure can be described as a discrete state-space, where each state is a conveyed *concept* (e.g., "*man face*") and each state-transition is given by a change of concept;

(2) The *observations* of concepts are stochastic since video segments seldom have identical raw features even if they represent the same concept (e.g., more shots showing the same "*man face*" from slightly different angles);

(3) The sequence of concepts is highly correlated in time, especially for scripted-content videos (movies, etc.) due to the presence of editing effects and typical shot patterns inside scenes (i.e., dialogues, progressive scenes, etc.).

For our aims, *HMM* states representing concepts will correspond to distinct clusters of visually similar shots (where clusters are obtained as described in [9]); state transition probability distribution will capture the shot pattern structure of the *LSU*, and shots will constitute the observation set (as shown in Figure 3).

5.2. HMM Definition. We now define how the *HMM* is built, and then how the models generate observation sequences in order to produce the video skim. Formally, an *HMM* representing an *LSU* is specified by the following.

(i) N, the Number of States. Although the states are hidden, in practical applications there is often some physical significance associated to the states. In this case, we define that each state corresponds to a distinct node of an *STG* subgraph: each state is one of the N clusters of the *LSU* containing a number of visually similar and temporally close shots. We denote states as $C = \{C_1, C_2, \ldots, C_N\}$, and the state at time t as q_t.

(ii) M, the Number of Distinct Observation Symbols. The observation symbols correspond to the output of the system being modeled. In this case, each observation symbol $S = \{S_1, S_2, \ldots, S_M\}$ is one of the M shots of the video.

(iii) $\Delta = \{\delta_{ij}\}$, *the State Transition Probability Distribution:*

$$\delta_{ij} = P\left[q_{t+1} = C_j \mid q_t = C_i\right], \quad 1 \le i,\ j \le N. \tag{1}$$

Transition probabilities are computed as the relative frequency of transitions between clusters in the *STG*, that is, δ_{ij} is given by the ratio of the number of edges going from cluster C_i to C_j to the total number of edges departing from C_i. In a *HMM*, states can be interconnected in such a way that any state can be reached from any other state (e.g., an ergodic model); for this special case, we would have $\delta_{ij} > 0$ for all (i, j). However, since in our case the interconnections of states are given by the transitions from shot to shot, and not all clusters are interconnected with all the others, this usually makes the model a nonergodic one; in this case it is likely that we have $\delta_{ij} = 0$ for one or more (i, j) pairs.

(iv) $\Sigma = \{\sigma_j(k)\}$, *the Observation Symbol Distribution,* where

$$\sigma_j(k) = P\left[S_k \text{ at } t \mid q_t = C_j\right], \quad 1 \le j \le N,\ 1 \le k \le M \tag{2}$$

We define the observation symbol probability in state C_j, that is, $\sigma_j(k)$, as the ratio of the salient feature value in the shot S_k to the total value of the salient feature of the cluster that contains S_k. It represents the probability for the shot S_k of being chosen as observation of the related visual concept, and it is defined as

$$\sigma_j(k) = \begin{cases} \dfrac{\mathcal{F}(S_k)}{\mathcal{F}\left(C_j\right)} & \text{if } S_k \in C_j \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathcal{F}(C_j)$ is defined as the sum of all the salient feature values of the shots belonging to cluster C_j, that is, $\mathcal{F}(C_j) = \sum_{S_h \in C_j} \mathcal{F}(S_h)$. Conversely, if the shot does not belong to the cluster, its observation probability is null, so that it cannot be selected to represent the cluster visual concept.

(v) $\pi = \{\pi_i\}$, *the Initial State Distribution,* where

$$\pi_i = P[q_1 = C_i], \quad 1 \le 1 \le N. \tag{4}$$

In order to preserve the information about the entry point of each *LSU*, $\pi_i = 1$ if the cluster C_i contains the first shot of the *LSU*, otherwise $\pi_i = 0$.

Therefore, a complete specification of the *HMM* requires two model parameters (N and M), the observation symbols S, and the probability distributions Δ, Σ, and π. Since the set $S = \{S_1, S_2, \ldots, S_M\}$ is common to all the *HMMs*, for convenience, we can use the compact notation $\Gamma = (\Delta, \Sigma, \pi, N)$ to indicate the complete parameter set of the *HMM* representing an *LSU*.

6. Stochastic Skim Generation

In order to generate an informative skim, the following solutions have been adopted to fulfill all the skimming criteria stated in Section 3.

(i) Coverage. Since the skim should include all the semantically important story units, each detected *LSU* participates to the final synopsis. As a general remark, please notice that the skim ratio r should be subject to a minimal value $r_{\min}$ for the skim to be representative of the movie structure and content.

(ii) Representativeness. Let $l(\Lambda_1), l(\Lambda_2), \ldots, l(\Lambda_n)$ be the lengths of the n *LSUs* that compose the original video. Then in the skim, for each *LSU*, a time slot of length $l(\lambda_i)$ is reserved, where

$$l(\lambda_i) = r \cdot l(\Lambda_i), \quad \forall i = 1, \ldots, n. \tag{5}$$

(iii) Structure Informativeness. In order to include in the synopsis the information conveyed by the shot patterns inside the story units, the following procedure is adopted. Since the state transition probability distribution of *HMM* Γ_i has statistically captured the structure of the transitions between shots inside the corresponding *LSU* Λ_i, a skimmed version λ_i of the *LSU* can be generated as an observation sequence of the associated *HMM*, Λ_i, that is:

$$\lambda_i = O_1 O_2 \cdots, \tag{6}$$

where each observation O_j is one of the symbols from S, that is, a shot of the original video.

Starting from the first Hidden Markov Model Γ_1, the sequence λ_i is generated as follows:

(1) choose the initial state $q_1 = C_h$ according to the initial state distribution π. Set $t = 1$;

(2) while (total length of already concatenated shots) < (time slot $l(\lambda_i)$ assigned to the current *LSU*),

   (a) choose $O_t = S_k$ according to the symbol probability distribution in state C_h, that is, $\sigma_h(k)$;

   (b) transit to a new state $q_{t+1} = C_j$, according to the state transition probability for state C_h, that is, δ_{hj};

   (c) set $t = t + 1$;

(3) repeat the previous steps for all Γ_i.

The above described procedure means that the generated skim for each *LSU* is one of the possible realizations of the stochastic process described by the corresponding *HMM*, where both interstate transitions and the shot selections are the results of random trials. In order to generate the whole skim, this method is applied to all Logical Story Units, and the obtained sequences of observed shots for each *LSU* are concatenated in the final synopsis.

Since the skim generation does not take into account the original shot order inside a story unit, it may happen that in the skim a shot which is later in the original *LSU* can appear before another one which is actually prior to it (as it sometimes happens in commercial trailers). In the circumstance of "anticasual" shots, they are repositioned in causal order inside each *LSU*, without altering the nature of the algorithm. The shot repositioning is automatically performed on the basis of the shot identifiers (corresponding to the shot positions in the movie), while shots belonging to different *LSUs* are already in casual order by construction.

(iv) Content Informativeness. In order to privilege more "informative" shots, the observation symbol probability distribution Σ depends on the presence of the salient feature. In particular, the higher is the value of the salient feature in a shot S_k of the cluster C_j, the higher will be $\sigma_j(k)$, that is, S_k will be more likely chosen for the skim.

For example, regarding action movies and the salient feature related to motion activity, by assigning higher probability of observation to more dynamic shots, we privilege "informative" segments for the skim generation. At the same time, we avoid to discard *a-priori* low-motion shots, that can be chosen as well for entering the skim, even if with lower probability. Moreover, once that one shot is chosen for the video skim, it is removed from the list of candidates for further time slots, and the observation symbol distribution is recomputed on remaining shots in the cluster. This prevents the same shot from repetitively appearing in the same synopsis, and at the same time allows also low-motion shots to enter the skim, if the user-defined skim ratio is large enough. Therefore, as it should be natural, in very short skims, "informative" shots are likely to appear first, while for longer skims, even less "informative" shots can enter the skim later on.

7. User Tests and Performance Evaluation

To quantitatively investigate the performance of the proposed method for video skimming, we carried out two main experiments using the feature films in Tables 1 and 3.

7.1. User Test A: Informativeness and Enjoyability. In this first test, for the evaluation of the skims, the method and the criteria of "*informativeness*" and "*enjoyability*" adopted in [2] have been used. *Informativeness* assesses the capability of the statistical model of maintaining content, coverage, representativeness, and structure, while reducing redundancy. *Enjoyability* assesses the performance of the salient feature employed in selecting perceptually enjoyable video segments for the skim. Starting from the *LSU* segmentation results, as generated in [31], we produced 20 dynamic summaries with their related soundtracks: for each video in Table 1, two associated skims have been produced, one with 10% of the original video length and the other with the 25%. The salient feature related to motion activity was used for 5 movies (no. 1, no. 4, no. 7, no. 8, no. 9) which are closer to the genre

Table 1: Set of feature films (TEST A).

No.	Video
1	A Portoguese farewell
2	Notting Hill
3	A beautiful mind
4	Pulp fiction
5	Camilo and Filho
6	Riscos
7	Altrimenti ci arrabbiamo
8	Don Quixotte
9	Più forte ragazzi
10	Don Camillo

"action movie", while the presence of faces was adopted for the other ones closer to the genre "drama".

A first set of 12 students (6 male, 6 female) assessed the quality of the produced skims by watching, for each movie, one randomly selected version among the available three: 10%, 25%, and the original movie 100%. Before starting the test, the participants were given a short oral introduction about the idea of automatic video skim generation and on the purpose of the test. After watching the selected version, each student assigned two scores ranging from 0 to 100, in terms of *informativeness* and *enjoyability*, also in case they watched the original movie if they thought that this was not 100% enjoyable or informative (e.g., when an intricate plot determines that the movie is not completely informative regarding situations and displayed events).

Table 2 shows the obtained average scores which have been normalized by the score assigned to the original movie.

In these experiments, average normalized scores for *enjoyability* are around 72% and 80% for video skims of 10% and 25% length, respectively. Regarding *informativeness*, average normalized scores are around 69% and 81%, respectively. These results are comparable with the ones presented in one of the most referenced works on video skims [2]. However, results presented here have been obtained on a larger set of videos, in particular on movies coming from different genres.

7.2. User Test B: Utility and Comparison. Based on the user test A only, it is not possible to completely assess the utility of the generated skim, nor to evaluate the algorithm performance with respect to other solutions. For this reason, another user set of 12 students (6 male, 6 female) were hired for performing a more severe test on another set of 10 movies (in Table 3) concerning the skim utility in a modern multimedia management system.

It is nowadays believed by broadcasters and content producers that a skim of a movie would be helpful for the user to assess whether he/she would be interested in paying to watch the entire movie. Of course the skim should not reveal too much about the movie plot, for example, being limited only to the introductory part of the film. A collection of movie skims could be offered as a preview on websites to be watched by users so that they can decide whether or not to download the whole movie.

Table 2: Performance evaluation of Video Skimming.

No.	Enjoyability			Informativeness		
	10%	25%	100%	10%	25%	100%
1	69.3	75.8	91.9	61.8	72.1	90.3
	75.4	**82.4**	**100**	**68.4**	**79.8**	**100**
2	62.8	70.5	86.2	65.4	75.8	93.1
	72.8	**81.8**	**100**	**70.3**	**81.4**	**100**
3	68.2	71.4	88.2	65.6	78.8	89.4
	77.2	**80.9**	**100**	**73.4**	**88.1**	**100**
4	57.5	67.2	84.6	63.3	72.9	91.5
	68.0	**79.4**	**100**	**69.1**	**79.6**	**100**
5	68.1	73.6	94.2	65.1	72.3	93.4
	72.3	**78.1**	**100**	**69.7**	**77.4**	**100**
6	55.2	68.8	93.0	64.0	78.1	94.0
	59.3	**73.9**	**100**	**68.1**	**83.0**	**100**
7	66.5	78.4	91.2	60.3	78.0	93.8
	72.9	**85.9**	**100**	**64.3**	**83.2**	**100**
8	69.5	74.9	90.5	60.2	72.1	89.5
	76.8	**82.7**	**100**	**67.2**	**80.5**	**100**
9	69.8	70.1	85.5	62.8	72.1	92.1
	81.6	**82.0**	**100**	**68.1**	**78.2**	**100**
10	65.8	68.1	95.3	65.6	71.2	93.0
	69.0	**71.4**	**100**	**70.5**	**76.5**	**100**
Aver.	72.5	79.8	—	68.9	80.8	—
Drop	27.5	20.2	—	31.1	19.2	—

With this scenario in mind, the user tests were performed according to the following procedure. From the introductory part (i.e., the first 30 minutes) of the 10 blockbuster movies in Table 3, three skims have been generated by three different methods with the same skim ratio $r = 0.1$, so that each skim is about 3 minutes long.

The participants were told that the video skims were automatically generated by algorithms and that the aim of the experiments was a comparison of three automatic video skim generation algorithms. After answering three questions about gender, age, and film watching behaviour, each user was requested to watch 10 randomly chosen skims, one per movie, without being aware of which algorithm was responsible for the creation of a particular skim.

The first adopted algorithm $\mathcal{A}$ generates a video skim by selecting video segments randomly from the original movie. The second algorithm $\mathcal{B}$ is the algorithm described in this work, based on the use of salient features and Hidden Markov Models, where *LSUs* were generated as in [31] and the salient feature related to motion activity was used for action movies no. 1, no. 2, no. 5, no. 9, and no. 10, while the presence of faces was adopted for the other ones closer to the genre "drama", according to their IMBd classification [35]. In Table 3, more details can be found about the movie shots, the number of *LSUs*, the skim length, and the number of visual concepts (i.e., the hidden states) used to generate the skim according to the proposed method.

Table 3: Set of feature films (TEST B) and details about the structure of movies and skims.

No.	Video	LSUs	Shots (30 min)	Shots (3 min)	Visual concepts
1	Raiders of the lost ark	13	353	35	91
2	Terminator	13	399	37	117
3	Gattaca	30	246	14	85
4	Donnie Darko	12	256	25	31
5	Finding Nemo	14	462	31	80
6	A Beautiful mind	15	353	40	39
7	Talk to her	10	168	24	41
8	Goodbye Lenin	20	499	47	91
9	Kill Bill vol 2	7	224	33	15
10	War of the worlds	15	280	30	74

Table 4: Questionnaire about the utility and quality of watched skims.

No.	Question
1	Is the skim useful for understanding the genre of the original movie? (1–5)
2	Is the skim able to give you a clear idea of the movie atmosphere? (1–5)
3	Is the skim able to give you a clear idea of the narration pace? (1–5)
4	Is the skim able to give you a clear idea about the involved characters? (1–5)
5	Is the skim useful for understanding whether you would/would not like to watch the entire movie? (1–5)
6	Please give the skim a global score. (1–5)

The third set of skims, generated with method $\mathcal{C}$, have been manually generated by a cinema lover and expert of editing systems, aiming at providing an overview of the storyline and a fair impression of the movie atmosphere.

We expect that skims generated using the *random* technique would be of lower quality than skims generated by our *HMM* approach. On the other side, *manually made* skims certainly represent an upper-bound for the overall quality of skims. Therefore, we assume that in terms of quality of results, the *random* method will be worse than the *HMM* approach, and *manually made* skims will have the highest possible quality.

After viewing each skim, participants were requested to fill out a questionnaire (as in Table 4) about the utility and the quality of the watched skim, and to mark each of the 6 questions on a Likert scale from 1 (min) to 5 (max). After answering, participants were also asked for detailed comments and whether they had already seen the original movie within the last 6 months, more than 6 months ago, only partially or never before, in order to accordingly weight their answers.

Results regarding the quality of the three compared skims are reported in Figure 4. It is evident that for all questions, the *manually made* skims stand out as better with respect to *HMM* and random. As expected, the *HMM* skims score on average better than the *random* method.

Table 5: Questionnaire about the informativeness of "Raiders of the lost ark".

No.	Question
1	Why does the room in the cave collapse?
2	What the main male character escape with?
3	What is the topic of the talk in the library?
4	What is the location of the last scene?

The informativeness of each skim was also investigated by asking the users 4 specific questions regarding the plot understanding of the original movie that can be inferred by watching the skimmed version. The proposed questions concern four main narrative key-points (judged by a human) which take place in the considered first 30 minutes of each movie (see, e.g., Table 5 with the questions related to the introductory part of the movie "Raiders of the lost ark").

Marks from 0 to 2 were given to wrong/missing, partially correct, and correct answers, respectively. Answers from users that declared that they have seen the movie before have been weighted accordingly.

Results regarding the informativeness of the three compared skims are reported in Figure 5. It is evident that for all questions regarding the level of understanding of the plot, the *manually made* skims stand out as better with respect to *HMM* and random. As expected, the *HMM* skims score on average better than the *random* approach.

Regarding the proposed methodology (algorithm $\mathcal{B}$), we consider $r = 0.1$ as the lowest skimming ratio that we can applied to a movie before producing a degenerate skim, that is, no more representative of the movie structure and content. For smaller values of r, in fact, the structure of some scenes would be completely lost. Therefore, we have produced our experiments testing the system in limit conditions, and we expect performance to be even better when bigger skimming ratios are applied.

Further analysis and discussion on obtained results are ongoing to critically revise results obtained for movies no. 4 and no. 10 whose results for both experiments in test B are not aligned with the rest of the films. In particular we plan to carefully analyse shooting scripts since we guess that,

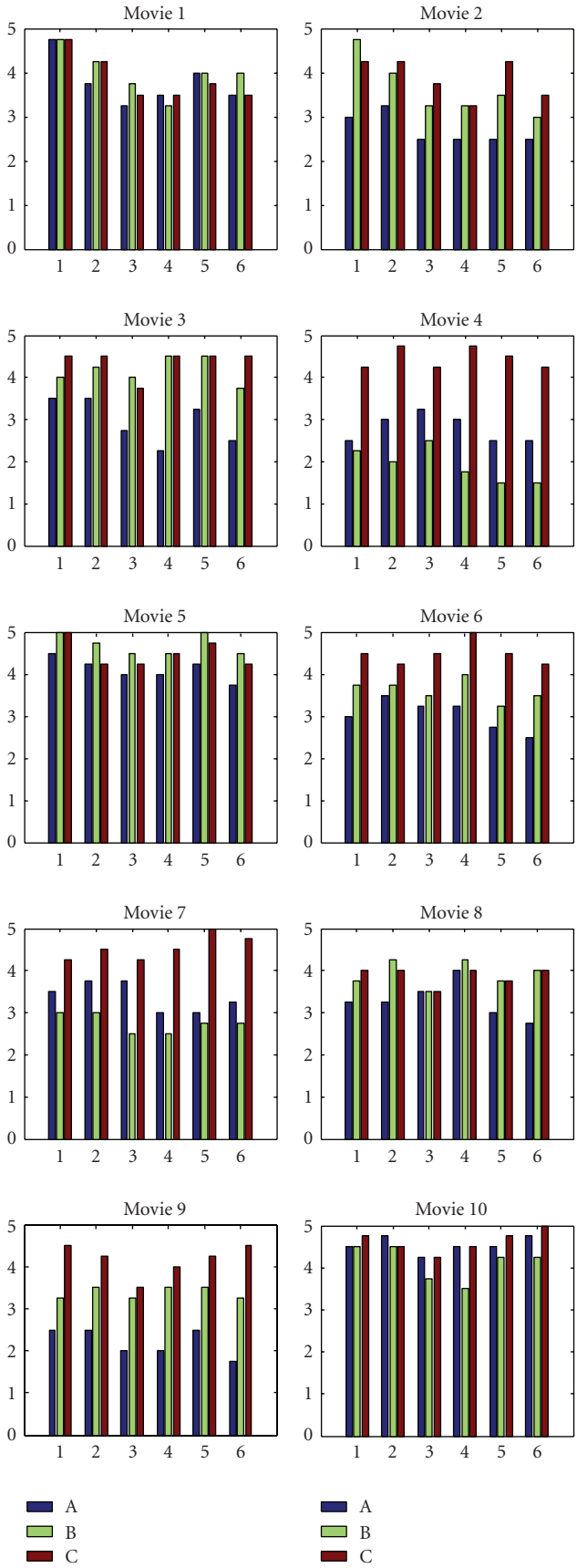


Figure 4: Average marks on skim quality (A = random sampling) (B = HMM and salient features) (C = manual).

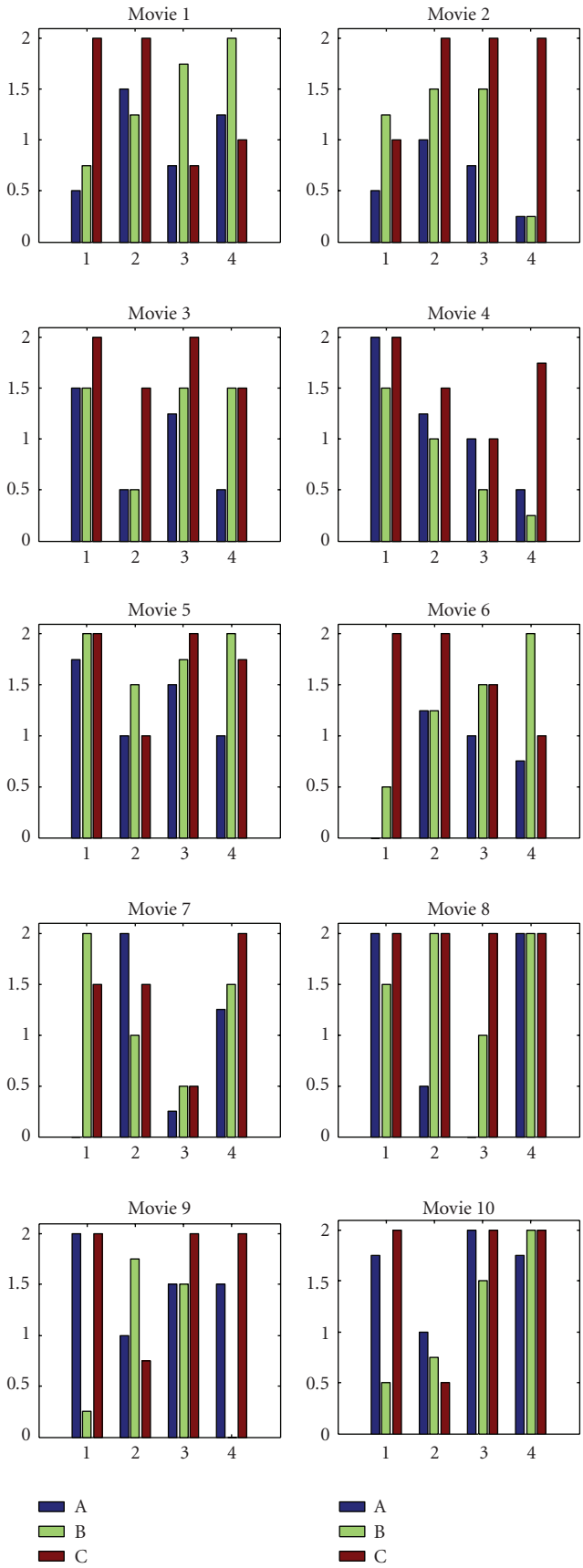


Figure 5: Average marks on plot understanding (A = random sampling) (B = HMM and salient features) (C = manual).

for both movies, misaligned results are probably due to the intricate plots (e.g., the use of flashbacks, the mixing of real scenes with ones taken from dreams, etc.) and the peculiar editing styles employed by both directors.

8. Conclusions

In this paper a method for video skim generation has been proposed. This technique is based on a previous high-level video segmentation and on the use of *HMMs*. The final skim is a sequence of shots which are obtained as observations of the *HMMs* corresponding to story units, and by a set of salient features which roughly estimates the informativeness of shots, depending on film genres. The effectiveness of the proposed solution has been compared and demonstrated in terms of informativeness and enjoyability on a large movie set coming from different genres. From the user study we can conclude that skims generated using the proposed method are not as good as manually skims, but have considerably higher quality than skims generated using a random sampling method.

Ongoing work aims at broadening the set of available salient features for different video genres, for example modifying the already described salient feature related to human faces according to the percentage of music/silence/speech inside each shot. The same salient feature based on audio classification could be useful to skim music programmes, for example to isolate songs from the other material in the show. Further applications of the proposed method to video mash-up are also envisaged and currently under investigation.

Acknowledgments

The authors would like to thank the reviewers for the accurate comments and PhD. student Luca Canini for priceless help during the experimental evaluation.

References

[1] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," in *Proceedings of the 10th ACM International Multimedia Conference and Exhibition*, pp. 533–542, Juan Les Pins, France, December 2002.

[2] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 296–304, 2005.

[3] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *Proceedings of the IEEE International Workshop on Content-Based Access Image Video Data Base*, pp. 61–67, January 1998.

[4] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[5] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, UK, 1999.

[6] Y. Wang, Z. Liu, and J. C. Huang, "Multimedia content analysis," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, 2000.

[7] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with hidden Markov models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*, vol. 4, pp. 4096–4099, Orlando, Fla, USA, May 2002.

[8] B. T. Truong and S. Venkatesh, "Video abstraction: a systematic review and classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 3, no. 1, p. 3, 2007.

[9] S. Benini, P. Migliorati, and R. Leonardi, "Hierarchical structuring of video previews by leading-cluster-analysis," *Signal, Image and Video Processing*, 2010.

[10] Y. Gao, W.-B. Wang, J.-H. Yong, and H.-J. Gu, "Dynamic video summarization using two-level redundancy detection," *Multimedia Tools and Applications*, vol. 42, no. 2, pp. 233–250, 2009.

[11] N. Omoigui, L. He, A. Gupta, J. Grudin, and Sanocki, "Time-compression: systems concerns, usage, and benefits," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 136–143, May 1999.

[12] A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, S. Srinivasan, and G. Cohen, "Using audio time scale modification for video browsing," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 3, pp. 3046–3055, January 2000.

[13] Z. Li, G. M. Schuster, and A. K. Katsaggelos, "Minmax optimal video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1245–1256, 2005.

[14] Y.-F. Ma and H.-J. Zhang, "A model of motion attention for video skimming," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '02)*, vol. 1, pp. 129–132, Rochester, NY, USA, September 2002.

[15] G. Evangelopoulos, A. Zlatintsi, G. Skoumas, et al., "Video event detection and summarization using audio, visual and text saliency," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09)*, pp. 3553–3556, Taipei, Taiwan, April 2009.

[16] J. Nam and A. T. Tewfik, "Video abstract of video," in *Proceedings of IEEE 3rd Workshop on Multimedia Signal Processing*, pp. 117–122, September 1999.

[17] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1280–1289, 1999.

[18] X. Zhu, X. Wu, J. Fan, A. K. Elmagarmid, and W. G. Aref, "Exploring video content structure for hierarchical summarization," *Multimedia Systems*, vol. 10, no. 2, pp. 98–115, 2004.

[19] Y. H. Gong and X. Liu, "Video summarization using singular value decomposition," in *Proceedings of the of International Conference on Computer Vision and Pattern Recognition (CVPR '00)*, vol. 2, pp. 174–180, 2000.

[20] "Rushes FP6-045189," http://www.rushes-project.eu/.

[21] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proceedings of the 8th ACM International Multimedia Conference and Exhibition (MIR '06)*, pp. 321–330, New York, NY, USA, 2006.

[22] E. Rossi, S. Benini, R. Leonardi, B. Mansencal, and J. Benois-Pineau, "Clustering of scene repeats for essential rushes preview," in *Proceedings of the 10th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '09)*, pp. 234–237, London, UK, May 2009.

[23] Q. Huang, Z. Liu, A. Rosenberg, D. Gibbon, and B. Shahraray, "Automated generation of news content hierarchy by integrating audio, video, and text information," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '99)*, vol. 6, pp. 3025–3028, March 1999.

[24] W.-T. Peng, Y.-H. Chiang, W.-T. Chu, et al., "Aesthetics-based automatic home video skimming system," in *Advances in Multimedia Modeling*, vol. 4903 of *Lecture Notes in Computer Science*, pp. 186–197, 2008.

[25] Y. Takahashi, N. Nitta, and N. Babaguchi, "Video summarization for large sports video archives," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 1170–1173, Amsterdam, The Netherlands, July 2005.

[26] H. Sundaram, L. Xie, and S.-F. Chang, "A utility framework for the automatic generation of audio-visual skims," in *Proceedings of the 10th ACM International Multimedia Conference and Exhibition*, pp. 189–198, Juan Les Pins, France, 2002.

[27] T. Tsoneva, M. Barbieri, and H. Weda, "Automated summarisation of narrative video on a semantic level," in *Proceedings of the IEEE International Conference on Semantic Computing (ICSC '07)*, Irvine, Calif, USA, September 2007.

[28] N. Dimitrova, M. Barbieri, and L. Agnihotri, "Movie-in-a-minute," in *Proceedings of the 5th IEEE Pacific-Rim Conference on Multimedia (PCM '04)*, Tokyo, Japan, December 2004.

[29] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, 1999.

[30] M. M. Yeung and B.-L. Yeo, "Time-constrained clustering for segmentation of video into story units," in *Proceedings of the 13th International Conference on Pattern Recognition (ICPR '96)*, vol. 3, pp. 375–380, Vienna, Austria, August 1996.

[31] S. Benini, A. Bianchetti, R. Leonardi, and P. Migliorati, "Video shot clustering and summarization through dendrograms," in *Proceedings of the Image Analysis for Multimedia Interactive Services (WIAMIS '06)*, pp. 19–21, Incheon, South Korea, April 2006.

[32] S. Jeannin and A. Divarakan, "MPEG7 visual motion descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 720–724, 2001.

[33] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 511–518, 2001.

[34] N. Adami and R. Leonardi, "Identification of editing effect in image sequences by statistical modelling," in *Proceedings of the Picture Coding Symposium (PCS '99)*, pp. 157–160, Portland, Ore, USA, April 1999.

[35] "Internet movie database," http://www.imdb.com/.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 864123, 9 pages
doi:10.1155/2010/864123

Research Article

An Optimized Dynamic Scene Change Detection Algorithm for H.264/AVC Encoded Video Sequences

Giorgio Rascioni, Susanna Spinsante, and Ennio Gambi

Università Politecnica delle Marche, Italy

Correspondence should be addressed to Susanna Spinsante, s.spinsante@univpm.it

Received 1 September 2009; Accepted 28 December 2009

Academic Editor: Ling Shao

Scene change detection plays an important role in a number of video applications, including video indexing, semantic features extraction, and, in general, pre- and post-processing operations. This paper deals with the design and performance evaluation of a dynamic scene change detector optimized for H.264/AVC encoded video sequences. The detector is based on a dynamic threshold that adaptively tracks different features of the video sequence, to increase the whole scheme accuracy in correctly locating true scene changes. The solution has been tested on suitable video sequences resembling real-world videos thanks to a number of different motion features, and has provided good performance without requiring an increase in decoder complexity. This is a valuable issue, considering the possible application of the proposed algorithm in post-processing operations, such as error concealment for video decoding in typical error prone video transmission environments, such as wireless networks.

1. Introduction

Scene change detection is an issue easy to solve for humans, but it becomes really complicated when it has to be performed automatically by a device, which usually requires complex algorithms and computations, involving a huge amount of operations. The process of scene change detection becomes more and more complex when other constraints and specific limitations, due to the peculiar environment of application, may be present. A scene in a movie, and, in general, in a video sequence, can be defined as a succession of individual shots semantically related, where a shot is intended as an uninterrupted segment of the video sequence, with static frames or continuous camera motion.

In the field of video processing, scene change detection can be applied either in preprocessing and postprocessing operations, according to the purposes that the detection phase has to achieve, and with different features and performance. As an example, in H.264/AVC video coding applications, scene change detection can be used in preprocessing as a decisional algorithm, in order to force Intraframe encoding (I) instead of temporal prediction (P), when a scene change occurs, and to confirm predicted or bi-predicted (B) coding for the remaining frames. As discussed in [1], a dynamic threshold model for real time scene change detection among consecutive frames may serve as a criterion for the selection of the compression method, as well as for the temporal prediction; it may also help to optimize rate control mechanisms at the encoder.

In lossy video transmission environments, the effects of the errors on the video presentation quality depend on the coding scheme and the possible error resilience strategy adopted at the encoder, on the network congestion status, and on the error concealment scheme eventually present at the decoder. In order to improve error resilience of the transmitted video signal, and stop error propagation during the decoding phase, Intra-picture refresh is usually adopted at the encoder, even if it is an expensive process, in terms of bit rate, as the temporal correlation among frames may not be exploited. In such conditions, a predictive picture refresh based on scene context reference picture, selected through a scene change detector, may ensure bit rate savings, while optimizing the choice of the refresh frames [2]. Scene cut detection may be also exploited to improve video coding,

attention based adaptive bit allocation, as presented in [3]. Scene cut detection is applied to extract frames in the vicinities of abrupt scene changes; visual saliency analysis on those frames and a visual attention-based adaptive bit allocation scheme are used to assign more bits to visually salient blocks, and fewer bits to visually less important blocks, thus improving the efficiency of the encoding process and the final quality of the compressed video sequence.

As previously introduced, besides being adopted in preprocessing operations, scene change detection may be usefully exploited in video postprocessing algorithms, such as in the context of error concealment of decoded video sequences affected by errors and losses. It is reasonable to expect that scene change detection at the decoder will have to face different conditions, with respect to scene change detection applied at the encoder. As an example, the input video sequence for a decoder could be the result of a video editing process, originating an encoded video stream with a lot of independent scene changes, as frequently happens in advertising videos. Moreover, the detector has to perform decisions and computations based on the available data, that may be missing or erroneous. H.264/AVC compressed video information is very sensitive to channel errors appearing during transmission. The adoption of Variable Length Coding (VLC) at the encoder side, together with more complex techniques like Motion Compensation, can lead to dramatic error propagation effects during decoding. Additionally, lost or damaged data cannot be retransmitted in real-time applications. As already discussed, error resilience techniques for compression, enhancing the robustness of the bitstream at the source coder, can be employed. They basically rely on adding extra parameters or more synchronization points at the encoder; however, this solution requires to change the encoding scheme and in some situations this is not possible, or not compatible with existing systems. Moreover, even if the bitstream is resilient to errors, errors may still occur. Hence, error concealment solutions at the decoder are usually preferred in most practical applications. When residual bit errors remain, error concealment approaches can conceal the error blocks by exploiting spatial and/or temporal correlation [4] of the correctly received data. Scene change detection algorithms may improve the performance of error concealment solutions, by allowing the selection of the proper spatial or temporal strategy. At the same time, the integration of a scene change detector in a real time concealment solution at the decoder poses strict constraints on complexity and computational time requirements. In a real world video, it is reasonable to expect that errors occurring at scene changes are less frequent than errors occurring in other pictures of the video sequence, basically because the number of scene change events is necessarily smaller than the total amount of frames in the sequence. However, besides being catastrophic for the decoding mechanism, errors at scene cuts can be really annoying to the viewer: the temporal correlation among two frames in a scene change is so insignificant that Intererror concealment also generates very poor results, and macroblocks that do not fit with the content of a frame can appear on the scene, disturbing the viewer's experience. As a consequence, errors at scene cuts should be avoided or compensated somehow, but conventional temporal error concealment strategies are inadequate for this case. Therefore, a suited scene change detector designed for real-time decoding of video signals can contribute to mitigate the effects of data losses at scene cuts, and to improve the final quality of the video sequence.

Several solutions may be implemented to provide scene change detection, differentiated on the basis of the target application, and the corresponding computational requirements. In the context of video storage and retrieval, it is reasonable to assume the possibility of performing an offline processing of the video sequence, that may allow for increased complexity and accuracy; in real time environments, strict requirements on available time and computing resources must be satisfied, thus determining the need for low-complexity solutions, anyway able to provide acceptable performance.

The remainder of this paper is structured as follows: a review of some previous works on real time scene change detection algorithms is provided in Section 2. The proposed detector is presented in Section 3, and its performance is discussed, by means of experimental evaluations, in Section 4. Finally, conclusions are given in Section 5.

2. Previous Work

Several solutions for scene change detection have been proposed in the literature, to be applied either at the video encoder or at the decoder.

Sastre et al. presented a low-complexity shot detection method for real time and low-bit rate video coding, in [5]. As clearly stated by the authors, the method is basically aimed at compression efficiency more than frame indexing or other purposes. The algorithm is based on Intra/Inter decision for each macroblock, during the encoding process, and on the use of two thresholds, a fixed one and an adaptive one. If the algorithm detects the first frame of a scene change, based on the fact that the number of Intra macroblocks used when encoding the frame as a P-frame overcomes the thresholds, the algorithm stops the encoding as a P-frame, and forces the Intraframe encoding. Shot changes represent the best choice to insert key frames in the video sequence: the next frames of the new shot may be then encoded via motion compensation and prediction, based on the first I-frame. Inserting the key frames in suited positions of the bitstream allows to obtain the best quality in the decoded stream, and to optimize the output bit rate.

The proposed algorithm relies on two basic thresholds expressed as a percentage of the total number of macroblocks in a coded picture. The fixed threshold is set to a high value, to ensure that any frame in which the number of Intra macroblocks (I-MB) exceeds the threshold is coded as an I-frame, independently of the rest of the algorithm's conditions. The second, adaptive threshold depends on the average number of I-MBs of all the pictures encoded since the last I-frame, and forces a frame to be coded as an I-frame if the number of I-MBs in it exceeds the average number of intra MBs of the previous frames, in a given

quantity. Several ideas implemented within this algorithm have been exploited also by the one proposed in this paper, that is described in the next Section. First of all, we use two different thresholds, a fixed one, expressed in terms of absolute number of macroblocks, and an adaptive one, used to sharpen the shot detection. A limit is also placed on the adaptive threshold, below the fixed one, to prevent a frame from being encoded as a P-frame when almost all of its macroblocks are Intracoded. A smoothing algorithm with memory is used to determine the average of Intra macroblocks of the previous frames within the shot, in order to avoid tracking the number of Intra macroblocks too fast, and to provide a stable value for the desired average. Finally, a span parameter is used to avoid shot detections too close in time: the span establishes a period of time after a shot detection, during which only the fixed threshold is active, and the adaptive threshold cannot cause the insertion of a key frame in the bitstream.

In [6], a pixel based-algorithm for abrupt scene change detection is presented. The algorithm requires a two-stage processing of the frames, before passing them to the H.264/AVC encoder. In the first stage, subsequent frames are tested against a dissimilarity metric, the Mean Absolute Frame Difference (MAFD):

$$\mathrm{MAFD}_n = \frac{1}{MN} \sum_{i=0}^{M-1N-1} \sum_{j=0} \left| f_n(i,j) - f_{n-1}(i,j) \right|, \quad (1)$$

which measures the degree of dissimilarity at every frame transition, with M and N being the width and height of the frames, $f_n(i,j)$ the pixel intensity at position (i,j) of the nth frame, and $f_{n-1}(i,j)$ the pixel intensity at the same position of frame $n-1$. Considering that most of the frames in a video sequence do not belong to scene changes, a quick frame skimming can be performed by such a metric. As a matter of fact, abrupt scene transitions produce a peak value in MAFD within a period of time, in contrast with normal motion of objects and camera in the scene, that usually causes a large MAFD signal over a period of time. In the second stage, the set of frames not previously discarded are normalized via a histogram equalization process, through a progressive refinement based on MAFD and other three metrics, applied on the normalized pixel values. The algorithm does not perform motion estimation but it only works on frame pixel values, thus avoiding high-computational costs. For this reason, it may be suitable for real-time video segmentation applications, and rate control. Experimental tests discussed by the authors show that the algorithm is efficient and robust in presence of abrupt scene changes, whereas it shows some limitations when gradual changes (such as dissolve and fade) or luminance variations (flickers) affect the video sequence. A combination of different metrics should be applied in those cases, in order to improve and refine the algorithm's detection capabilities.

A prominent reference for the scene change detector proposed in this paper is the scheme presented in [1], by Dimou et al.. The fundamental result is the definition of a Dynamic Threshold Model (DTM) that can efficiently trace scene changes, based on the use of an adaptive and dynamic threshold which reacts to the sequence features, and does not need to be calculated before the detection, and after the whole sequence is obtained. The method is based on the extraction of the Sum of Absolute Differences (SAD) between consecutive frames from the H.264 codec, that is then used to select the compression method and the temporal prediction to apply. The SAD defines a random variable, whose local statistical properties, such as mean value and standard deviation, are used to define a continuously updating automated threshold. Statistical properties are extracted over a sliding window, whose size is defined in terms of the number of frames over which the random variable is observed. The algorithm also applies a function-based lowering of the detection threshold, in order to avoid false detections immediately after a scene change. As a matter of fact, each time a scene change is detected, the SAD value of this frame is assigned to the threshold; for the following K frames, the threshold value is set according to an exponentially decaying law, with a suitably chosen parameter to control the speed of decaying.

Scene changes generate high SAD values that make them detectable. Given the classical SAD definition for the n-th frame,

$$\mathrm{SAD}_n = \sum_{i=0}^{M-1N-1} \sum_{j=0} \left| f_n(i,j) - f_{n-1}(i,j) \right|, \quad (2)$$

a random variable X_i is defined, which models the SAD value in frame i. A sliding window of length K, with respect to the nth frame, is defined as the subset of frames whose index lies in $[n-(K+1), n-1]$. Over the sliding window, the empirical mean value m_n and the standard deviation σ_n of X_i are computed as follows:

$$m_n = \frac{1}{K} \cdot \sum_{i=n-K-1}^{n-1} X_i, \quad (3)$$

$$\sigma_n = \sqrt{\frac{1}{K-1} \cdot \sum_{i=n-K-1}^{n-1} (X_i - m_n)^2}. \quad (4)$$

Both (3) and (4), together with X_{n-1}, are used to define threshold $T(n)$ as follows:

$$T(n) = a \cdot X_{n-1} + b \cdot m_n + c \cdot \sigma_n, \quad (5)$$

where n denotes the current frame, and a, b,,and c are constant coefficients. The algorithm's performance is strongly related to the proper selection of the values assigned to constants a, b, and c: not only may they determine better or worse detection rates, but they must also be tailored to the application context, which means they will have different values if used at the encoding or decoding stage. Constant a rules the way threshold $T(n)$ follows the evolution of the random variable X_i: it is suggested to keep the value of a small, as many factors different from true scene changes can cause the rapid variation of X_i, and could consequently affect the correct detection. Constant b, on its turn, gives different weight to the average SAD computed over the

sliding window: if b takes high values, the threshold becomes more rigid and does not approach the X_i sequence. This avoids wrong change detection in presence of intense motion scenes, but, on the other hand, can also cause some missed detections, in presence of difficult scene changes featuring low SAD values. As σ_n is the standard deviation of variable X_i, high values of constant c prevent detecting intense motion events as scene changes. From this brief discussion, it is evident that the selection of a, b, and c is a hot point, and only a good tradeoff according to the target application can ensure proper functioning of the whole algorithm. Once a scene change has been detected in the pth frame, threshold $T(n)$ assumes the value of the SAD computed over the last frame. In order to avoid false detections immediately after a scene change, the threshold to use for the successive frames is forced to decay exponentially, according to the following law:

$$T_e(n) = X_{n-1} \cdot \exp^{-s(n-p)}, \tag{6}$$

where parameter s controls the speed of decaying. In experimental tests reported by the authors, constants a, b, and c were empirically chosen; the sliding window size was set to 20 frames, and the decaying parameter equal to 0.02. Remarkable improvements can be obtained by the algorithm, when compared to a scene change detector based on an optimal fixed threshold, chosen after having computed the SAD over all the frames, and *manually* identified the true scene changes.

3. The Proposed Scene Change Detection Algorithm

The object of this paper is to present a robust scene change detector, based on an improved version of the DTM discussed in the previous Section, but aimed at being applied in the different context of postprocessing applications, as in the case of an error concealment framework for H.264/AVC decoders. As a consequence, besides the strict requirements on low-complexity and real-time capability, the algorithm should be able to detect incorrelation between consecutive frames, that is, scene changes, even when applied to a corrupted bitstream, where the information needed to reveal a scene cut may be missing or not complete, due to errors and losses happened during video transmission. Besides that, the algorithm cannot rely on information about future frames to locate scene change events (as, on the contrary, it may happen in applications addressing the encoder side), and, considering the target application context of concealment at the decoder, it is important to design a detector able to locate changes affecting parts of the frame content, and not only the whole scene.

In encoded video streams of the YUV color space, the SAD computation may be performed on each single color component. However, considering a YUV 4:2:0 stream, it is obvious that the luminance component Y carries the greatest amount of information, so that SAD computation can be executed on the Y component only. Besides that, the luminace component is the one the human visual system is most sensitive to. In the proposed detector, a random variable X_i is defined as the average number of pixels per MB for which the SAD value is greater than 30. It is important to note that, being the random variable defined over frames affected by errors and losses, the average number of pixels is computed with respect to all the correctly received MBs shared (i.e. co-located) between consecutive frames. The threshold value of 30 has been set empirically, by observing that in case of a scene change, it is highly probable that colocated pixels have an absolute difference value greater than the threshold chosen. The dissimilarity measure provided by X_i seems more reasonable than a *pure* SAD metric in a context of possibly missing information; however, misbehaviours may still be present and are to be faced by proper adjustments.

As a first condition to consider, given the fact that the dissimilarity metric adopted is defined on the basis of the Y component only, it is clear that it will show a marked sensitivity to rapid variations in the luminance content of the scene, even if not due to a real scene cut. Looking at Figure 1, flashing lights produce a rapid increase in the luminance level of consecutive frames, even if no scene change has happened at all. In these situations, the metric previously defined could reveal a false scene cut, so that a proper correcting action is to be applied.

In order to avoid false scene cut detection, during the decoding phase, and before computing the dissimilarity metric value, a second parameter is computed, named ΔY, defined as the difference between the average value of the MB luminance of two consecutive frames. The MBs included in the computation may be not the co-located ones in the two frames, given the possible losses during video signal transmission. The positive or negative ΔY value is subtracted to the dissimilarity value obtained by the SAD-based computation, to get the final metric. The curves reported in Figure 2 show how this simple modification may improve the reliability of the dissimilarity metric: peaks in the average Y value per MB curve ($\langle Y \rangle$) correspond to flashing light events in the video sequence and are obviously revealed by associated couples of peaks in the value assumed by X_i (there are 2 peaks in X_i for each peak in $\langle Y \rangle$, as a flashing light event affects two consecutive frames). The modified metric curve maintains the correct location of peak couples, but avoids a false scene cut detection, by properly lowering the resulting dissimilarity measure, with respect to the unmodified metric curve.

A second modification to the original scene cut detector inspiring this work is motivated by the target application context of error concealment solutions at the decoder. In view of concealment operations possibly performed on the same frames analyzed by the scene cut detector, it may be useful to collect, through the application of the detector, information related not only to global changes affecting the whole frames, but also referred to parts of the frame, on a local scale. This *granular* information may be possibly exploited to identify parts of the frame where Intraconcealment could be more suitable than Inter, because of local changes, even if the frame under processing is temporally related to the previous one, and so could claim for a global Inter concealment. An example of this possible situation is

Frame ($n-1$) (a)

Frame (n) (b)

Figure 1: Luminance variation between consecutive frames due to flashing lights, with no scene change event.

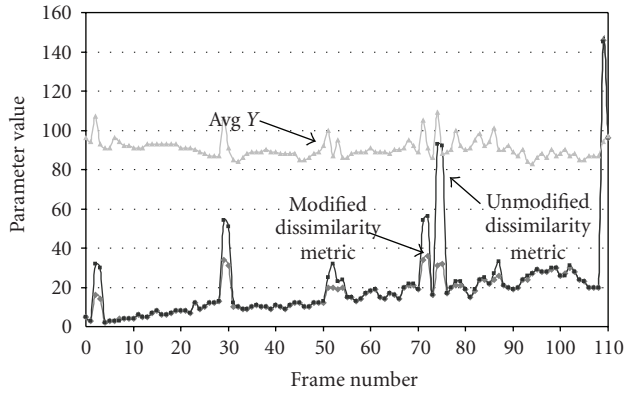


Figure 2: Variations of the average Y per MB, unmodified dissimilarity metric, and modified dissimilarity metric by ΔY parameter, for a test video sequence, due to flashing light events.

shown in Figure 3: comparing the two consecutive frames, it is clear that a scene cut does not take place, however, part of the background changes substantially. If several MBs get lost in the background area evidenced on the figure, an Inter concealment algorithm based on temporal correlation would fail in properly restoring the scene, whereas an Intra, spatial-based concealment, could be effective. Availability of such an information about local changes in the frame could enable an adaptive concealment strategy, based on differentiating the recovery technique on a group-of-MBs level.

In order to collect local scale information about the frame content, each frame has been virtually divided into macro areas, the number of which depends on the frame format; for CIF frames, 20 areas are located. By this way, each area includes 4×4 MBs, with the exception of the edge areas where the number of MBs may be 4×5 or 5×5, as shown in Figure 4. By such a virtual chessboard pattern, it is possible to track useful local information about the frame, even if not on a pixel basis, which would require unacceptable storage resources. At the decoder, a memory buffer is defined, whose elements are indexed according with the label associated to each macro area; each buffer element, in its turn, stores the average dissimilarity value evaluated over the specific macro area identified by the element index.

Besides being useful in the case of subsequent concealment operations, the virtual frame partition may help in correctly revealing a true scene cut, with respect to variations in the content which could affect most of the frame, without anyway representing a true scene change. As a matter of fact, if a true scene cut takes place, evident variations in the dissimilarity value will affect all the macro areas, and not only a limited subset of them. According to such a reasoning, a further decision step is included in the detector: once having computed the average and median dissimilarity values over the virtual partition buffer, if they both result in greater than an empirically set value of 100, the dissimilarity measure is increased by 20%; otherwise, if both the values are lower than 80, the dissimilarity metric X_i is reduced by the same percent value. Figure 5 highlights the effects of such a modification on the behaviour of the dissimilarity metric X_i: possible true scene cuts are emphasized by the modified metric, thus permitting their correct detection, whereas possible false cuts are minimized, to reduce the probability of an erroneous detection.

Information collected by the virtual frame partition process, as said before, may be exploited to analyze the local dynamic evolution of a frame. As shown in Figure 6(b), the average dissimilarity values for each macro area denote a change in the central part of the frame, which, however, is not due to a true scene cut, as many of the edge areas show a zero value. In the specific case reported, the algorithm provides an average value of 30.2, and a median value of 17: consequently, by lowering the dissimilarity value by a 20% amount, the risk of false cut detection is avoided.

The last modification added to the dynamic detector is conceived to face the case of high-motion scenes, as, for example, in the case of panning effects of the video camera. These situations show a typical effect over the set of frames included in the observation window (i.e., the sliding window cited in previous section), which spreads over 5 frames in the proposed scheme: given the threshold definition in (5),

Frame $(n-1)$ (a) Frame (n) (b)

Figure 3: Two sample consecutive frames with local changes in the background, but no scene change.

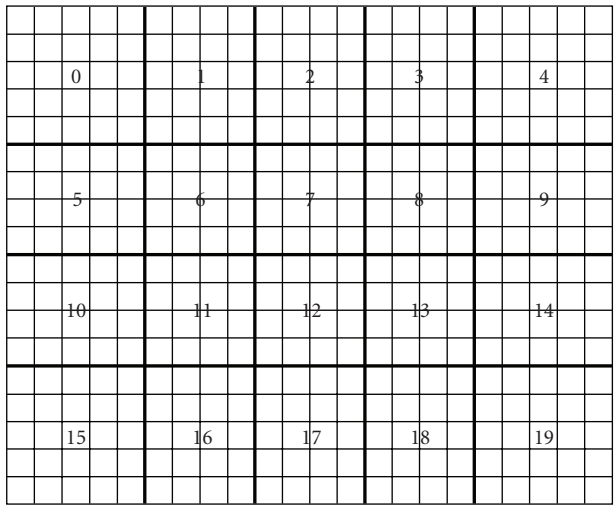


Figure 4: Virtual frame partition.

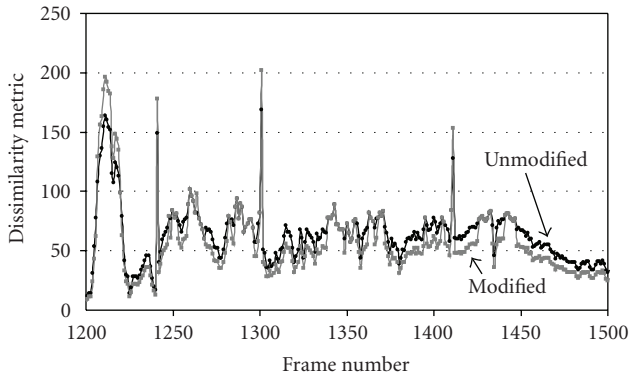


Figure 5: Performance of unmodified and modified dissimilarity metric exploiting local scale information provided by virtual frame partition.

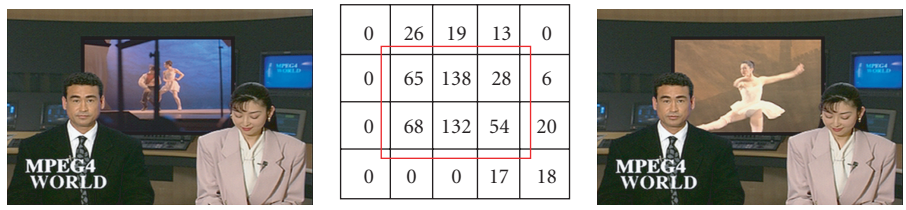
high values of the parameter m_n and low values of σ_n are jointly observed. In such situations, the probability of a false scene cut detection may be very high; consequently, the dissimilarity value is forced to decrease by a 20% amount, when $m_n > 80$ and $\sigma_n < 10$. These specific thresholds have been derived by extensive empirical tests over different video sequences. Figure 7 shows the behaviour of the modified dissimilarity metric in presence of a high-motion video sequence. It is important to notice that the motion degree of a video sequence, besides being an intrinsic property of the sequence itself, is also influenced by the frame rate set at the encoder. If a YUV sequence encoded at 25 ÷ 30 fps is decimated by a coefficient of 2 or 3, the final effect is to increase the motion degree of the decimated video sequence; this is an issue to take into account, as frame decimation is a typical operation performed on video sequences in order to reduce their bit rate and allow transmissions over limited bandwidth channels (i.e., wireless systems).

Table 1: Dynamic variation conditions for the threshold coefficients.

Condition	Value
Default	$a = -0.4$
	$b = 1.7$
	$c = 2$
$0 < \sigma_n \leq 5$	$c = 10$
$5 < \sigma_n \leq 10$	$c = 5$
$10 < \sigma_n < 30$	$c = 3$
$T(n) < 70$	$T(n) = 70$

Further tuning operations in the detection algorithm involve the a, b, and c coefficients defining the detection threshold (5). Imposing an adaptive and dynamic variation of these coefficients adds flexibility to the detection threshold, thus maintaining its effectiveness for a correct scene cut detection. Variations applied on coefficients a, b, and c, and extracted by empirical observations over many different video sequences are summarized in Table 1. Besides that, in order to avoid false detections, as soon as the dissimilarity value obtained for a true scene cut goes out from the sliding window, a correcting action, named *lowering condition*, is applied, by comparing the value of X_i to the value given by $(m_n + \sigma_n + X_i/2)$ and taking the lowest one, as the new value for scene cut detection.

Figure 8 shows the behaviour of the dissimilarity measure X_i, and of the threshold $T(n)$ used for scene cut detection,



(a) (b) (c)

Figure 6: Local frame dynamics evidenced by virtual frame partition.

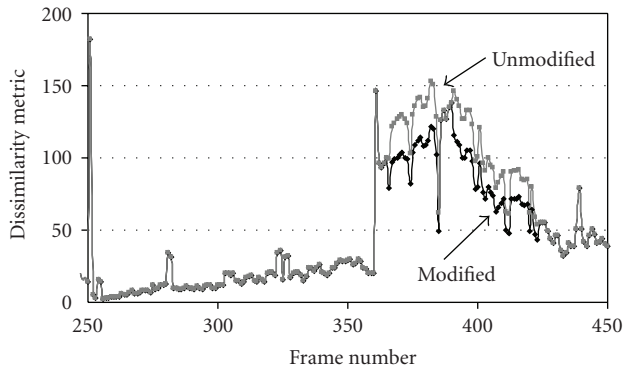


Figure 7: Improvement of the modified dissimilarity metric behaviour in the case of high-motion video sequences.

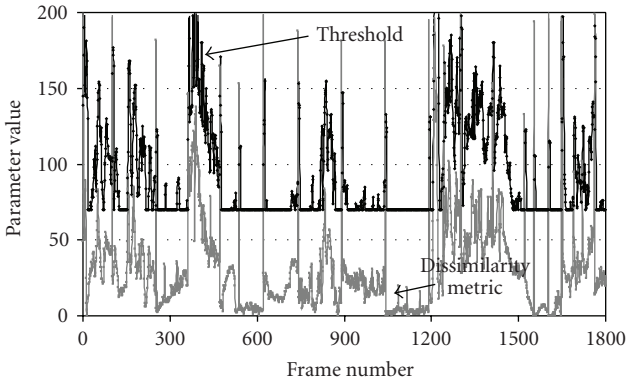


Figure 8: Threshold $T(n)$ and dissimilarity metric X_i variations over a whole 12.5 fps video sequence with no losses.

over a test video sequence of 12.5 fps frame rate, with no losses. The dynamics obtained by modifying the detection algorithm, according to the solutions described above, allow to adaptively change the detection threshold in order to increase the correct detection rate and reduce the false or missed detections.

Before moving to the experimental evaluation of the proposed detector, as discussed in the next Section, Figure 9 summarizes the detector main components and the data processing flow in a block diagram fashion.

4. Experimental Evaluation and Results

As a preliminary evaluation, the proposed algorithm has been compared to other scene cut detection solutions, by the application of the MSU Video Quality Measurement Tool [7], which is able to implement four different similarity metrics, defined as follows:

(1) *Pixel-Level Comparison*: the similarity measure of two frames is the SAD computed over the intensity values of corresponding pixels;

(2) *Global Histogram*: the histogram is obtained by counting the number of pixels in the frame, with specified luminance level. The difference between two histograms is then determined by calculating the SAD over the pixels having the same luminance level;

(3) *Block-Based Histogram*: each frame is divided into 16×16 pixels blocks. For each block, a luminance distribution histogram is constructed, the similarity measure for each block is obtained, and the average value of these measures is accepted as the frame similarity measure as the frame similarity measure;

(4) *Motion-Based Similarity Measure*: a Motion Estimation algorithm with block size 16×16 pixels is applied on adjacent frames. The average value of the Motion Vector errors is accepted as the similarity value.

The MSU tool has been applied offline, and comparisons with the proposed detector were performed on a 12.5 fps CIF video sequence in YUV format, not affected by losses, showing 38 true scene changes located by visual inspection. The test video sequence has been generated by composition of 29 subsequences collected from the Video Quality Expert Group repository, in order to include as many different effects as possible, such has low and high motion, panning, zooming, light variations, scene changes, and so on.

Results presented in Table 2 confirm the effectiveness of the proposed detector: besides being able to provide a local-scale information about the sequence dynamics, which is not provided by the MSU software tool, the proposed detector has been designed to process sequences affected by losses, as reported in the following discussion.

In order to evaluate the performance of the proposed scene cut detection algorithm in presence of losses, tests have been executed on H.264/AVC encoded video sequences, encapsulated according to the Real Time Protocol (RTP) packet format. Before applying the H.264/AVC reference

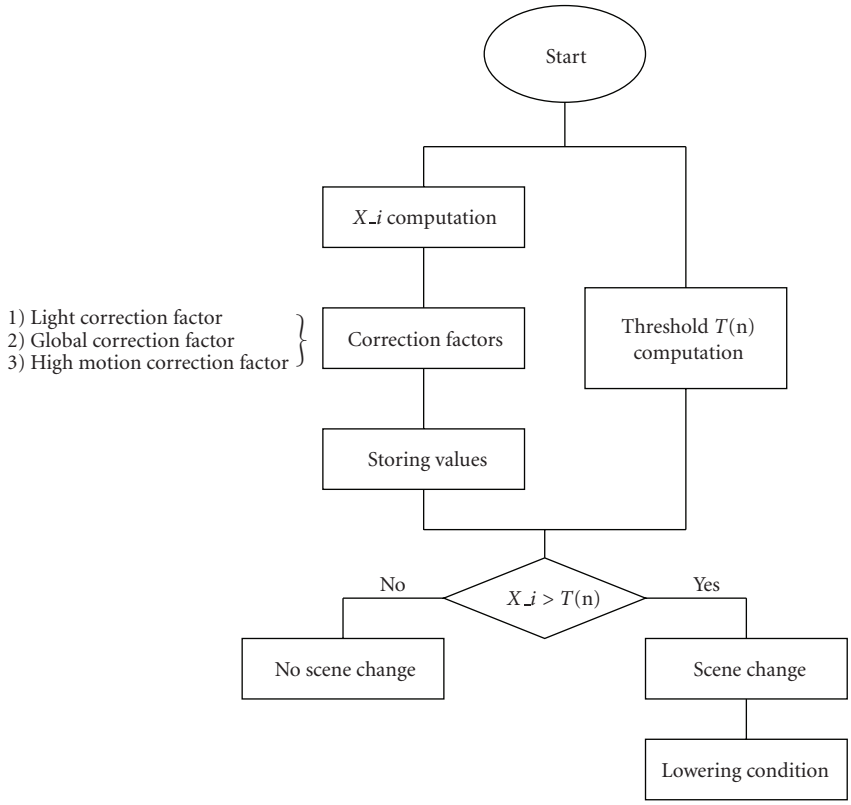


Figure 9: The scene change detector main components and processing flow.

Frame $(n-1)$ (a)

Frame (n) (b)

Figure 10: False scene change detection caused by lost MBs in the 12.5 fps sequence, for a 4% packet loss rate.

decoder properly modified to include the detector, H.264 encoded bitstreams have been subjected to a packet erasure process, in such a way as to simulate different packet loss rates, of 1%, 2%, 4%, and 10%, which may be considered representative of realistic environments, such as video transmission over packet-based wireless networks. For each packet loss rate value, 5 simulations over the same video bitstream have been executed, and the average result was considered, in order to account for different error patterns randomly generated. Simulations have been performed over sequences encoded at 25 fps, and over their decimated versions at 12.5 fps, in order to test the detector behaviour with respect to frame rate. Other main encoder parameters have been set as follows: the selected H.264 profile is Baseline, with a CIF, YUV 4 : 2 : 0 format, QPISlice = 28 and QPPSlice = 28.

The detector performance is defined with respect to two parameters, namely, the *Recall* (Re) and *Precision* (Pr) rates, that, in their turn, depend on the number of fake

Table 2: Performance comparison of the proposed detector and four different detection algorithms implemented by the MSU software tool, for a 12.5 fps CIF sequence with no losses.

Detection Algorithm	no. Correct Detections	no. False Detections
Proposed detector	38	0
MSU - 1	38	5
MSU - 2	37	13
MSU - 3	38	1
MSU - 4	38	3

Table 3: *Recall* and *Precision* average performance of the detector, for the same video sequence at 25 and 12.5 fps, and different packet loss rates.

Packet Loss Rate	25 fps		12.5 fps	
	Recall	Precision	Recall	Precision
No loss	1	1	1	1
1%	1	1	0.994	0.994
2%	0.994	1	0.976	1
4%	0.988	1	0.964	0.988
10%	0.982	0.994	0.952	0.966

detections (FD), the number of missed detections (MD), and the number of correct detections (CD) over a given sequence, as follows:

$$\text{Re} = \frac{\text{CD}}{\text{CD} + \text{MD}}, \qquad \text{Pr} = \frac{\text{CD}}{\text{CD} + \text{FD}}. \tag{7}$$

The test sequence adopted shows 38 true scene changes, revealed through visual inspection. Table 3 reports the *Recall* and *Precision* performance of the detector, for the same sequence at 25 fps and 12.5 fps, and for different packet loss rates; the values in the Table refer to average performance evaluated over 5 decoding iterations for each packet loss rate.

Results show a very satisfactory behaviour of the proposed detector, either at 25 and 12.5 fps, even if with a very small degradation in the latter case, with a *Recall* and a *Precision* figure always greater than 0.95. As reasonable and expected, performances degrade as the packet loss rate increases, according to the frame areas affected by data losses that may cause a false detection, or a missed one. Figure 10 shows a peculiar case for the 12.5 fps sequence at a 4% packet loss rate: missing MBs in the frame (represented as green MBs), due to packet losses, cause a variation in the dissimilarity metric which determines a false scene change detection. If losses do not occur, the detector correctly does not reveal any scene change, despite the evident variation of the frame in its bottom areas.

5. Conclusion

This paper presented an optimized scene change detector for H.264/AVC video sequences, based on a dynamic threshold model properly designed to be applied at the decoder side, even in presence of losses and errors in the received bitstreams. On the contrary, most of the detection algorithms presented in the previous literature are conceived for application at the encoder side, and cannot deal with data losses in the video bitstreams. The proposed detector, as discussed in the paper, besides performing better than the most popular detection solutions over error-free video sequences, also shows remarkable results when dealing with missing information. Given its effectiveness and joining its low-complexity and limited resource requirements, the proposed detector could be effectively included in error concealment strategies applied at the decoder, in order to improve the final video quality delivered to the user and compensate for quality degradation due to error-prone transmissions.

References

[1] A. Dimou, O. Nemethova, and M. Rupp, "Scene change detection for H.264 using dynamic threshold techniques," in *Proceedings of the 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Service*, Smolenice, Slovak Republic, July 2005.

[2] Y.-H. Ai, W. Ye, S.-L. Feng, B. Hu, and M. Xie, "Predictive picture refresh based on scene-context reference picture for video transmission," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '06)*, pp. 1–4, Wuhan, China, September 2007.

[3] Z. Chen, G. Qiu, Y. Lu, et al., "Improving video coding at scene cuts using attention based adaptive bit allocation," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 3634–3638, New Orleans, Calif, USA, May 2007.

[4] W.-Y. Kung, C.-S. Kim, and C.-C. J. Kuo, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 789–802, 2006.

[5] J. Sastre, P. Usach, A. Moya, V. Naranjo, and J. M. Lopez, "Shot detection method for low bit-rate H.264 video coding," in *Proceedings of the 14th European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, September 2006.

[6] X. Yi and N. Ling, "Fast pixel-based video scene change detection," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 4, pp. 3443–3446, Kobe, Japan, May 2005.

[7] D. Vatolin, "The MSU Video Quality Measurement Tool," http://compression.ru/video/quality_measure/video_measurement_tool_en.html, August 2009.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 153160, 16 pages
doi:10.1155/2010/153160

Research Article

Automatic TV Broadcast Structuring

Gaël Manson and Sid-Ahmed Berrani

Orange Labs, France Telecom, Cesson Sévigné 35510, France

Correspondence should be addressed to Gaël Manson, gael.manson@gmail.com

Received 2 October 2009; Accepted 11 January 2010

Academic Editor: Jungong Han

TV broadcast structuring is needed to precisely extract long useful programs. These can be either archived as part of our audio-visual heritage or used to build added-value novel TV services like TVoD or Catch-up-TV. First, the problem of digital TV content structuring is positioned. Related work and existing solutions are deeply and carefully analyzed. This paper presents then DealTV, our fully automatic system. It is based on studying repeated sequences in the TV stream in order to segment it. Segments are then classified using an inductive logic programming-based technique that makes use of the temporal relationships between segments. Metadata are finally used to label and extract programs using simple overlapping-based criteria. Each processing step of DealTV has been separately evaluated in order to carefully analyze its impact on the final results. The system has been proven on a real TV stream to be very effective.

1. Introduction

Broadcasted digital TV contents have incredibly increased over the last few decades. The resulting huge and continuously growing content has given rise to many novel services around TV and video platforms like TV-on-Demand (TVoD), interactive TV, Catch-up-TV, Network Personal Video Records (NPVRs), and so forth. This content is also part of our audio-visual heritage and must be properly archived. Archiving digital TV content is generally achieved by national public institutions like INA in France, Beeld en Geluid in Netherlands, ORF in Austria or BBC archives in UK. Therefore, the digital TV content has to be analyzed and indexed in order to be used within services and to be easily retrieved from archives.

Basically, analyzing and indexing a digital TV content consists in finding key instants in the content. These correspond to events of interest (e.g., goals in soccer footage or key scenes in a movie) users may like to directly find through either search engines (in the case of querying an archive) or services built on top of the content. These key instants could also be features and positions that allow structuring the content. Here, the objective is twofold: (1) to properly prepare the content before archiving it in order to easily answer user queries later on; (2) to repurpose the content in another more convenient format to final users.

In the case of TV structuring, the main key instants are the start and end times of each program in TV broadcasts. These times allow automatically recovering the structure of the TV stream. They are at the root of novel added-value services or any archiving service. They allow extracting programs and making them available through a catalog, without any constraint on time. They can therefore be viewed in a nonlinear manner through the aforementioned services. They also allow identifying, isolating and properly archiving *useful* programs that might interest users later on.

This paper focuses on this latter use-case of analyzing and indexing digital TV content coming from TV broadcasts. This is generally referred to as TV broadcast structuring or TV broadcast macro-segmentation. The main contribution of the paper is a novel and fully automatic system for TV broadcast structuring. This system aims at precisely extracting useful TV programs.

The rest of the paper is organized as follows. Section 2 explains why TV broadcast structuring is required for real-world applications, finely analyzes related works and classifies existing approaches into three categories. Section 3 presents DealTV, our fully automatic system that analyzes

the video signal, segments it, classifies the segments, and extracts and annotates programs. Section 4 describes the application we address: "TV program extraction for TVoD services". This application is evaluated and validated on real TV broadcasts collected from a French channel over two weeks. In these experiments, each processing step of our system is separately evaluated. An evaluation of the full system is also provided and compared to a manually created ground-truth.

2. TV Broadcast Structuring and Related Works

In short, the objective of TV broadcast structuring is to recover the original structure of the TV stream. In TV streams, useful long TV programs (like movies, news, series, etc.), short programs (like weather forecast, very short games, etc.) and interprograms (commercials, trailers, sponsorships, etc.) are concatenated and broadcasted without any precise and reliable flags that identify their boundaries. Hence, TV broadcast structuring consists in automatically and accurately determining the boundaries (i.e., the start and end) of each broadcasted program and inter-program as depicted in Figure 1. In addition to precise and automatic boundary detection, TV broadcast structuring gathers different parts of the same program and labels them when metadata are available.

Structuring a TV broadcast can be seen, by many, as a problem that TV channels could solve. Theoretically, this is true. TV channels aggregate the audio-visual content and broadcast it. Hence, they should be able to provide appropriate and reliable metadata on what they broadcast, namely the title, the type, the start and end times and any additional data on each broadcasted program or inter-program. In practice, most TV channels are technically unable to provide such data. Their broadcasting chains are too complex and do not have the appropriate tools to save and, more importantly, forward these metadata. The few other channels that may provide accurate metadata do not necessarily accept to provide them. On the other hand, archiving and building novel services might be done by third parties without any collaboration with TV channels. This is the case of national public institutions in charge of archiving audio-visual heritage. It is also the case of NPVR services for which provided services may even be considered by channels as competing services.

Existing techniques for structuring a TV broadcast are classified into 3 categories. They are described in the following three sections.

2.1. Manual Approaches. As most video analysis and indexing problems, TV broadcast structuring can be manually performed by skilled-workers. In this case, the TV broadcast can either be structured online or offline. Online, workers have to continuously watch the TV broadcast. Each time an event of interest is encountered, it is tagged. Offline, workers linearly browse the saved TV stream and annotate it. Both cases require adequate software applications that allow workers to efficiently structure the stream.

These approaches are currently the most widely used, in particular for structuring and indexing audio-visual heritage before archiving it. It is however prohibitively expensive and unable to handle the currently huge amount of available content. For instance, manually and offline structuring a TV stream of 28 days has taken more than 30 working days using a very powerful and customized software. We have performed this manual annotation in order to create the ground-truth which is required for the evaluation of automatic TV structuring approaches (cf. Section 4). It also suffers the imprecision and errors that workers can make. Indeed, contrary to what one could assume, manual approaches are not always the most reliable. Structuring a TV broadcast is a laborious repetitive task that requires permanent concentration.

2.2. Metadata-Based Approaches. There are mainly two types of metadata that are provided by TV channels and that are used to describe TV broadcasts: (1) metadata that are associated and broadcasted with the TV stream, and (2) metadata that can be retrieved from specialized websites which gather electronic program guides. Both metadata provide information on programs only. Interprograms are not mentioned.

Metadata associated with the stream depend on standards and broadcasting modes (analog/digital). In the case of analog TV that is in the process of becoming defunct, metadata were available within teletext (or Closed Caption in US). Teletext encloses a large amount of data such as news, weather forecasts, and so forth, and also *static* information on the program schedule. European standard teletext could also include Program Delivery Control (PDC) [1]. PDC is a system that properly controls equipped video recorders by using hidden codes in the teletext service. An equivalent service named VPS (Video Programming System) exists in some EU countries (e.g., Czech Republic). These codes allow the user to precisely control the record start and end times of a specific program.

In digital TV, broadcasted metadata are called Event Information Tables (EIT) and are of two types:

(1) EIT schedule: stores the TV program over a number of days

(2) EIT present and follow: contains the details (start and end times, title, and possibly a summary) of the program currently being broadcasted, as well as the following one.

If EIT "present and follow" is generally available, the EIT "schedule" is rarely provided.

Apart from PDC, both for digital and analog TV, broadcasted metadata are static, that is, they are not updated or modified in order to take into account any delay or change that may occur in the broadcast with respect to the initial program schedule.

Unfortunately, PDC cannot be used for TV structuring for many reasons. The main problem is that PDC is very rarely provided, as it allows users to skip commercials. Currently, these represent the main income for TV channels.

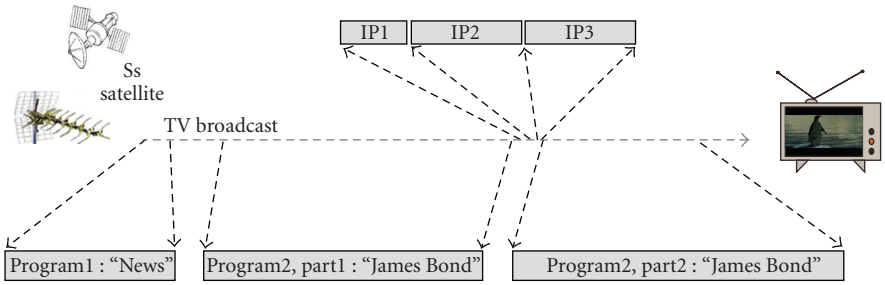


Figure 1: TV broadcast structuring. Detection of program and interprogram boundaries.

Another limitation of PDC is that it is well defined and standardized for analog TV, but in digital TV, standards are not built in and most broadcasters are not transmitting the appropriate data.

Metadata available on the web are typically program schedules. These are called Electronic Program Guides (EPG). Many companies (like *emapmedia*) also provide a service in which they gather EPGs from a large number of channels. These EPGs are then made available on a unique server that can be directly queried through the web service.

In order to assess the reliability and the precision of these metadata, a study has been conducted in which the EIT and the EPG metadata have been compared to a manually created ground-truth. This study, presented in [2], shows that metadata are generally imprecise, do not cover all the broadcasted programs and do not take into account late modifications of the schedule. For instance, it shows that over a 24 hour broadcast, more than 40% of the programs start more than 5 minutes earlier or later than that expected in the metadata. Another study [3] has been performed on program guides from 5 channels over 3 years. It has also shown that more than 75% of broadcasted programs and interprograms are not mentioned in the program guides. Based on these results, metadata cannot be *directly* used to structure TV broadcasts.

On the other hand, apart from traditional techniques for metadata aggregation and fusion [4, 5] which could be used to enrich them and increase their accuracy, only very few studies (among which [6]) have proposed novel ways to make use of these metadata.

In [6], Poli proposes a statistical predictive approach that allows correcting an EPG using a model learned from a ground-truth created on a one year TV broadcast. This approach is based on a simple observation: channels have to follow roughly the same schedule in order to increase their audience. The main drawback to this approach is the required ground-truth data for training. This ground-truth is very difficult and prohibitively expensive to be collected. Moreover, it has to be separately collected for each channel as the program schedule differs from one channel to the other. Poli's study was feasible because it was conducted at INA, the French National Audiovisual Institute in charge of indexing and archiving French channels (http://www.ina.fr). On the other hand, the model does not take into account program schedule of special events that may occur without any regularity from one year to the next (e.g. political events, sports competitions, etc.).

2.3. Content-Based Approaches. Content-based approaches rely on the study of the basic audio and video signals in order to recover the high-level structure of the stream. Basic techniques one can think of are shot and scene (or story) segmentation. Low level features like motion and color are extracted from each frame and are analyzed in order to segment the stream into shots. Shots are then gathered into scenes with respect to their similarities and temporal order [7–9]. Scenes are subjective and not very well defined but, for instance for a movie, they generally try to meet the main chapters of the movie like those prepared for the DVD. In the context of TV structuring, precisely extracting programs would require clustering all the scenes of the same program and separating them from those of the other programs. However, TV programs, and thus their scenes, are very heterogeneous and do not generally share any common features or structure. It is therefore very hard to make use of scene detection and clustering to perform TV stream structuring.

Another content-based approach would focus on detecting boundaries between programs and interprograms in the TV stream. This has been investigated by Wang et al. [10] and a multimodal boundary classification system has been proposed. The system uses visual, audio and textual features within an SVM classifier in order to find transitions between programs among all the possible transitions in the stream. However, this solution cannot be used to structure any TV stream. It heavily relies on some assumptions on the structure of programs (e.g., presence of specific images at the beginning and at the end of each program and interprogram) and on a complex training procedure.

Finally, a novel content-based approach makes use of the fact that some sequences are broadcasted several times in the stream and are then repeated sequences. These are commercials, trailers, credits of TV series, sponsorships... If the occurrences of these repeated sequences can be automatically detected and identified in the stream, then the stream can be segmented. The resulting segments can be then classified and analyzed toward performing the structuring. This content-based approach is the most promising. Techniques following this principle are classified into two categories and are described in the two following subsections.

2.3.1. Reference Database-Based Techniques. The basic idea of reference database-based techniques is to manually label repeated sequences and store them in a reference database. Here, labeled and stored repeated sequences should include most of the interprograms, opening and closing credits of recurring programs (like TV series) and possibly any other program shot. These sequences are identified later on in the TV stream using a content-based matching technique. TV structuring is therefore reduced to content-based real-time sequence identification in an audio-visual stream. The start (resp., end) time of recurring programs that have a stable opening (resp., closing) credit is detected when the credit is identified. Other parts of the stream are segmented by detecting interprograms. Each gap of a significant duration between two consecutive segments of interprograms is considered as a program segment and is labeled using metadata (like EIT) when these are available. A program segment can also be labeled, if one of its shots has been matched with a shot that has previously labeled and stored in the reference database.

Methods following this principle can be built on top of audio or video fingerprinting [11, 12] techniques. These can be used to detect in the TV stream, referenced sequences stored in the database. Perceptual hashing can also be used[13–15].

Naturel et al. [16] propose a complete system for TV structuring based on this principle. In addition to a hashing-based identification of stored and labeled shots, a dynamic time warping procedure has been proposed in order to *match* extracted program segments with metadata provided in the EPG. The set of interprograms is also updated using a commercial detection method based on the same features as in [17] (i.e., monochrome frames, silence, etc.).

These approaches mainly have two drawbacks, both related to the reference database. First, this database has to be created manually for each TV channel. It must also contain a sufficient amount of interprograms, credits and labeled shots in order to achieve a good and precise TV structuring. Second, the database has to be periodically updated as new interprograms, new series (and hence new credits) are continuously introduced.

2.3.2. Techniques Based on Automatic Detection of Repeated Sequences. Following the same principle as reference database-based approaches, other techniques rely on segmenting the stream by automatically detecting inter-program segments, deducing program segments and then labeling them using metadata. Unlike reference database-based approaches, these techniques make use of the repetition property of interprograms in order to directly and automatically detect them using a non-supervised solution.

Inspired by video retrieval techniques, Gauch and Shivadas [18, 19] propose a video shot-based solution. Shots are described and indexed using perceptual hashing. Repeated shots are then detected using a two step procedure. The first step is based on collisions in the hash table. The second one is based on the visual similarity between shots. Adjacent repeated shots are merged and classified (commercials or not). Covell et al. [20] propose an approach following the same principle as Gauch et al., but technically different. Repeated objects are detected using audio features and a hashing-based method. Detections are then checked using visual features. As for Herley [21], interprograms are detected as *repeating objects* using a correlation study of audio features. At time t, the current *object* (an audio segment of predefined length) is compared to a past stored buffer of fixed size in order to detect any possible correlation.

Even if fully automatic, these techniques are not sufficient to perform TV structuring. They all require a post-processing step in which automatically detected repeated sequences need to be mined before being used in the structuring process. Indeed, repeated sequences also include sequences that are broadcasted several times but that are not interprograms. Examples of such sequences are news reports, flashback sequences in movies and series, and so forth.

On the other hand, these solutions are technically limited. Solutions proposed in [18–20] focus on the detection of commercials. They suffer the drawbacks of content-based matching techniques using hash-tables, which are mainly related to the difficulty of choosing a suitable hash function with respect to the target similarity. They are also *brute force* in the sense that all descriptors of the whole audio-visual stream have to be inserted in the hash table and also saved, which could raise efficiency problems when dealing with a large amount of audio-visual data or with a continuous TV broadcast in a real-world system. The method by Herley [21] requires some crucial parameters related to the size of descriptors, the search window and the length of the search buffer. These restrict the depth of the search and limit the detection to a *pre-defined fixed* size range of repeating objects.

Additionally to all these approaches, it is worth pointing out in this section that the TV structuring research field also includes works on commercial detection and program genre classification. Commercials have attracted a lot of attention because of their importance in the business model of TV broadcasting. They are still the main income of TV channels. Existing works on commercial detection rely generally on intrinsic features of commercials (e.g., motion, audio, action) and on detecting separations between commercials (monochrome frames and audio cuts) [17]. They also rely on detecting the logo channel and on studying the shot duration [22]. Many program genre detection techniques have also been proposed [23–25]. They generally classify programs into categories like news, commercials, cartoon, sport, TV series, weather, and so forth. They assume that programs have already been properly segmented and extracted.

3. DealTV: A fully Automatic System

In this section, we describe our novel fully automatic system for TV Broadcast structuring. It is based on the same principle as techniques based on the detection of repeated sequences. It addresses, however, the limitations of existing approaches and focuses on extracting long useful programs. We recall that useful programs are long TV programs, International Journal of Digital Multimedia Broadcasting 5 namely movies, news, TV series, TV shows, etc. These are the most important content of TV streams.

Our system uses two methods from our previous works for repeated sequence detection [26] and for program segment classification [27]. These methods are improved, adapted and put together with other techniques in order to efficiently and effectively structure continuous TV broadcasts.

When launching the system for the first time, it needs to accumulate a sufficient amount of stream. The analysis of the stream starts when there are enough repeated sequences in the accumulated stream. As will be shown in the experiments (Section 4.1), 96 hours is sufficient. When the analysis is started, it structures the previously accumulated stream and delivers results. The system then starts accumulating the stream gain. However, this time, the structuring process can be launched anytime on demand or periodically. In this case, if the duration of the newly accumulated stream is not sufficient, it is merged with the previous period of the stream. This processing scheme is depicted in Figure 2.

Each time the stream structuring is launched on a portion of the stream (periodically or on demand), the following processing steps are performed (as depicted in Figure 3):

(1) stream segmentation using detected repeated sequences,

(2) resulting segments classification in order to detect useful program segments,

(3) useful programs extraction and labeling.

First, repeated sequences detection uses a micro-clustering technique that does not make any assumptions on the length or frequency of the repeated sequences. Moreover, this detection can be performed whatever the length of the TV stream is. Our system also covers all of the steps from the first step of repeated sequence detection to the final step of program extraction. Detected repeated sequences are used to segment the stream and the resulting segments are then classified in order to isolate segments that belong to useful programs from the rest of the segments (inter-program and short program segments). This classification step is based on inductive logical programming. Program segments are finally labeled using a matching procedure with respect to metadata.

For the sake of simplicity, in the following we consider that we have accumulated a portion of the stream and we describe each step of the structuring process given that portion.

3.1. Stream Segmentation Using Detected Repeated Sequences. Stream segmentation is based on detecting repeated sequences in the stream, that is, sequences that are broadcasted several times. These include (but are not limited to) interprograms, whole programs and parts of programs that are broadcasted several times. Our repeated sequence detection relies on extracting and clustering visual features.

3.1.1. Stream Description. Our repeated sequence detection technique uses a two level visual description scheme. A first level at which an exhaustive description is performed. A basic visual descriptor (BVD) is extracted from each frame of the video stream. It is used to match almost identical frames and only needs to be invariant to small variations due to compression, for instance. The second level focuses on carefully chosen keyframes of the video stream. The descriptor associated with these keyframes is called key visual descriptor (KVD). It is more sophisticated and has to be more robust. KVDs are used during the clustering step to cluster similar shots and to create the set of repeated sequences. However, at this stage, the boundaries of detected repeated sequences cannot be determined. A KVD is associated with a frame of the repeated sequence but does not provide any information on the sequence boundary. The BVD is thus used to precisely determine these boundaries by matching corresponding frames in all occurrences of the repeated sequence.

Both BVDs and KVDs are DCT-based descriptors. To compute a BVD, the frame is divided into 4 blocks and each block is sub-sampled to a matrix of 8×8. A DCT is then applied on each block and one DC coefficient and the 15 first AC coefficients (according to the zig-zag order) are computed. Each coefficient is then binarized and a 64-bit descriptor is created. BVDs are compared using the Hamming distance.

As for KVDs, they are computed from keyframes. These are chosen after a shot segmentation of the video stream following the method described in [26]. To make the KVD robust to spatial variations, like subtitles incursion or logo insertion/removal, the keyframe is divided into 3×2 blocks. Six independent descriptors are computed on the six blocks and then concatenated into a single descriptor. To compute block descriptors, each block is first subsampled to a 8×8 matrix. A DCT is computed on this matrix and the first five coefficients (according to the zig-zag order) are selected to build the descriptor. The KVD is hence a 30-dimensional vector. The similarity between KVDs is measured using the L_2 distance.

3.1.2. Clustering Step. To gather similar keyframes that will be used to detect repeated sequences, a clustering technique is used. However, unlike most applications using clustering, we are interested in finding a large number of very small clusters within a huge amount of uniformly distributed and isolated vectors. The number of KVDs per cluster is determined by the number of times a sequence is repeated. If a sequence is repeated three times, and it is described by five KVDs (i.e., five keyframes have been selected from the sequence), then we should ideally discover five clusters with three KVDs inside each one. The number of KVDs per cluster ranges thus from two to few hundreds. The number of clusters corresponds to the number of KVDs in the repeated sequences. As for the rate of outliers, it corresponds to the rate of KVDs that do not belong to any repeated sequence, which could be very high. On the other hand, the clustering algorithm should also be able to process KVDs on the fly as they are computed from the video stream. Based on these criteria, we propose to use a micro-clustering technique similar to BIRCH [28].

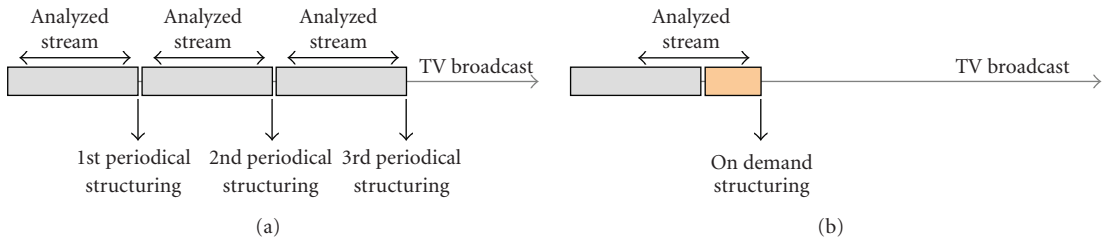


Figure 2: Processing scheme of the TV broadcast by DealTV. (a) Periodical structuring. (b) On demand structuring.

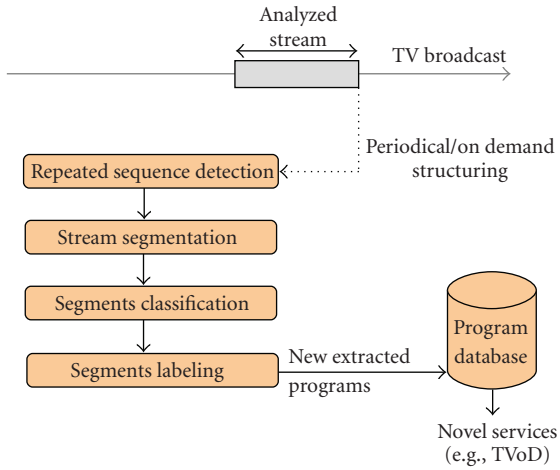


Figure 3: Structuring steps of DealTV.

It is an iterative procedure that builds spherical clusters whose radii are controlled and must be below a threshold Ft. At the beginning, Ft is chosen to a very low value. During the first iteration, KVDs are inserted: a KVD is associated with a cluster of previously inserted KVDs only if the radius of the resulting bounding hyper-sphere of the cluster is less than Ft_1 (the value of Ft in the first iteration). If no existing cluster can *absorb* the KVD, it is put in a new cluster as a singleton (a cluster that contains only one KVD and of which the radius is equal to 0). To characterize clusters and facilitate their use, a clustering feature (CF) vector is associated to each cluster. A CF vector is a triple composed of the number and the sum of the KVDs belonging to the cluster, and the radius of its bounding hyper-sphere. When merging two clusters, the CF vector of the resulting cluster can be easily computed using only the CF vectors of the two clusters. That way, KVDs are read only once. In the following iterations, only CF vectors are manipulated.

After each iteration, the singletons are isolated as outliers. If the number of the remaining clusters falls below a fixed number then the clustering process is finished. Otherwise, the Ft is increased and a new iteration is performed.

As explained previously, the maximum number of clusters is related to the number of repeated sequences and the number of associated keyframes. It can, therefore, be experimentally determined by a study over a sample of TV broadcast.

On the other hand, in order to efficiently cope with the general periodical working scheme of DealTV, the clustering procedure should be incremental. This is important to reduce the cost of this crucial and costly processing step. In our system, this is taken into account at 2 levels. First, the CF vectors and the way the clustering technique works allows processing KVDs as they are computed. Second, when a new portion of the stream is accumulated and processed (P9 in Figure 4), the equivalent oldest portion in the clustering buffer is removed (P2 in Figure 4). The clustering process works hence as a continuously updated sliding window. This is allowed by the powerful way CF vectors are computed and manipulated. This way, when the structuring analysis is launched, the first (and the most costly) iteration of the clustering is already performed. Figure 4 depicts how the incremental clustering works.

3.1.3. Repeated Sequences Detection. Once the clusters are generated, they are analyzed in order to create repeated sequences. First, clusters that contain KVDs extracted from the same shot or from neighboring shots are removed. For example, this happens during a debate when shots are alternatively centered around the antagonists.

To make use of the rest of clusters, an inter-cluster similarity is defined. This similarity measures the chance for two clusters to generate a repeated sequence. To explain how it is computed, lets consider two clusters C_i and

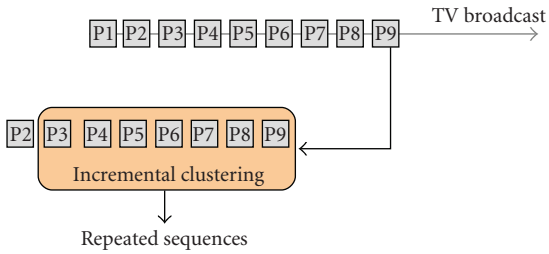


Figure 4: Working scheme of incremental clustering.

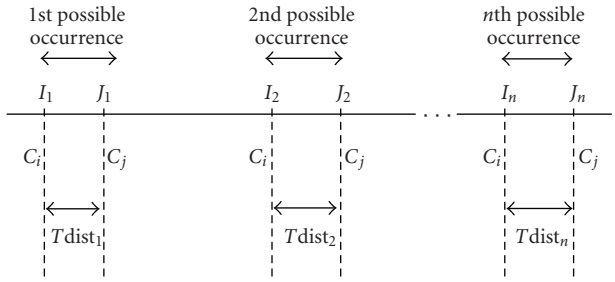


Figure 5: Two interlaced clusters with n KVDs each.

C_j, containing n KVDs each. The set $\{I_1, \ldots, I_n\}$ (resp., $\{J_1, \ldots, J_n\}$) is the set of keyframes whose KVDs belong to C_i (resp. C_j). The first condition for two clusters to generate a repeated sequence concerns the temporal order of keyframes $\{I_k\}_k$ and $\{J_k\}_k$. They have to be alternating as depicted in Figure 5. In this case, we say that C_i and C_j are interlaced.

The temporal distance between each couple of keyframes from the two clusters is the second condition. Temporal distances between each couple (I_k, J_k) must be nearly the same if the $\{I_k\}_k$ and the $\{J_k\}_k$ are keyframes of a repeated sequence. The chance for C_i and C_j to generate a repeated sequence is thus related to the constancy of the temporal distance (Tdist) between their keyframes. The inter-cluster similarity must be one if all the temporal distances are identical and decreases when they differ. The following formula is used to compute the inter-cluster similarity between clusters C_i and C_j:

$$\mathrm{Sim}_{ij} = \begin{cases} 0, & \text{if } C_i \text{ and } C_j \text{ are not interlaced,} \\ e^{-\sigma}, & \text{otherwise,} \end{cases} \quad (1)$$

where σ is the standard deviation of $T\mathrm{dist}_1, \ldots, T\mathrm{dist}_n$ (cf. Figure 5).

The inter-cluster similarity is computed between each couple of clusters. The results are stored in a matrix B, where element b_{ij} is the similarity between clusters C_i and C_j (i.e., $b_{ij} = \mathrm{Sim}_{ij}$). To process clusters, we define a basic relation between clusters as follows: if $b_{ij} > \delta_{\mathrm{sim}}$ then clusters C_i and C_j are related. This relation is aimed at gathering clusters that have a high inter-cluster similarity and, hence, that are likely to generate a repeated sequence.

The rest of the process is summarized in the following steps.

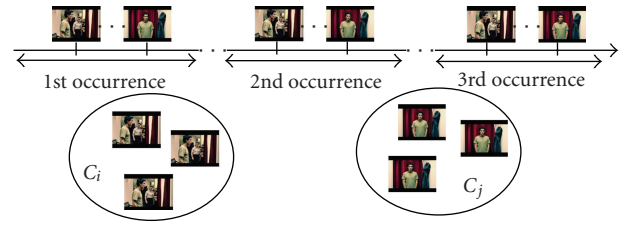


Figure 6: Generation of a three times repeated sequence from two clusters.

(1) Select clusters that have at least one relation with another cluster.

(2) Select the set of most populated clusters, that is, clusters having the highest number of KVDs.

(3) Perform transitive closure within the selected set. Again, the objective is to partition the set into subsets into which each cluster is related to one or more other clusters from the same subset.

(4) Generate a repeated sequence from each subset as depicted in Figure 6 and remove the used clusters from B.

(5) Continue with the process from step (1) until no cluster is selected.

In step (2), the process starts with the most populated clusters in order to first retrieve the most frequent repeated sequences. Indeed, as depicted in Figure 5, for a chosen subset of clusters, the number of occurrences of a repeated sequence is equal to the number of KVDs per cluster (we recall that all the clusters within the subset are interlaced and have the same number of KVDs each).

In step (4), the generation of repeated sequences consists in defining the boundaries of each occurrence of the repeated sequence. First, the boundaries are defined by the most left and the most right keyframes. These boundaries are then extended using BVDs computed for all the frames of the stream. The occurrences are extended to the left (resp., the right) if BVDs of all the left (resp., right) neighboring frames for all the occurrences are similar. To make the extension procedure more robust, we propose to simultaneously compare a set of neighboring frames, that is, the extension procedure compares the m left (resp., right) frames of all the occurrences every time. If the average dissimilarity is less than a threshold, then the occurrences are extended to the left (resp. right) by m frames.

The repeated sequence detection procedure is also able to retrieve trailers as repeated sequences and can sometimes match them with the corresponding program. This is not described here. The interested reader can refer to [26] for a complete and detailed description of the procedure.

3.1.4. Stream Segmentation. Finally, the stream is segmented as depicted in Figure 7. First, each occurrence of a repeated sequence is considered as a segment. Then, each *gap* between two consecutive segments is also considered as a segment.

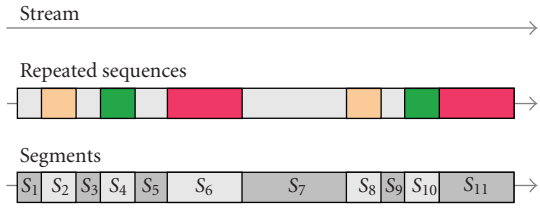


Figure 7: Stream segmentation using detected repeated sequences. Eleven segments generated.

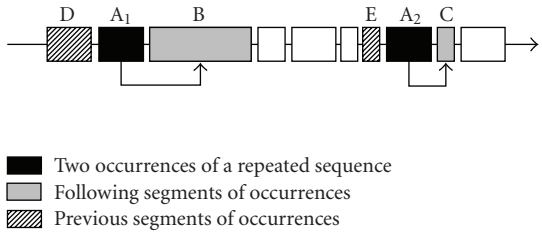


Figure 8: Illustration of adjacent segments to the occurrences of a repeated sequence.

3.2. Segment Classification Using ILP. Once the stream is segmented, the problem is to automatically detect segments that are part of long useful programs. It is a classification task with two classes: (1) the class of long program segments and (2) the class of the other segments that include segments of interprograms and segments of short programs.

To perform this task, local features of each segment (like the duration) can be used. However, the linear nature of the stream and the way long programs, short programs and interprograms are sequenced within the stream provide powerful features that greatly help distinguishing these segments. In the following, these features are referred to as neighborhood and relational features.

In order to take into account both kind of features (local and relational), our classification module uses Inductive Logical Programming (ILP) following the method described in [27]. ILP allows us to train offline a classifier that implicitly models the relational features and to easily take into account prior knowledge. Moreover, the resulting classification rules are easily understandable.

The features and the ILP classification module are described in the following two subsections.

3.2.1. Segment Features. Three kinds of features are used to characterize a segment: local, contextual and relational features.

(1) *Local Features.* These features describe a segment with features that do not depend on others segments. We use only the duration of the segment and the number of times it repeats in the stream (0 if the segment does not repeat).

(2) *Contextual Features.* These features take into account the context of the segment. We define the following features.

(i) If the segment is an occurrence of a repeated sequence, we compute the mean of the number of repetitions of the following segments adjacent to each occurrence. This is illustrated in Figure 8. This feature applies to occurrences $\{A_1, A_2\}$ and is equal to $(|B|+|C|)/2$, where $|X|$ denotes the number of occurrences of the segment X. In the same manner, we compute the mean of the number of repetitions of the *previous* segments that are adjacent to each occurrence (segments D and E in Figure 8). We propose this feature in order to help discriminating opening/closing credits from the set of repeated sequences. Indeed, when this feature is null, that means that the segment always lies before/after a long segment that does not repeat and that is most likely a long program. This feature may also help to classify segments that always lie between other very frequent repeated segments.

(ii) As contextual features, we also consider the local and contextual features of an adjacent segment: segments in the neighborhood of the considered segment.

(3) *Relational Features.* These features are the class of the neighboring segments. Therefore, they apply only when at least one of the neighboring segments has already been classified. They allow to take into account the class of neighboring segments.

3.2.2. ILP Classification. ILP can directly manage complex logical relationships between segments and returns explicit rules in the form of first order logic. Prior knowledge can also be easily taken into account. They just have to be encoded as first order logical rules and added to the background knowledge. However, as ILP does not handle numerical data, all the local and relational features have to be transformed to symbolic attributes. Categories for symbolic attributes are defined using numerical intervals based on prior knowledge.

An ILP system builds a *logical program* from the background knowledge and a set of training examples represented as a set of logical facts. This logical program is composed of a set of first order logical rules that cover all the positive and none of the negative training examples. ILP infers rules from examples by using computational logic as the representation mechanism for hypotheses and examples. An example of logical rule that can be learned is "*If a segment A does not repeat and A is long then A is a long program segment*" or "*If a segment A repeats often and A is followed by a segment B that is not a long program segment then A is not a long program segment*".

The neighboring relationships are hence represented in the training set by a set of facts that gives the following segment for each segment of the stream. They are also represented by a recursive rule that transitively defines this relation, which allows to define a "distance" between segments.

In our implementation, we have used *Aleph (ex-P-Progol)*, a descendant ILP system which performs training from general to specific hypotheses [29, 30].

The logical rules computed by ILP define requirements for a segment A to belongs to the class of "long programs" or the class of "others" (IP or short programs). They can be sorted into four categories following how they model segment features.

(1) *Simple not recursive rules (SNR-rules)* rely only on the local and contextual features of A.

(2) *Simple recursive rules (SR-rules)* rely in addition to (1) on the fact that some neighboring segments belong to the class of A.

(3) *Relational not recursive rules (RNR-rules)* rely in addition to (1) on the fact that some neighboring segments belong to a class distinct of the class of A.

(4) *Relational recursive rules (RR-rules)* rely in addition to (3) on the fact that some neighboring segments belong to the class of A.

In order to compute logical rules that define "long programs" or "others", we first encode a part of the TV stream as a database of logical facts. This is the training set. The ILP system infers then a set of logical rules. Some of these rules are generic and very relevant. However, some other inferred rules are very specific to special cases of the training set and may confuse the classifier. Thus, in order to select the relevant rules, we propose to use an additional validation phase. The learned rules are applied on the validation set and depending on their precision, a confidence level is associated with each of them. The higher the precision, the higher the confidence level. Details on the number of considered confidence levels are given in Section 4.4.

The training phase provides hence a set of rules (SNR, SR, RNR, RR) ordered by their levels of confidence (the highest level of confidence = 0). The classification step takes into account the confidence levels and the types of rules. Prior knowledge rules are considered as the most reliable. They are applied first and at the beginning of each iteration. The classification phase consists in the following procedure:

(1) apply prior knowledge rules,

(2) select a subset E_i (initially $i = 0$) of the rules with the level of confidence i,

(3) select and apply SNR-rules for the class "long programs" from E_i,

(4) select and apply recursively SR-rules for "long programs" from E_i,

(5) do (3) and (4) for the class "others".

(6) select and apply RNR-rules for "long programs" from E_i,

(7) select and apply recursively RR-rules for "long programs" from E_i,

(8) do (6) and (7) for "others".

(9) select the next level of confidence: $i = i + 1$ and continue with step (1).

3.3. Program Extraction and Labeling. The ILP classification step detects and isolates segments that are parts of long programs from the whole step of segments. When no metadata is available, no further processing steps can be achieved. Segments can only be presented to a user to be manually annotated. However, at least the EPG is generally available for most channels. In this case, despite their imprecision, metadata are very helpful to fuse, extract and label long programs. Metadata only provide approximate start and end times of the broadcasted programs. These starts and ends are used to build the metadata segments that are analyzed and compared to the extracted segments in order to perform program extraction and labeling.

An interesting approach for TV segment labeling is presented in [16]. The authors have considered labeling as a problem of sequence alignment between the detected program segments and the metadata segments. A Dynamic Time Warping (DTW) algorithm has been used to fuse the segments. It uses the edit distance between the set of program segments and the set of metadata segments. The edit distance is a well-known method for aligning two sequences X and Y. It evaluates minimum weight for transforming X into Y by a set of weighted edit operations. The used operations are here defined as substitution, insertion and deletion. In order to drastically improve the results, the authors use a landmarked DTW to force local alignment. This forced alignment is based on previously manually labeled segments that are recognized.

In general, the labeling step is not a straightforward alignment issue. Indeed, TV programs may be cut into several parts that are separated by interprograms (commercials in particular) and this is not mentioned in the metadata. The basic substitution, insertion and deletion operations are not sufficient to deal with this. They are not able to fuse different parts of the same program.

In our system, the labeling procedure is depicted in Figure 9.

(1) Segments classified as long programs (P) are selected.

(2) Consecutive long program segments are gathered into a single long program segment.

(3) Resulting long program segments are labeled using metadata segments following a temporal overlapping criteria.

(4) Consecutive long program segments labeled with the same label are fused into a single TV program.

The labeling procedure is based on studying the temporal overlapping between the detected program segments and the metadata segments. For each detected long program segment, metadata segments that have a non empty temporal intersection with the long program segment are selected and for each one, the temporal overlapping rate is computed. The metadata segment with the highest overlap is selected if its overlapping rate is significantly greater than the one of the second most overlapping segment. If metadata segments having the highest overlaps present overlapping rates that are very close, the segment whose duration is the closest to the segment to label is selected.

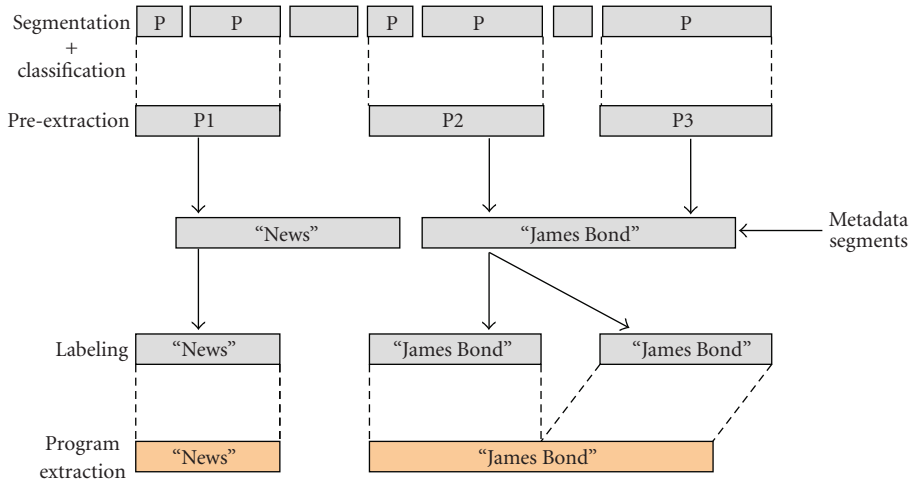


Figure 9: Program extraction and labeling.

This labeling procedure is a local approach that does not heavily rely on the metadata. It is also able extract the start and the end of each program while keeping information on the location of interprograms separating the different parts of the programs. This is very important to remove/replace commercials in services like TVoD.

4. Experiments

In this section, we evaluate and validate the DealTV system for automatic TV structuring. For this purpose, we have performed a set of experiments using real TV broadcasts. First, we present the real TV broadcast-based dataset that we used. Next, we evaluate each step that makes up DealTV namely, the repeated sequence detection, the TV stream segmentation, the ILP-based classification for program segment detection and finally, the TV program extraction.

The main and more important experiment is the last one. It allows us to validate DealTV for TV program extraction, our ultimate goal. The other experiments are presented in order to evaluate each processing step of DealTV and to understand its impact on the overall system performance.

All the algorithms have been developed in C++. Experiments have been performed on a PC under Windows XP. Its CPU is a 2 GHz Intel Xeon, with 3 GB of main memory.

4.1. Dataset and Ground-Truth. The dataset we have used is a real TV broadcast collected from a French channel over two weeks. It is called *TVData* in the following sections. We have selected one week for training. It is called *TVTrain*. The other week has been used for testing. It is called *TVTest*.

The dataset being collected from a French TV channel that is regulated by the European Union legislation, the duration and the frequency of commercial breaks are limited. We have measured about 15 hours and 22 minutes of interprograms over the 7 days of *TVTrain*. Table 1 presents the different categories of segment that compose *TVTrain*. The last category "other interprograms" gathers all the other small categories of interprograms and includes short channel games, single music clips, and so forth.

Table 1: Composition of *TVTrain*.

Long programs	87.63%
Commercials	5.02%
Trailers	3.16%
Short programs	2.99%
Sponsorships	0.54%
Jingle	0.33%
Logos	0.12%
Other interprograms	0.21%

Table 2: Day intervals.

Night	from 1 am to 8 am
Morning	from 8 am to 12 am
Midday	from 12 am to 2 pm
Afternoon	from 2 pm to 6 pm
Early evening	from 6 pm to 8 pm
Evening	from 8 pm to 1 am

In order to present detailed results, we have chosen to partition the day into six intervals. These are defined in Table 2. They follow the structure of a TV guide.

In our evaluation, we were unfortunately unable to compare our results to related works. To the best of our knowledge, there is no evaluation campaign for TV structuring and there is no available international corpus that can be used for this purpose. The TREC Video Retrieval Evaluation (TRECVid) only provides a corpus of already segmented TV programs. It does not contain any continuously recorded TV broadcast over several days.

In order to evaluate our solution, we have manually segmented and labeled TV Data. This has provided a ground-truth that has been used to compute evaluation metrics such as precision and recall. This ground-truth has been also used

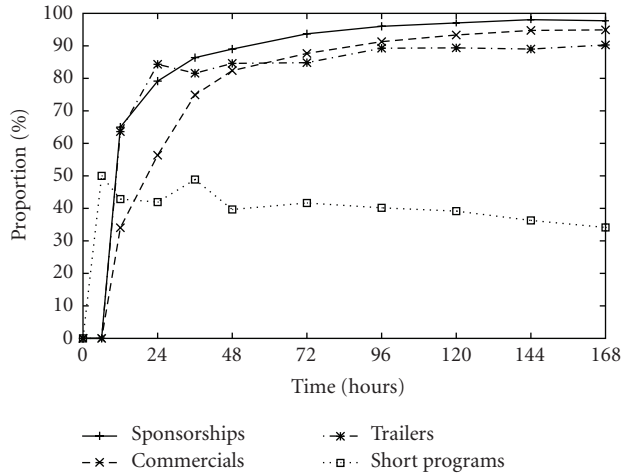


Figure 10: Proportion of repeated segments by categories in *TVTrain* with respect to the accumulated hours of TV stream.

to study the repetition rate of interprograms in *TVTrain*. It has also provided useful information on the structure of the stream. In particular, a set of 374 groups of repeated segments has been discovered with a total number of 2782 occurrences (repeated segments) in *TVTrain*. The most frequent segment is a sponsorship that has been broadcasted 34 times.

Within *TVTrain*, we have also focused on the three most important inter-program categories (i.e., commercials, sponsorships and trailers) and on short programs. We have then computed the proportion of commercials, sponsorships, trailers and short programs that repeat with respect to the accumulated TV stream. This is shown in Figure 10.

This figure shows two main points. First, more than 90% of commercials, sponsorships and trailers are broadcasted at least twice within 4 days. Moreover, this rate does not increase anymore. The remaining 10% of these interprograms do not repeat. They can thus only be detected using the neighborhood and the ILP classification. Nevertheless, this result validates our main idea behind our solution regarding the repetition property of interprograms.

The second point is that only about 30% of short programs are repeated. This implies that detecting short program segments heavily relies on the ILP-classification step. This also lets us suppose that detecting short programs is the key for an accurate TV program extraction.

As explained previously, our system performs a periodical analysis of the TV stream. For each period, the system computes a clustering of the accumulated TV stream in order to detect repeated sequences and to perform segmentation. The results of this first analysis on the content of *TVTrain*, allow us to choose the size of the required accumulated TV stream. With a background of 7 days, we ensure that most of interprograms are within the set of repeated segments. The period is therefore fixed to 7 days.

It is important to note that *TVTest* has been recorded one week after the end of *TVTrain* as depicted in Figure 11. This is required because the clustering step uses an accumulated

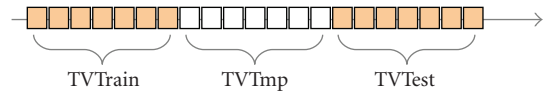


Figure 11: Positions of *TVTest* and *TVTrain* within the TV stream.

stream of 7 days for each processed day of *TVTest*. The week *TVTmp* that is between *TVTrain* and *TVTest* has thus also been recorded. This way, the stream used for training is completely separated from the stream used to testing.

4.2. Repeated Sequence Detection. The analysis of the *TVTrain* dataset shows that most of interprograms are broacasted several times. In this section, we present experiments that evaluate the ability of our system to automatically detect repeated sequences in the stream.

This evaluation has been conducted separately on *TVTrain* and on *TVTest* in order to show also the stability of the repeated sequence detection. Precision and recall have been used as evaluation metrics.

4.2.1. Repeated Sequence Detection on TVTrain. Repeated sequence detection has been applied on *TVTrain*. A set of 775 repeated sequences has been discovered with a total number of 3718 occurrences. This is higher than the number of manually gathered repeated segments (i.e., 2782). The most frequent detected repeated sequence is also a sponsorship that is repeated 30 times. It is not the same sponsorship that has been manually annotated and that repeats 34 times.

This does not mean that our repeated sequence detection technique does not perform well. Due to the position of the keyframes, many repeated sequences have been divided into several repeated subsequences. Therefore, in order to evaluate the automatically detected repeated sequences, we have computed the recall R_{seqret} on a per shot basis. The number of shots that belong both to detected repeated sequences and to manually gathered repeated segments have been calculated. The recall R_{seqret} is then the proportion of this number with respect to the total number of shots of the manually gathered repeated segments. Table 3 shows the obtained results. We have also focused on the three main inter-program categories and on short programs. The results show that repeated segments are very well detected. We can notice the sponsorships are less detected than commercials. This can be explained as follows. In order to be detected, repeated segments must contain two keyframes that have to be gathered into similar clusters. However, sponsorships are often made up of only one shot with only one keyframe.

As for short programs, missed repeated sequences are due to only one short TV game. In this TV game, only a few portions of the segments are changing. There are too many versions of this short TV game that share too many features, which confuses our system.

We have also measured the precision on a per shot basis of the detected repeated sequences with respect to all the manually segmented interprograms and short programs. We call this precision P_{seqret}. The computed precision is

Table 3: Recall R_{seqret} of repeated segments per category in *TVTrain* on a per shot basis.

Commercials	98.55%
Trailers	97.85%
Sponsorships	92.54%
Short programs	92.38%

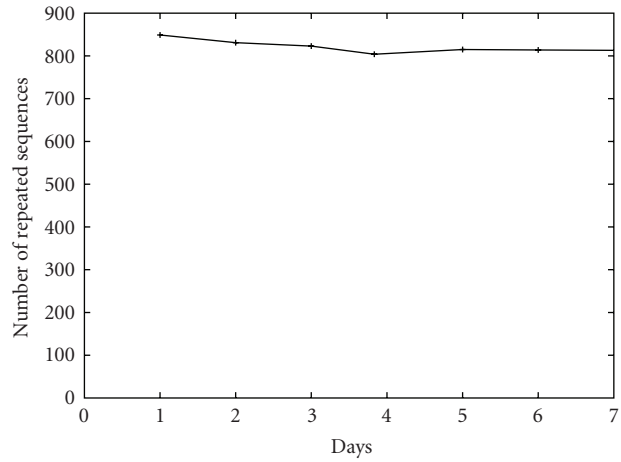


Figure 12: Variation of the number of detected repeated sequences.

64.54%. In other words, about 35% of the detected repeated sequences are repeated segments that are parts of long programs.

To evaluate the precision with respect to the repeated sequences, we have selected the 50 most repeated sequences and we have randomly chosen 50 other sequences. Then, we manually evaluate the results. The observed precision is equal to one. This means that all detected repeated sequences have been effectively repeated sequences.

These experiments allow us to conclude that our repeated sequence detection system is very reliable. However, an efficient classification step is required (as expected) in order to filter out the 35% of repeated sequences that are parts of long programs.

4.2.2. Repeated Sequence Retrieval over TVTest. Repeated sequence detection has been applied on each day of *TVTest*. We have assumed that at the end of each day, an on-demand analysis is launched. Each day is then processed with a sliding accumulated TV stream of 7 days (the six previous days + the analyzed day). We recall that for this reason, the TV stream of the week before *TVTest* has also been recorded. Table 4 summarizes the set of repeated sequences that has been detected with their total number of occurrences. Figure 12 shows how the number of detected repeated sequences varies on the days of *TVTest*. Both the figure and the table show that the system is stable. We can then suppose that we have a recall and a precision similar to those computed on *TVTrain*.

We have also studied the processing time of the repeated sequence detection step. Indeed, our system has to accumulate the TV stream and to periodically or on-demand

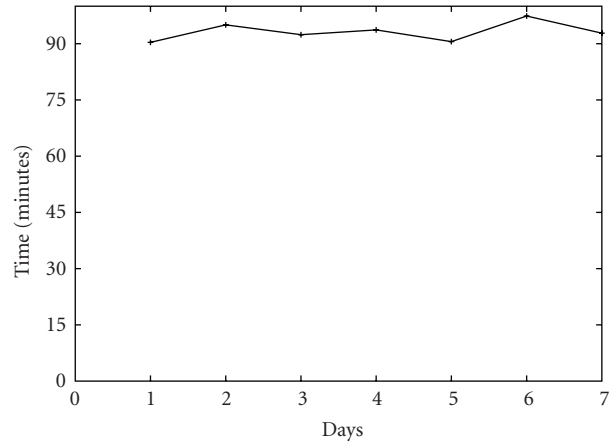


Figure 13: Processing time of repeated sequence detection on each day of *TVTest*. For each day, a sliding accumulated TV stream of 7 days has been taken into account.

analyze the accumulated TV stream. The analysis have thus to be processed before the next analysis is launched. Figure 13 shows the variation of the processing time for each day of *TVTest*. We recall that each day is processed within a sliding accumulated TV stream of 7 days. However, the clustering is not performed from scratch each time. It is updated as explained in Section 3.1. Figure 13 shows that the processing time is quite the same and is always less than 1 hour and 40 minutes. This suggests that the analysis can be launched every two hours which is sufficient in a real world service.

4.3. TV Stream Segmentation. In order to evaluate the segmentation that is performed based on the detected repeated sequences, we have focused on TV programs' boundaries. Indeed, this segmentation provides potential TV program boundaries. Its *quality* depends on the ability to provide boundaries that match those of TV programs given in the ground-truth.

To evaluate the alignment between the computed segmentation and the ground-truth program boundaries, we define specific precision and recall metrics that are tolerant to small imprecisions. The boundaries do not requires an accuracy at the frame level. In practice, a program that is extracted with the sponsorship sequence or without its opening/closing credit is still considered as correctly extracted. Hence, the tolerated imprecision corresponds to the average duration of a sponsorship or a credit; it is equal to 30 seconds.

For each ground-truth boundary, its nearest detected boundary is then found and the temporal distance is computed. The evaluation is achieved using the 3 following metrics.

(i) *The precision* P_{bound} is the number of detected boundaries whose distances to their nearest ground-truth boundaries are less than 30 seconds divided by the total number of detected boundaries.

Table 4: Detected repeated sequences on each day of *TVTest*. For each day, a sliding accumulated TV stream of 7 days has been taken into account.

Days	Repeated Sequences	Number of occurrences	Most frequent sequence
1	849	4026	Commercial (34 times)
2	831	3887	Commercial (32 times)
3	823	3847	Sponsorship (35 times)
4	804	3851	Sponsorship(33 times)
5	815	3936	Sponsorship (35 times)
6	814	3940	Trailer (38 times)
7	813	3894	Trailer (41 times)

Table 5: Evaluation of the TV stream segmentation on *TVTest*.

Day interval	Precision P_{bound}	Recall R_{bound}	Imprecision I_{bound}
Night	0.37	0.94	1.6 s
Morning	0.35	0.95	2.9 s
Midday	0.47	1	0.8 s
Afternoon	0.46	1	1.5 s
Early evening	0.37	0.98	2.3 s
Evening	0.29	0.99	2.3 s

(ii) *The recall* R_{bound} is the number of ground-truth program boundaries whose distances to their nearest detected boundaries are less than 30 seconds divided by the total number of ground-truth program boundaries.

(iii) *The imprecision* I_{bound} is the mean of the absolute temporal distance between the ground truth program boundaries and their nearest detected boundaries.

These metrics have been evaluated on each day of *TVTest*. The obtained results are presented in Table 5. They are averaged and separately presented per day interval.

The obtained results show that the system has performed a good TV stream segmentation. In particular, for the midday and the afternoon intervals, all the ground-truth boundaries are correctly retrieved. Moreover, the average distance between a ground-truth boundary and its detected boundary match is always less than 3 seconds. This is very important for the TV program extraction. However, our system has failed to detect on average about 5% of boundaries from the night and the morning intervals. This is due to the fact that less interprograms and more specifically commercials are broadcasted during these intervals. In particular during the night, many long programs are sequenced without any separating inter-program. These results suggest also that TV program extraction is likely to be more accurate from midday to the evening.

4.4. ILP-Based Classification. In this experiment, the performance of segment classification using the ILP-based technique is studied. We recall that the problem here is to automatically detect segments that are part of long useful programs. It is a classification task with two classes: the class of long program segments and the class of the other segments that include segments of interprograms and short programs. As we are interested at the end of the process on extracting long useful programs, in this study we focus on detecting long program segments.

To train our ILP classifier, we divided *TVTrain* into two. The first part contains 5 days and has been used for learning the logical rules. The second part of the 2 remaining days have been used for validation. As explained earlier, the validation aims at evaluating the effectiveness of the inferred rules. We have defined 4 levels of confidence and we have used only the three highest levels during the classification phase. Rules with the lowest level of confidence have been discarded. We have also defined numeric intervals for the symbolic attributes required by ILP. For example, we have partitioned the duration domain into the following intervals:]0, 2.5 s[, [2.5 s, 7.5 s[, [7.5 s, 12.5 s[, [12.5 s, 17.5 s[, [17.5 s, 22.5 s[, [22.5 s, 27.5 s[, [27.5 s, 32.5 s[, [32.5 s, 37.5 s[, [37.5 s, 42.5 s[, [42.5 s, 75 s[, [75 s, 1 m 45 s[, [1 m 45 s, 2 m 45 s[, [2 m 45 s, 7 m 30 s[, [7 m 30 s, ∞[. We have chosen these intervals because they are centered around multiples of 5 that characterize the interprograms.

This training phase has created a set of 333 rules: 70 rules associated with the highest level of confidence, 96 rules with the second level of confidence, 16 rules with the third level of confidence and 151 rules with the lowest level of confidence. We have added one prior knowledge rule that states that "*if a segment A lasts more than 5 minutes, then A is a long program segment*".

Precision and recall measures have been used to evaluate the results. They have been computed on a duration basis. The precision P_{class} is here the total duration of segments classified as long program segments that are effectively parts of long programs in the ground-truth, divided by the the total duration of segments classified as long program segments. The recall R_{class} is in the same way the total duration of segments classified as long program segments that are effectively parts of long programs, divided by the total duration of long programs in the ground-truth.

The results contain also the total number of computed segments for each processed day, the total number of segments classified as long program segments and the total number of long program segments in the ground-truth.

Table 6 summarizes the obtained results. Both the precision and the recall are expressed in percentages; they are both very high.

In order to put into perspective the obtained results, we have calculated the score that a naive solution could

Table 6: Evaluation of the ILP-based technique in detecting long program segments on *TVTest*.

Days	1	2	3	4	5	6	7
Precision P_{class}	98.99	98.86	97.57	98.80	99.20	97.62	99.14
Recall R_{class}	97.69	96.58	95.43	94.60	95.57	97.57	98.22
No. of computed segments	674	681	727	712	739	602	500
No. of detected long program segments	136	121	143	134	154	131	99
No. of ground-truth program segments	40	30	42	39	36	42	35

obtained. The naive solution consists of applying a simple classification rule that classifies all the segments as long program segments. We have measured an average precision P_{class} of 88.09% with a recall R_{class} obviously equals to 100%. This naive solution classifies as long programs on average, each day, about 2 hours and 53 minutes of interprograms or short programs. However, our system reaches a precision of 98.59%. It is wrongly classified as long program segments only about 17 minutes each day.

In order to understand how these 17 minutes impact on the accuracy of extracting long programs, it is important to precisely study where they are located. This is presented in the next and last experiment.

We can also notice from Table 6 that the number of detected program segments is very high with respect to the number of actual program segments in the ground-truth. This is due to an over segmentation that is easily dealt with using a merge procedure. This procedure properly fuses consecutive program segments that belong to the same program during the labeling step.

4.5. TV Program Extraction. Finally, we present the experiment that evaluates the final step of our system, that is program extraction. We have conducted this experiment through a TVoD application. TVoD is a novel service that fulfills the needs of the users to make use of the huge and continuously growing audio-visual content without any constraint on time. It makes watching previously broadcasted TV programs possible at anytime and anywhere. In order to make TV programs available, they must firstly be automatically extracted and stored in a catalog. We evaluate here this automatic TV program extraction.

Despite the very good performance of previous processing steps, the labeling of detected long program segments and the extraction of long programs have to cope with many issues that have been revealed by the evaluation experiments of the previous steps. Not all inter-program segments and short program segments are isolated by the previous processing steps. Some of them could have been misclassified and then considered as long program segments. Moreover, long programs could have been over segmented.

TV program extraction also depends on the metadata information provided by TV channels. These metadata are required to be able to give names to the automatically extracted long program segments. To handle the most complete and accurate metadata, we have chosen to merge the EIT with the EPG. Based on the results in [2], EIT are more reliable than EPG. Therefore, we have completed the EIT with the EPG when the EIT informations were not available.

This experiment evaluates the quality of the final extracted programs. It evaluates the effectiveness of our labeling and program extraction techniques and also its ability to deal with the limitations of the previous steps and the imprecision of metadata.

In order to evaluate the quality of extracted long programs, we have first counted the number of programs in the ground-truth, the number of programs mentioned in the metadata and the number of extracted programs by DealTV. We have then counted the number of programs from metadata that are effectively in the ground-truth and the number of extracted programs that are also effectively in the ground-truth. It is worth mentioning that the number of extracted programs is always less or equal to the number of programs in the metadata. This is due to the fact that metadata is used to label automatically extracted programs and that for the evaluation, we keep only labeled programs. The obtained results on *TVTest* are summarized in Table 7. They are presented per day interval.

The accuracy of extracted programs has also been evaluated. The start (resp. end) of each extracted program has been compared to the actual start (resp. end) given by the ground-truth. The temporal difference is summed up and averaged for all the extracted programs. It is referred to as the imprecision. This imprecision has also been computed for metadata programs. The obtained results on *TVTest* are presented in Table 8.

From Tables 7 and 8, we can notice that DealTV greatly outperforms the metadata and provides very good results. In particular, the obtained results for the intervals of the midday and the evening are very accurate.

In order to further analyze these results with respect to a TVoD service, we have selected five categories that are the most relevant for the users. These categories are: movies, in live show, series, sport and news. We have then classified extracted TV programs into these five categories and we have separately evaluated the imprecision for each category. The obtained results are presented in Table 9.

Table 9 shows that movies are very accurately extracted. This is very interesting as movies are the most relevant content for real world services, TVoD in particular. Another very important content is TV news. Results show that their starts are very accurately detected. However, their ends are sometimes missed. This is mainly due to the weather that is broadcasted right after the end of the news and that is wrongly classified as a long program segment. The results

Table 7: *TVTest*. The number of long programs in the Ground-Truth—programs mentioned in the metadata—extracted programs by DealTV.

Day intervals	No. of ground-truth	Metadata		DealTV	
	GT programs	No. of programs	No. of programs ∈ GT	# programs	No. of programs ∈ GT
Night	45	46	37	33	30
Morning	29	24	24	24	24
Midday	15	20	15	16	15
Afternoon	17	14	12	12	11
Early evening	14	13	12	13	12
Evening	27	26	24	24	24

Table 8: Accuracy of extracted programs by DealTV, comparison with metadata.

Day intervals	DealTV		Metadata	
	Start mean imprecision	End mean imprecision	Start mean imprecision	End mean imprecision
Night	8 min 10 s	9 min 27 s	9 min 57 s	10 min 19 s
Morning	5 min 16 s	3 min 32 s	8 min 25 s	9 min 46 s
Midday	16 s	1 min 37 s	2 min 49 s	7 min 52 s
Afternoon	1 min 32 s	3 min 15 s	3 min 23 s	9 min 58 s
Early evening	4 min 41	7 min 2 s	8 min 3 s	11 min 9 s
Evening	13 s	1 min 21 s	2 min 45 s	6 min 52 s

Table 9: Accuracy per category of extracted programs by DealTV from *TVTest*.

Category	Extracted	Start imprecision	End imprecision
Movies	7	1 s	5 s
In live show	1	10 s	1 s
Series	14	27 s	1 min 50 s
Sport	12	42 s	1 min 48 s
News	14	2 s	2 min 56 s

shown in Table 9 do not match the results in Table 8. The high imprecision values in Table 8 are caused by others categories.

In the previous experiment presented in Section 4.4, we have measured that about 17 minutes of TV stream are incorrectly classified each day. A careful analysis of this duration has shown that it is mainly made up of few short programs. These short programs are classified as long programs and they are hence handled as parts of programs. This reduces the overall imprecision of our automatically extracted programs. Based on the results presented in Table 8, it is likely that these mis-classified segments are located outside the midday and the evening intervals.

To sum up, the obtained results show that overall our system is able to perform an accurate and fully automatic TV structuring that greatly outperforms a metadata-based structuring.

5. Conclusion

In this paper, we have studied one aspect of the problem of audio-visual content analysis and indexing. We have focused on TV structuring that is needed to automatically and precisely extract long useful programs. These can be either archived as part of our heritage or used to build added-value novel TV services like TVoD and Catch up TV.

We have first positioned the problem and then carefully and deeply presented related works and existing solutions.

We then presented DealTV, our fully automatic system. It is based on studying repeated sequences in the TV stream in order to segment it. Segments are then classified using an ILP-based technique that makes use of the temporal relationships between segments. Finally, metadata are used to label and extract programs using simple overlapping-based criteria.

Each processing step of DealTV has been separately evaluated in order to carefully analyze its impact on the final results. The system has been proven on real TV streams to be very effective.

Future work will focus on further improving the performance of the system. In particular, short program segments must be better filtered out during the ILP-classification step. It is the main source of imprecision of extracted programs. We will also study the ability of the system to structure thematic TV streams that are collected from specialized channels, like sport channels.

References

[1] ETSI EN 300 231, "Television systems; specification of the domestic video programme delivery control system," European Standard (Telecommunications series), European Telecommunications Standards Institute, 2003.

[2] S. A. Berrani, P. Lechat, and G. Manson, "TV broadcast macro-segmentation: metadata-based vs. content-based approaches," in *Proceedings of the ACM International Conference on Image*

and Video Retrieval, pp. 325–332, Amsterdam, The Netherlands, July 2007.

[3] J. P. Poli and J. Carrive, "Television stream structuring with program guides," in *Proceedings of the IEEE International Symposium on Multimedia*, pp. 329–334, San Diego, Calif, USA, December 2006.

[4] I. Foster and R. L. Grossman, "Data integration in a bandwidth-rich world," *Communications of the ACM*, vol. 46, no. 11, pp. 51–57, 2003.

[5] A. Motro and P. Anokhin, "Fusionplex: resolution of data inconsistencies in the integration of heterogeneous information sources," *Information Fusion*, vol. 7, no. 2, pp. 176–196, 2006.

[6] J. P. Poli, "Predicting program guides for video structuring," in *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, pp. 407–411, Hong-Kong, November 2005.

[7] Z. Rasheed and M. Shah, "Detection and representation of scene in videos," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1097–1105, 2005.

[8] J. R. Kender and B. L. Yeo, "Video scene segmentation via continuous video coherence," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 367–373, Santa Barbara, Calif, USA, January 1998.

[9] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, 1999.

[10] J. Wang, L. Duan, Q. Liu, H. Lu, and J. S. Jin, "A multimodal scheme for program segmentation and representation in broadcast video streams," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 393–408, 2008.

[11] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Proceedings of the 3rd International Symposium on Music Information Retrieval*, Paris, France, 2002.

[12] A. Joly, C. Frelicot, and O. Buisson, "Content-based video copy detection in large databases: a local fingerprints statistical similarity search approach," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, pp. 505–508, Genova, Italy, September 2005.

[13] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems*, pp. 117–128, Hsin Chu, Taiwan, 2002.

[14] J. Barr, B. Bradley, and B. Hannigan, "Using digital watermarks with image signatures to mitigate the threat of the copy attack," in *Proceeedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol. 3, pp. 69–72, Hong Kong, 2003.

[15] B. Coskun and B. Sankur, "Robust video hash extraction," in *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, pp. 292–295, Vienna, Austria, 2004.

[16] X. Naturel, G. Gravier, and P. Gros, "Fast structuring of large television streams using program guides," in *Proceedings of the 4th International Workshop on Adaptive Multimedia Retrieval*, pp. 223–232, Geneva, Switzerland, March 2006.

[17] R. Lienhart, C. Kuhmunch, and W. Effelsberg, "On the detection and recognition of television commercials," in *Proceedings od the IEEE International Conference on Multimedia Computing and Systems*, pp. 509–516, Ottawa, Canada, November 1997.

[18] J. Gauch and A. Shivadas, "Identification of new commercials using repeated video sequence detection," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 1252–1255, Genova, Italy, 2005.

[19] J. M. Gauch and A. Shivadas, "Finding and identifying unknown commercials using repeated video sequence detection," *Computer Vision and Image Understanding*, vol. 103, no. 1, pp. 80–88, 2006.

[20] M. Covell, S. Baluja, and M. Fink, "Advertisement detection and replacement using acoustic and visual repetition," in *Proceedings of the 8th IEEE International Workshop on Multimedia Signal Processing*, pp. 461–466, Victoria, Canada, 2006.

[21] C. Herley, "ARGOS: automatically extracting repeating objects from multimedia streams," *IEEE Transactions on Multimedia*, vol. 8, no. 1, pp. 115–129, 2006.

[22] A. Albiol, M. J. Fulla, A. Albiol, and L. Torres, "Detection of TV commercials," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 3, pp. 541–544, Montreal, Canada, 2004.

[23] S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," in *Proceedings of the 3rd ACM International Conference on Multimedia*, pp. 295–304, San Francisco, Calif, USA, 1995.

[24] A. G. Hauptmann and M. J. Witbrock, "Story segmentation and detection of commercials in broadcast news video," in *Proceedings of the IEEE Forum on Reasearch and Technology Advances in Digital Libraries*, pp. 168–179, Santa Barbara, Calif, USA, April 1998.

[25] B. T. Truong, S. Venkatesh, and C. Dorai, "Automatic genre identification for content-based video categorization," in *Proceedings of the 15th International Conference on Pattern Recognition*, pp. 4230–4233, Barcelona, Spain, 2000.

[26] S. A. Berrani, G. Manson, and P. Lechat, "A non-supervised approach for repeated sequence detection in TV broadcast streams," *Signal Processing: Image Communication*, vol. 23, no. 7, pp. 525–537, 2008.

[27] G. Manson and S. A. Berrani, "An inductive logic programming-based approach for TV stream segment classification," in *Proceedings of the IEEE International Symposium on Multimedia*, pp. 130–135, Berkeley, Calif, USA, 2008.

[28] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 103–114, Montreal, Canada, 1996.

[29] S. Muggleton, "Inverse entailment and progol," *New Generation Computing*, vol. 13, no. 3-4, pp. 245–286, 1995.

[30] A. Srinivasan, "Aleph: a learning engine for proposing hypotheses," 2007, http://web2.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 539796, 10 pages
doi:10.1155/2010/539796

Research Article

Unsupervised Segmentation Methods of TV Contents

Elie El-Khoury, Christine Sénac, and Philippe Joly

IRIT Laboratory, Toulouse University, 118 route de Narbonne, 31062 Toulouse Cedex 9, France

Correspondence should be addressed to Elie El-Khoury, elie.el-khoury@irit.fr

Received 1 September 2009; Accepted 26 March 2010

Academic Editor: Ling Shao

We present a generic algorithm to address various temporal segmentation topics of audiovisual contents such as speaker diarization, shot, or program segmentation. Based on a GLR approach, involving the ΔBIC criterion, this algorithm requires the value of only a few parameters to produce segmentation results at a desired scale and on most typical low-level features used in the field of content-based indexing. Results obtained on various corpora are of the same quality level than the ones obtained by other dedicated and state-of-the-art methods.

1. Introduction

Nowadays, due to an explosive growth of digital video content (both online and offline and available by means of public or private databases and TV broadcasts), there is an increasing of accessibility for these data. Actually, the wealth of information raises the problem of an adapted access to video content which includes heterogeneous information that can be interpreted at different granularity levels, thus leading to many profiles of requests.

Under these conditions, automatic indexing of the structure, which provides direct access to the various components of the multimedia document, becomes a fundamental issue.

For this purpose, a temporal segmentation of audiovisual is required as a preprocessing operation. Results of this segmentation may be directly used for delinearization purposes such as providing a direct access to the content itself. They can also feed other analysis algorithms aiming at producing synoptical views of the content or exploiting temporal redundancy properties inside homogeneous segments to speed up the processing time.

Basically, temporal segmentation tools work on a low-level feature (or a small set of low-level features) extracted from the content along the time. Commonly, these low-level features express meaningful properties that can be observed or processed directly from the signal, such as spectrum/cepstrum features for an audio signal or color histograms for an image. They are expressed numerically and represented through vectors whose dimensions depend on the number of those features.

Two kinds of segmentation strategies can then be applied. Some algorithms try to gather set of successive values which are supposed to belong to a same homogeneous segment. Some others are focusing on transitions detection between segments.

Such algorithms have been developed independently one with the others for different temporal segmentation problems.

Among the most addressed ones, we find the "audio turn" segmentation. An "audio turn" denotes a homogeneous audio segment related to basic semantic audio classes namely, speech, music, speech superimposed with music, ambient sounds, and silence. This is generally a preprocessing tool. This is also the case for shot segmentation algorithms. The goal here is to identify successive frames in an edited video content which are belonging to a same cinematic take. More recently, some algorithms have been also proposed for TV programs segmentation. They can be used for Electronic Program Guide (EPG) synchronization or simply to provide some entries in recordings.

All those algorithms are dedicated to specific tasks of segmentation. They are based on more or less explicit models and properties of the concepts associated to the segments or to transitions and so cannot be applied to any other segmentation tasks.

In this paper, we develop the idea that, on light processing architectures, a single operator able to produce audio turns, or shots, TV programs segmentation could be of interest if the results are of nearly the same quality than the ones obtained with dedicated tools.

So, after an overview of related works concerning video and audio segmentation methods in Section 2, we present, in Section 3, a generic unsupervised segmentation we firstly developed to process audio contents. Then we show how this segmentation method can be adapted for different granularity levels such as shot detection (Section 4), programs boundaries detection in days of television recordings (Section 5), and speaker segmentation (Section 6).

2. State of the Art of Video and Audio Segmentations

2.1. Video Segmentation. Video segmentation has been studied extensively. Traditionally, a four-layer hierarchical structure is adopted for video structure analysis which consists of a frame layer, a shot layer, a sequence layer, and a program layer. At the bottom of the structure, continuous image frames taken by a single camera are grouped into shots. A series of related shots are then grouped into a sequence. Shot change detection is performed as the first step for video segmentation by most previous approaches.

2.1.1. Video Shot Boundaries Detection. Historically, the first studied video segmentation task is the shot boundaries detection which aims at breaking the massive volume of video into smaller chunks. Shots are concatenated by editing effects such as hard cuts, fades, dissolves, and wipes. A reliable shot detection algorithm should identify such short breaks.

Because it is not an easy task, quite a lot of approaches were proposed in the literature [1–3]. See for example the report of TRECVid for a review and a comparison of state-of-the-art systems and [4] for an overview of the methods and evaluation of shot-segmentation algorithms.

Among the classical algorithms, one can cite the color histogram differences used to detect hard cuts, the standard deviation of pixel intensities for fade cuts, the edge-based contrast for dissolve cuts, and the edge-change ratio for hard, fade, and dissolve cuts. Parameters are often chosen in order to describe color or luminous intensity of the video. However the challenging problem is to distinguish shot boundaries from the following: fast object or camera motion, fast illumination changes, reflections, sudden change to explosion, and flash photography. Each potential artifact leads to develop an adhoc processing tool and explains the myriad of method.

2.1.2. Video Sequence Boundaries Detection. Some works have attempted to find an upper level structuring, mainly by grouping together adjacent shots semantically linked up in scene form [5] using more or less explicit rules used in the audiovisual production domain [6]. This task is tricky for movies because it obeys to subjective criterions. Furthermore it is difficult to process heterogeneous corpora, but scenes can be detected from special programs having a quite stable structure such as broadcast news or sports [7, 8]. But the implemented methods use a lot of many a priori knowledge.

Reference [9] presented an approach with no models or decision rules to define "story units" according to the following method When two shots are very similar and nearby temporally, they are grouped together with all the intermediary shots in order to form a segment. To calculate the similarity between two shots, intensity histograms of keyframes are used. Similar shots are grouped together in graph nodes form, and nodes are linked up when the two corresponding groups are temporally adjacent. Then segments are produced by partition of the graph, deleting the more minor links.

2.1.3. Program Boundaries Detection. Very few researches have been done for program boundaries detection on TV broadcast. Let us mention here those published by Liang et al. [10], Poli and Carrive [11] and Naturel et al. [12]. Here, a "program" must be understand as a regular television program such as news, weather broadcast, talk show, sports, or sitcoms.

Poli proposes to predict forthcoming TV programs by modelling the past ones in order to boil down the television stream structuring problem to a simple alignment one with the EPG thanks to Hidden Markov Models trained on data about television schedules collected over a full year. The stream is first segmented to find boundaries of programs which are labeled later.

In the same time, Naturel proposes a fast structuring of large TV streams using also program guides to label the detected programs. The method for segmenting a TV stream which is built on the detection of nonprogram segments (such as commercial breaks) uses two kinds of independent information. The first one is a monochromatic frame and silence detector appearing between commercials on French TV. The second kind of information comes from a duplicate detection process. Nonprograms are detected in this way because they are usually broadcasted several times and so already present in a labeled reference video dataset.

Liang proposes a less ambitious work, closer to our proposition, as it only detects programs boundaries without labeling them. He supposes that TV videos have two intrinsic characteristics. On one hand, for a TV channel, programs appear and end at a relatively fixed time every day. On the other hand, for programs of the same type, they have stable or similar starting and ending clips even when they appear in different days. As such, the approach consists of two steps: model construction and program segmentation. The program boundary models for the selected TV channel are constructed by detecting the repeating shots on different days. Then, based on the obtained models, videos recorded from the same TV channel can be segmented into programs. This approach is not valid for more complex streams and can not take into account any possible change of TV schedule.

2.2. Audio Segmentation. Auditory scene segmentation is an important step in the process of high-level semantic inference from audio streams, and in particular, a prerequisite for auditory scene categorization.

As opposed to single modal audio (e.g., pure speech in the context of speech recognition task), composite audio of multimedia databases usually contains multiple audio modalities such as speech, music and various audio effects, which can be mixed.

This is why, in the audio indexing context, first works were focusing on music/speech discrimination obtained directly with a set of low-level characteristics [13] or using multi-Gaussian models learned with huge corpora [14].

Other segmentation methods identify key sounds, such as whistle sound, crowd noise, or commentator voice in a soccer broadcast [15]. Here again, segmentation is possible only with the use of a priori knowledge.

In [16], authors first extract audio elements such as speech, music, various audio effects, and any combination of these in order to detect key audio elements and to segment the auditory scene to obtain a semantic description.

Some works, within a musical program, try to identify the musical type [17] or musical instruments used [18]. In [19] a microsegmentation of musical sequences is performed by detecting onsets of notes and percussive events.

This vast number of audio segmentation and classification methods is due on the one hand to heterogeneousness of the content and on the other hand to the aimed semantic level.

Reference [5] presents a method to detect the different audio scenes without a priori knowledge. An audio scene change occurs when a majority of the sources present in the data change. The dominant sources are assumed to possess stationary properties that can be characterized using a few features extracted from the signal [20]. In order to detect scene change, a local correlation function is then used.

2.3. Knowledge-Free and Generic Methods. We have seen that nearly all methods of audio or video segmentation perform with a priori knowledge. These approaches are based on a spatial-temporal modelling of the content and use decision rules. Currently, it is the only way to reach the semantic quality required by search engines. But only recording collections highly structured, such as broadcast videos of news and sports programmes, and homogeneous in terms of production can benefit from such methods. Furthermore, model or decision rules-based methods are limited because, for each new collection, they need either a new learning or a new expertise and often new tools have to be defined.

However some generic methods exist, permitting to process both audio and video documents. Foote and Cooper [21] show that a similarity matrix applied to well-chosen features allows a visual representation of the structural information of a video or audio signal. The similarity matrix can be analyzed to find structure boundaries. Generally, the boundary between two coherent segments produces a checkerboard pattern. The two segments will exhibit high within-segment similarity, producing adjacent square regions of high similarity along the main diagonal of the matrix. The two segments will also produce rectangular regions of low between-segment similarity off the main diagonal. The boundary is the crux of this checkerboard pattern.

A supplementary approach was developed by Haidar et al. [22]. This approach is also generic because it is independent of the size and of the type of the document. Several similarity matrices, each one representing one feature, are accumulated, and the resultant matrix shows the temporal areas homogeneous in terms of the set of the different used features. But automatically inferring a document structure from such a matrix is not easy.

3. The Proposed Segmentation Method

Contrary to the methods seen in the above section, we present a priori knowledge-free segmentation approach relaying mainly on the hypothesis that it is possible to segment at some different granularity levels any audiovisual document and that is equivalent to segment into homogeneous segments at the adequate scale.

The segmentation we propose was firstly designed for audio segmentation in the context of speaker diarization [23]. Because the traditionally used metric approaches (symmetric Kullback-Leibler, Hotteling's T2-Statistic) did not give us sufficient results in presence of multiple simultaneous audio sources, we turned towards approaches based on model selection like the Generalized Likelihood Ratio (GLR) [24] and the Bayesian Information Criterion (BIC) [25]. Though the results are better, we observed that usual GLR and BIC methods present some weaknesses: too many parameters are required to tune the algorithm, and a bad precision is obtained in detecting boundaries when segments are small.

So, we propose some improvements to the general algorithm described hereafter.

3.1. Overview of the Basic Segmentation Algorithm. The basic method for detecting a change between homogenous zones is the GLR applied on a temporal signal in which each sample is a vector of several low-level features.

For genericity reasons, we will describe this method using an unknown signal that may be an acoustic signal, a video signal or an audiovisual signal.

Let $X = x_1, \ldots, x_{N_x}$ be the sequence of observation vectors of dimension d to be modeled, M the estimated parametrical model, and $L(X, M)$ the likelihood function. The GLR introduced by Gish et al. [24] considers the two following hypotheses.

H_0: This hypothesis assumes that the sequence X corresponds to only one homogeneous segment (in the case of audio signal, it corresponds to only one audio source). Thus, the sequence is modeled by only one multi-Gaussian distribution

$$(x_1, \ldots, x_{N_x}) \sim N(\mu_X, \sigma_X). \quad (1)$$

H_1: This hypothesis assumes that the sequence X corresponds to two different homogeneous segments $X_1 = x_1, \ldots, x_i$ and $X_1 = x_{i+1}, \ldots, x_{N_x}$ (in the case of audio signal, it corresponds to two different audio sources or more particularly to two different speakers). Thus, the sequence is modelled by two multi-Gaussian distributions

$$\begin{aligned}(x_1, \ldots, x_i) &\sim N(\mu_{X_1}, \sigma_{X_1}),\\ (x_{i+1}, \ldots, x_N) &\sim N(\mu_{X_2}, \sigma_{X_2}).\end{aligned} \tag{2}$$

The generalized likelihood ratio between the hypothesis H_0 and the hypothesis H_1 is given by

$$\mathrm{GLR} = \frac{P(H_0)}{P(H_1)}. \tag{3}$$

In terms of likelihood, this expression becomes

$$\mathrm{GLR} = \frac{L(X, M)}{L(X_1, M_1)L(X_2, M_2)}. \tag{4}$$

If this ratio is lower than a certain threshold Thr, we can say that H_1 is more probable, so a point of change in the signal is detected.

By passing through the log

$$R(i) = -\log \mathrm{GLR} \tag{5}$$

and by considering that the models are Gaussian, we obtain

$$R(i) = \frac{N_X}{2}\log|\Sigma_X| - \frac{N_{X_1}}{2}\log|\Sigma_{X_1}| - \frac{N_{X_2}}{2}\log|\Sigma_{X_2}|, \tag{6}$$

where Σ_X, Σ_{X_1}, and Σ_{X_2} are the covariance matrices of X, X_1, and X_2 and N_X, N_{X_1}, and N_{X_2}, are, respectively the number of the acoustic vectors of X, X_1, and X_2.

Thus, the estimated value of the point of change by maximum likelihood is given by

$$\hat{i} = \arg\max_i R(i). \tag{7}$$

If $\hat{i}$ is higher than the threshold $T = -\log \mathrm{Thr}$, a point of speaker change is detected. The major disadvantage resides in the presence of the threshold T that depends on the data. That is why, Rissanen [26] introduced the Bayesian Information Criterion (BIC).

3.1.1. Bayesian Information Criterion. For a given model M, the BIC is expressed by

$$\mathrm{BIC}(M) = \log L(X, M) - \frac{\lambda}{2} n \log N_X, \tag{8}$$

where n denotes the number of the observation vectors of the model. The first term reflects the adjustment of the model to the data, and the second term corresponds to the complexity of the data. λ is a penalty coefficient theoretically equal to 1. [26].

The hypotheses test of (3) can be viewed as the comparison between two models: a model of data with two Gaussian distributions (H_1) and a model of data with only one Gaussian distribution (H_0). The subtraction of BIC expressions related to those two models is

$$\Delta\mathrm{BIC}(i) = R(i) - \lambda P, \tag{9}$$

where the log-likelihood ratio $R(i)$ is already defined in (6) and the complexity term P is given by

$$P = \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right)\log N_X, \tag{10}$$

d being the dimension of the feature vectors.

The BIC can be also viewed as the thresholding of the log-likelihood distance with an automatic threshold equal to λP.

Thus if $\Delta\mathrm{BIC}(i)$ is positive, the hypothesis H_1 is privileged (two different speakers). There is a change if

$$\left\{\max_i \Delta\mathrm{BIC}(i) \geq 0\right\}. \tag{11}$$

The estimated value of the point of change can also be expressed by

$$\hat{i} = \arg\max_i \Delta\mathrm{BIC}(i). \tag{12}$$

A well-known BIC segmentation method was proposed by Sivakumaran et al. to detect multipoints changes in audio recordings [27]. In our work we applied this method, and we figure out some limitations. Although the amount of parameters to be tuned is important, the penalty coefficient is not as stable as expected, and there is a possible cumulative error due to the sequential segmentation process: if a point is erroneously detected, the next point might be affected by this error and might not be detected correctly. All those limitations encouraged us to propose a new segmentation based on GLR and BIC.

3.2. Proposed Improvements. The proposed method for signal segmentation follows four main steps as explained below and in Figure 1.

Row a is a time line on which the expected segmentation points are shown.

(1) This time line is split into fixed size temporal windows (shown on row b) of duration d. On each window, a GLR point detection is performed independently. Doing so, one potential segmentation point P_i^0 on each window W_i^0 is so obtained. Row c shows these intermediate results. At this step, actual segmentation points closed to a temporal window boundary have a poor probability to be detected. Furthermore, some of the candidate segmentation points may have only a local significance in their temporal window but not at a larger scale.

(2) To overcome these problems, we now consider overlapping temporal windows whose boundaries correspond to one candidate segmentation point over two, obtained during the previous step of the process.

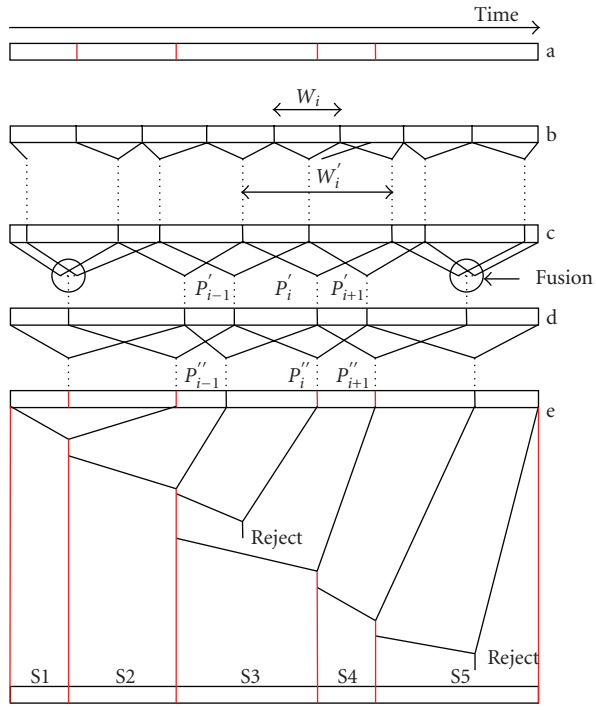


Figure 1: The proposed segmentation.

At the first iteration of the process, the first window W_1^1 goes from the beginning to point P_2^0; the second window W_2^1 goes from point P_1^0 to P_3^0; ...; the ith window W_i^1 goes from point P_{i-1}^0 to P_{i+1}^0. On each window, a new GLR point detection is performed. We obtain after this step a new set of potential segmentation points P_i^1. More generally, we note P_i^k the GLR point detected over the window W_i^k whose temporal boundaries are defined by points P_{i-1}^{k-1} and P_{i+1}^{k-1} at the kth iteration.

(3) We then go for a readjustment step where the closest segmentation points are merged (two points are closed if the distance between them is less than 3 samples). Step (2) and step (3) are iterated until stable results are obtained, that is we look for k so that $P_i^{k+1} = P_i^k$.

(4) At the last step, all the candidate segmentation points, obtained during the last iteration, are tested against the BIC Criterion.

One may have noticed that if there is N fixed size temporal window defined at the beginning of this process, we will obtain M segmentation points at the end with $M \leq N$. This means that the size of the window must be a priori fixed at a lower value than the minimal length of segments expected as a result.

Moreover, we found that a bidirectional segmentation of the signal (i.e., both forward and backward) may be useful in some cases where the transitions between two homogeneous regions are not very discriminative (interactive acoustic regions, fade or dissolve transition effects between shots, etc.). Indeed, due to the shifted variable size window introduced in the segmentation method, processing from "left to right" may detect different points of change than processing from "right to left", and therefore, there is a chance that a missed boundary in the first direction can be detected in the other direction and vice versa.

The purpose of all those steps is to generate an as stable as possible segmentation that gives homogeneous zones in terms of features distributions.

4. Application to Shot Boundaries Detection

We formulate now the hypothesis that shots are homogeneous video segments and that we may find features that can at the same time be modelled by Gaussian distributions. If we can assume that it exists a large range of such features, the first hypothesis (shot are homogeneous segments) is far from be always observed. Some lighting effects (such as flashes) or fast moving objects are strong limitations to this hypothesis. To take this specificity into account, once we have applied the previously described segmentation algorithm, we go for a rather simple postprocessing step aiming at removing false detections generated by those kinds of effects.

In this present work, a feature vector is extracted as follows. Each image provided every 40 ms (25 images/second) is divided into 4 equal parts.

The mean values of R, G, and B colour space descriptors are computed in each part. Therefore, the feature vector of dimension d ($d = 3 \times 4$) is composed of those values.

Then, the segmentation algorithm is applied as explained in Section 2. The window size W is fixed equal to 50 feature vectors because we took the assumption that the minimum duration in which a point of change can be detected is greater than 2 seconds. The penalty coefficient λ was tuned to 3.

In order to eliminate some false alarm detections due fast motion and lighting effects, a final step of histograms comparison is applied on the detected boundaries using the Manhattan (or City-Block) distance.

Suppose the video is composed of frames $I_1, I_2, I_3, \ldots, I_n$ and the segmentation step returns the following frames $I_1, I_{k_1}, I_{k_2}, \ldots, I_n$ as boundaries. For each boundary frame I_{k_j}, we consider the window of frames $[I_{k_{j-a}}, I_{k_{j+a}}]$ (a is fixed experimentally to 6), and we compute all the Manhattan distances between the histograms of the frames I_{p-1} and I_{p+1} where $p \in [k_{j-a+1}, k_{j+a-1}]$. Then, if all the distances are lower than a threshold, the frame I_{k_j} is withdrew from the boundaries set.

4.1. Experiments

4.1.1. The Corpus. We experiment our method on the corpus of the French ARGOS campaign [28]. The content set of the ARGOS campaign was made of various TV recordings, gathering TV news programs as well as commercials, weather forecast, documentaries, and fiction.

We used the two files of development (about 1 hour) to tune parameters. Then test was performed on 10 other hours.

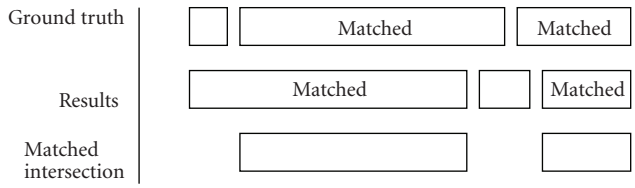


Figure 2: Identification of matched segment intersections.

Table 1: Shot boundaries detection: evaluation of the proposed system with the two types of metrics.

	Recall	Precision	F_measure
ARGOS metrics	0.935	0.931	0.933
TRECVid metrics	0.893	0.918	0.905

4.1.2. The Metrics. Two types of metrics both from ARGOS and TRECVid campaign (http://www.nlpir.gov/projects/tv2007/tv2007.html) were used. The TRECVid metric highlights the ability to localize transitions in opposition with the ARGOS metric which highlights the ability of the segmentation tool to gather units belonging to a same segment.

(a) The TRECVid Metric. This is the traditional F_measure computed from precision and recall as follows:

$$\text{precision} = \frac{\text{number\_of\_correctly\_detected\_boundaries}}{\text{total\_number\_of\_detected\_boundaries}}$$

$$\text{recall} = \frac{\text{number\_of\_correctly\_detected\_boundaries}}{\text{total\_number\_of\_boundaries\_to\_be\_detected}}$$

$$F\text{\_measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{13}$$

(b) The ARGOS Metric. The reference and system outputs are transformed into a list of continuous segments. Each segment of the ground truth is matched with the longest overlapping segment obtained as a result. A segment of the results can be matched only once. The temporal intersection between matched segments is then identified (cf. Figure 2).

Dynamic programming is used to find an optimal matching. Once the optimal matching is found, the F_measure is defined as

$$F\text{\_measure} = \frac{2 \times |\text{matched\_intersection}|}{|\text{Ground\_truth}| + |\text{Results}|}, \tag{14}$$

where $|\cdot|$ represents the overall duration of the segments.

4.1.3. Results. Table 2 shows the results (with ARGOS metric) of the proposed system compared to the average system and the best system of the campaign. We can see that our system and the best system of ARGOS give quite similar results. The method of the best system is specific to the task: it detects cuts by image comparisons after motion compensation. Then gradual transitions are detected by comparing norms of the first and second temporal derivations of the image

Table 2: Shot boundaries detection: the proposed system versus the average one and the best one at the ARGOS campaign (ARGOS metric).

	Proposed Sys.	Average Sys.	Best Sys.
ARGOS metrics	0.933	0.87	0.94

5. Application to Program Boundaries Detection

We consider here that hypotheses made for shot detection can be extended to program segmentation. It means that a selected set of features during a program behave in an homogeneous manner so that their values distribution can be modelled by a Gaussian law and that features of two consecutive programs follow two rather different Gaussian laws. The last hypothesis is that a segment is of a minimal duration (in order to fix the size of the window used at the beginning of the algorithm and to determine when fusion of boundaries must be operated—see Figure 1(d)).

In our work, the goal is to check if typical video and audio features could validate the above hypotheses.

5.1. Program Boundaries Detection Using Visual Features. Each TV program has a certain number of visual characteristics that make this program different from the others. For example, the luminance, the dominant colors, and the activity rate in a soap episode are different from those observed on a TV game or a TV News program.

As input for the system, a vector of features is originally provided as follows. Every k seconds where k denotes the approximate value of the most frequent shot duration in seconds (experimentally $k = 8$) for the tested content set, a frame is extracted and then, the three corresponding 2^m-dimension color histograms (R, G and B) are computed and their 3×2^m ($m = 8$ if the images are 8-bit images) values are concatenated in a vector. Furthermore, the Singular Value Decomposition (SVD) is applied in order to reduce the vectors dimension. Experimentally, an inertia ratio higher than 95% is reached with a vector dimension reduced to 12.

Finally, the segmentation method explained above is applied on the sequence of these 12-dimension feature vectors. Results in table 3 show a precision of about 78% on 5 days of television (120 hours).

The major errors appear when there are commercial breaks: it may be typically explained because in this type of programs, in addition to their short duration, the homogeneity hypothesis is not still verified.

The variation and the distribution of the first "video" feature (after SVD) on 3 consecutive programs are given on Figures 3 and 4. Figures 5 and 6 show the same phenomena for the third "video" feature obtained after SVD.

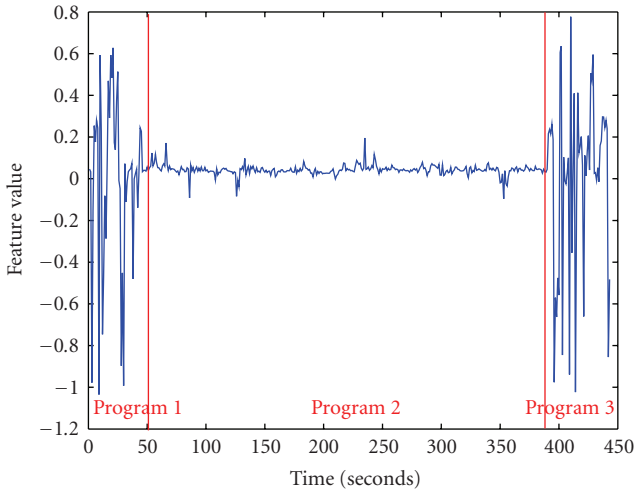


Figure 3: Variation of Feature F1 for 3 consecutive programs.

We can verify that both variation and repartition are different for the three programs.

5.2. Program Boundaries Detection Using Acoustic Features. In this subsection, we evaluate the ability of our segmentation method to detect program boundaries using only audio features.

The input feature vectors are provided as follows. The first p Mel Frequency Cepstrum Coefficients ($p = 16$) are extracted every 10 ms using a sliding window of 20 ms. Those coefficients are then normalized and quantified between 0 and $D - 1$ ($D = 48$). Every k seconds ($k = 8$), histogram vectors are computed for each MFCC coefficient and concatenated to build a super-vector of dimension $p \times D$. Then, the SVD is applied in order to reduce the dimension of those vectors. Practically, an inertia ratio of about 90% is obtained for a resulting vector dimension of 40. Finally the segmentation is applied.

Table 3 shows that scores are lower with the acoustic features (75%) than with the visual features.

5.3. Program Boundaries Detection Using Audiovisual Features. In order to exploit the complementary information brought by the two different modalities, the previous audio and video features are simultaneously used. Because we took the same temporal sampling to produce feature vectors ($k = 8$) with the same dimensions value ($3 \times 2^{\mathrm{m}} = p \times D$) reduced by the same processing (SVD, histograms) for the above two methods, it is very easy to combine them using two kinds of fusion: fusion at the decision level and fusion at the feature level.

At the decision level, the fusion was done by computing

$$\Delta \mathrm{BIC}_{\mathrm{AV}} = \Delta \mathrm{BIC}_{\mathrm{A}} + \Delta \mathrm{BIC}_{\mathrm{V}}, \tag{15}$$

where $\Delta \mathrm{BIC}_{AV}$ corresponds ideally to a change between two TV programs.

Table 3: Results of the program boundaries detection (120 hours of test).

	Visual System	Audio System	AV System
F_measure	78.04%	75.16%	80.72%

At the feature level, the early fusion aims to concatenate the visual vector of features (dimension $= 3 \times 2^m$) and the acoustic vector of features (dimension $= p \times D$). SVD is then applied: a resulting vector of dimension 60 is obtained for an inertia ratio of about 90%. Finally, the segmentation is processed to detect the frames of change. Experimentally, the early fusion at the features level gives (about 80.7%) better results than the fusion at the decision level (about 78.5%).

5.4. Experiments. Tests were carried out on 120 hours of TV videos recorded continuously from a general French TV channel during 5 days (including various kinds of programs such as news, weather forecast, talk shows, movies, sports, and sitcoms) with a rate of 25 frames/second. The size of the programs is very variable: from few minutes for weather forecast to 3 hours for a film.

For the segmentation step, we had to define the length of the fixed size window W, the penalty coefficient which depends on W, and the dimension of the feature vectors (12 for video features, 40 for audio features and 60 for audio/video features). We chose a window size of 4 minutes (corresponding to 30 vectors) as the hypothesis on the minimal duration of a program. The penalty coefficient λ was tuned to 5 for the Video system, to 1.2 for the Audio system, and to 1 for the AV system.

To evaluate those systems, the ARGOS F_measure metric, described above, was used. It highlights the ability of the segmentation tool to gather units belonging to a same segment.

Results in Table 3 show that the visual system is better (about 78%) than the audio one (about 75%). With audio features, the majority of errors appear especially when there are commercial breaks. This might be explained typically because this type of program does not follow the homogeneity hypothesis. We can see that the two modalities audio and visual bring complementary information because the results are better than those obtained with only one modality.

Many improvements can be done while taking into account some knowledge already identified in the state of the art. For example, on French TV, commercials are separated by a sequence of monochrome images (white, blue, or black). As this kind of effect can be easily detected, improvements of about 9% (F_measure = 87.34%) can easily be reached while gathering advertisements in a single program.

Comparison of the above results with those obtained by the state-of-the-art systems is a difficult task because corpora, units, and metrics are different for each experience and cannot be shared. To our knowledge, there is no international campaign addressing this topic. In this case, the evaluation we provide here should be considered as a basic reference which can be used later to evaluate improvements of this method or to compare with other future approaches.

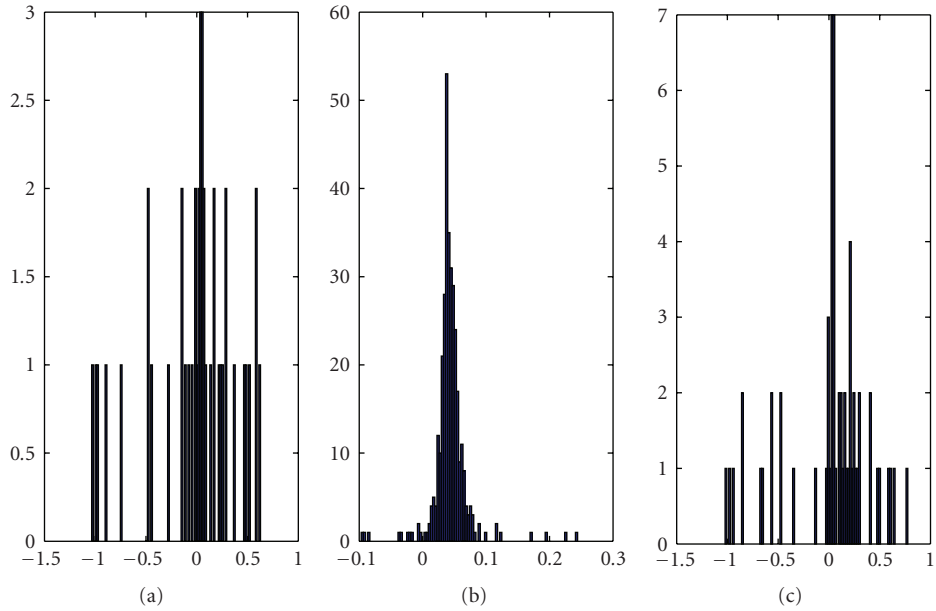


Figure 4: Distribution of Feature F1 for 3 consecutive programs.

As our system is almost knowledge-free, it can process any kinds of TV content without any prior training phase. In this way, it can be seen as a useful preprocessing step in the context of video indexing for example.

As part of the project ANR EPAC (http://www.epac.univ-lemans.fr/), the program boundaries detection was applied on 1700 hours of TV and Radio contents: the processing took less than 16 hours (lower than (recording duration $\times 10^{-2}$) with a nonoptimized version written in Matlab on a classical PC architecture.

6. Application to Speaker Diarization

In the context of speech processing on meeting data, with high interaction between speakers, one of the most difficult and unsolved problems is "speaker diarization". Speaker diarization is the process that detects speaker turns and groups those uttered by the same speaker. It is based on a first step of segmentation that consists in partitioning the regions of speech into segments: each segment must be as long as possible and must contain the speech of only one speaker. The second step is speaker clustering which consists in giving the same label to all the segments corresponding to the same speaker.

6.1. Experiments. In order to evaluate our method applied on speaker segmentation, we compare it to a well-known state-of-the-art method based only on BIC as described in [27, 29, 30].

The test set is the one used in ESTER 2009 evaluation competition (http://www.afcp-parole.org/ester/index.html). This set contains 20 shows for a total duration of about 7 hours recorded from 4 French radio stations.

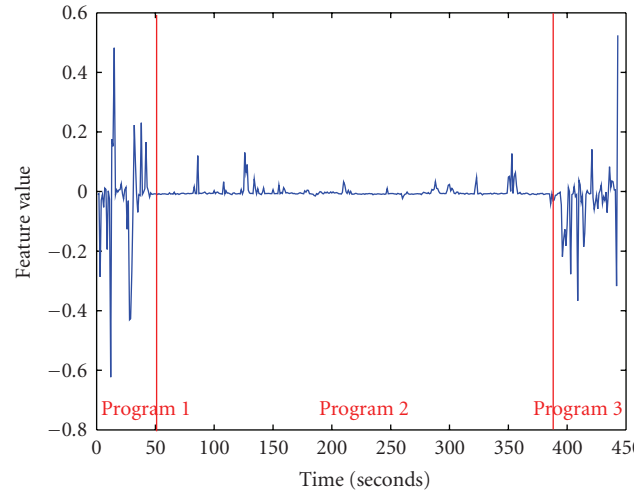


Figure 5: Variation of Feature F3 for 3 consecutive programs.

In these experiments, a centisecond approach is used that is, the soundtrack is firstly decomposed into 10 ms frames: the feature vectors used are the first 12 Mel Frequency Cepstrum Coefficients (MFCCs). The bidirectional segmentation is then directly applied on these vectors by fixing the size of the window equal to 2 seconds, and the penalty coefficient λ is tuned to 1.

As part of the speaker diarization task, the segmentation step is followed by a clustering step which consists in grouping all segments corresponding to the same speaker. The clustering step we use, presented in [31], allows adjusting the boundaries previously detected by the segmentation. Therefore, an additional improvement of 2.77% is

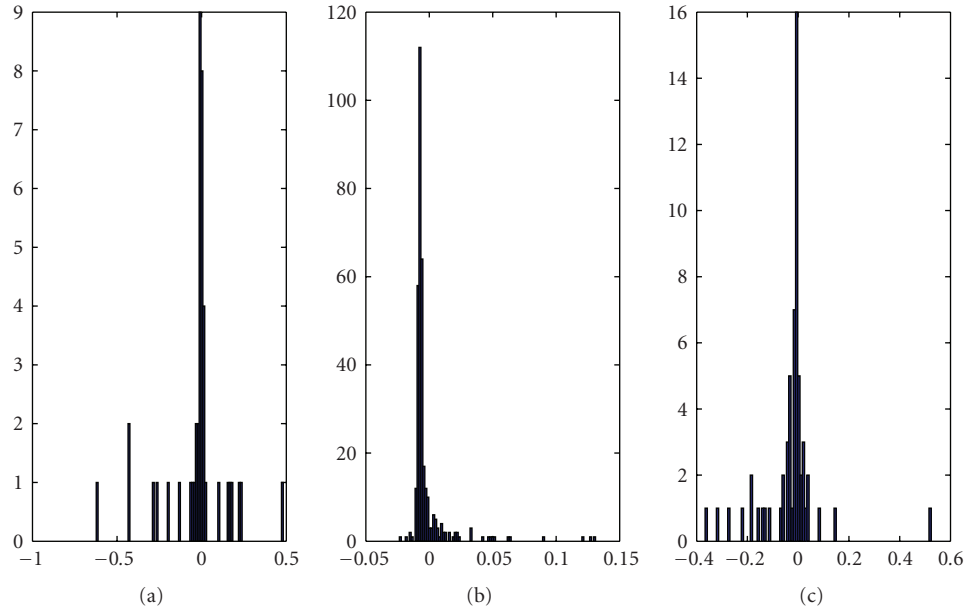


Figure 6: Distribution of Feature F3 for 3 consecutive programs.

Table 4: Evaluation of the output segments provided by the state-of-the-art segmentation method, our segmentation method, and our diarization system.

	Recall (%)	Precision (%)	F_measure (%)
State-of-the-art method	74.25	71.62	72.91
Our segmentation method	**85.36**	**85.37**	**85.37**
Our diarization method	87.95	88.26	88.10

obtained when we take into consideration this clustering process.

7. Conclusions

We have presented in this paper a temporal segmentation algorithm aiming at detecting stable boundaries between homogeneous segments. The parameters of this algorithm allow adapting it in order to address different types of segmentations problems. The size of temporal windows used at the first step of the algorithm allows controlling the size of the generated segments and the algorithm complexity. The penalty coefficient used in the DBIC criterion allows to adapt the decision sensitivity to the given problem parameters.

Applied on typical audiovisual data, the performances of this algorithm can only be compared with state-of the-art methods if we apply the same kind of postprocessing tools which are already involved in these methods (lighting effects or fast motion detection, dedicated commercial breaks detection, etc.). This algorithm can be applied on a light processing architecture (such as a set top box for example) in order to produce segmentation results on a large variety of content and for a large variety of applications.

References

[1] B. T. Truong, C. Dorai, and S. Venkatesh, "New enhancements to cut, fade, and dissolve detection processes in video segmentation," in *Proceedings of the 8th ACM International conference on Multimedia (ACM '00)*, pp. 219–227, New York, NY, USA, 2000.

[2] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 90–105, 2002.

[3] Z. Liu and Y. Wang, "Major cast detection in video using both speaker and face information," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 89–101, 2007.

[4] A. M. A. Ahmad, "Multimedia content and the semantic web: methods, standards and tools: book reviews," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 3, pp. 457–458, 2007.

[5] H. Sundaram and S.-F. Chang, "Video scene segmentation using video and audio features," in *Proceedings of the IEEE International Conference on Multi-Media and Expo (ICME '00)*, vol. 2, pp. 1145–1148, Beijing, China, 2000.

[6] P. Aigrain, P. Joly, and V. Longueville, "Medium knowledge-based macro-segmentation of video into sequences," in *Intelligent Multimedia Information Retrieval*, pp. 159–173, MIT Press, Cambridge, Mass, USA, 1997.

[7] M. Bertini, A. Del Bimbo, and P. Pala, "Content-based indexing and retrieval of TV news," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 503–516, 2001.

[8] G. Piriou, P. Bouthemy, and J.-F. Yao, "Extraction of semantic dynamic content from videos with probabilistic motion models," in *Proceedings of the 18th European Conference on Computer Vision (ECCV '04)*, vol. 3023, pp. 145–157, 2004.

[9] M. Yeung, B.-L. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation," in *Readings in Multimedia Computing and Networking*, pp. 360–369, Morgan Kaufmann Publishers, San Francisco, Claif, USA, 2001.

[10] L. Liang, H. Lu, X. Xue, and Y.-P. Tan, "Program segmentation for TV videos," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '05)*, pp. 1549–1552, Kobe, Japan, May 2005.

[11] J.-P. Poli and J. Carrive, "Modeling television schedules for television stream structuring," in *Proceedings of the 13th International Multimedia Modeling Conference (MMM '07)*, vol. 2, pp. 680–689, Singapor, 1996.

[12] X. Naturel, G. Gravier, and P. Gros, "Fast structuring of large television streams using program guides," in *Proceedings of the 4th International Workshop on Adaptive Multimedia Retrieval (AMR '06)*, vol. 4398 of *Lecture Notes in Computer Science*, pp. 222–231, Paris, France, 2007.

[13] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '97)*, vol. 2, pp. 1331–1334, Munich, Germany, 1997.

[14] J. Pinquier and R. André-Obrecht, "Audio indexing: primary components retrieval: robust classification in audio documents," *Multimedia Tools and Applications*, vol. 30, no. 3, pp. 313–330, 2006.

[15] S. Lefevre, B. Maillard, and N. Vincent, "Deux niveaux et deux outils d'analyse pour une meilleure segmentation de données audio," in *Proceedings of the 19th Colloque GRETSI sur le Traitement du Signal et des Images*, Paris, France, September 2003.

[16] L. Lu, R. Cai, and A. Hanjalic, "Audio elements based auditory scene segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. 17–20, Orlando, Fla, USA, May 2006.

[17] G. Tzanetakis and G. Essl, "Automatic musical genre classification of audio signals," in *Proceedings of the IEEE Transactions on Speech and Audio Processing*, pp. 293–302, New York, NY, USA, 2001.

[18] T. Heittola and A. Klapuri, "Locating segments with drums in music signals," Tech. Rep., Tampere University of Technology, Tampere, Finland, August 2002.

[19] O. Gillet and G. Richard, "Comparing audio and video segmentations for music videos indexing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '06)*, vol. 2, pp. 873–876, Toulouse, France, May 2006.

[20] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '96)*, vol. 2, pp. 993–996, Atlanta, Ga, USA, 1996.

[21] J. T. Foote and M. L. Cooper, "Media segmentation using self-similarity decomposition," in *Storage and Retrieval for Media Databases*, vol. 5021 of *Proceedings of SPIE*, pp. 167–175, Santa Clara, Claif, USA, January 2003.

[22] S. Haidar, P. Joly, and B. Chebaro, "Style similarity measure for video documents comparison," in *Proceedings of the 4th International Conference on Image and Video Retrieval (CIVR '05)*, vol. 3568 of *Lecture Notes in Computer Science*, Springer, Singapore, July 2005.

[23] E. El Khoury, C. Sénac, and R. André-Obrecht, "Speaker diarization: towards a more robust and portable system," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '07)*, vol. 4, pp. 489–492, Honolulu, Hawaii, USA, 2007.

[24] H. Gish, M.-H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '91)*, vol. 2, pp. 873–876, Toronto, Canada, 1991.

[25] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, vol. 2, pp. 645–648, Seattle, Wash, USA, May 1998.

[26] J. Rissanen, *Stochastic Complexity in Statistical Inquiry Theory*, vol. 2, World Scientific Publishing, River Edge, NJ, USA, 1989.

[27] P. Sivakumaran, J. Fortuna, and A. Ariyaeeinia, "On the use of the Bayesian information criterion in multiple speaker detection," in *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech '01)*, vol. 2, pp. 795–798, Aalborg, Denmark, 2001.

[28] P. Joly, J. Benois-Pineau, E. Kijak, and G. Quénot, "The ARGOS campaign: evaluation of video analysis and indexing tools," *Signal Processing: Image Communication*, vol. 22, no. 7-8, pp. 705–717, 2007.

[29] A. Tritschler and R. Gopinath, "Improved speaker segmentation and segments clustering using the Bayesian information criterion," in *Proceedings of the European Speech Processing (Eurospeech '99)*, vol. 2, pp. 679–682, Budapest, Hungary, 199.

[30] P. Delacourt, D. Kryze, and C. J. Wellekens, "DISTBIC: a speaker-based segmentation for audio data indexing," *Speech Communication*, vol. 32, no. 1, pp. 111–126, 2000.

[31] E. El-Khoury, C. Sénac, and J. Pinquier, "Improved speaker diarization system for meetings," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 4097–4100, Taipei, China, 2009.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 856761, 17 pages
doi:10.1155/2010/856761

Research Article

A Video Browsing Tool for Content Management in Postproduction

Werner Bailer, Wolfgang Weiss, Gert Kienast, Georg Thallinger, and Werner Haas

JOANNEUM RESEARCH Forschungsgesellschaft mbH, Institute of Information Systems, Steyrergasse 17, 8010 Graz, Austria

Correspondence should be addressed to Werner Bailer, werner.bailer@joanneum.at

Received 31 August 2009; Revised 24 November 2009; Accepted 17 December 2009

Academic Editor: Jungong Han

We propose an interactive video browsing tool for supporting content management and selection in postproduction. The approach is based on a process model for multimedia content abstraction. A software framework based on this process model and desktop and Web-based client applications are presented. For evaluation, we apply two TRECVID style fact finding approaches (retrieval and question answering tasks) and a user survey to the evaluation of the video browsing tool. We analyze the correlation between the results of the different methods, whether different aspects can be evaluated independently with the survey, and if a learning effect can be measured with the different methods, and we also compare the full-featured desktop and the limited Web-based user interface. The results show that the retrieval task correlates better with the user experience according to the survey. The survey rather measures the general user experience while different aspects of the usability cannot be analyzed independently.

1. Introduction

With the increasing amount of multimedia data being produced, there is growing demand for more efficient ways of supporting exploration and navigation of multimedia data. Viewing complete multimedia items in order to locate relevant segments is prohibitive even for relatively small content sets due the required user time for viewing and the amount of data that needs to be transferred. Classical search and retrieval approaches require sufficient metadata to index the content and the feasibility to formulate a query in terms of the available metadata. Manual annotation of the content yields semantically meaningful metadata that can be effectively searched by human users, but at high annotation costs. Automatic metadata extraction approaches are in many cases not able to fully capture the semantics of the content, which makes the metadata difficult to query.

Multimedia content abstraction methods are complementary to search and retrieval approaches, as they allow for exploration of an unknown content set, without the requirement to specify a query in advance. This is relevant in cases where only few metadata are available for the content set, and where the user does not know what to expect in the content set, so that she is not able to formulate a query. In order to enable the user to deal with large content sets, it has to be presented in a form which facilitates its comprehension and allows to quickly judge the relevance of segments of the content set. Media content abstraction methods will (i) support the user in quickly gaining an overview of a known or unknown content set, (ii) organize content by similarity in terms of any feature or group of features, and (iii) select representative content for subsets of the content set that can be used for visualization.

The term *video abstract* is defined in [1] as "a sequence of still or moving images presenting the content of a video in such a way that the respective target group is rapidly provided with concise information about the content while the essential message of the original is preserved". The authors of [2] use the term *video abstraction* to denote all approaches for the extraction and presentation of representative frames and for the generation of video skims. In this paper we use the term *content abstraction* to refer to all approaches that aim at providing condensed representations of segments of a single media item or a collection of items, that are relevant or salient, independent of the purpose, context, form, creation method and presentation style of the abstract. Despite their differences in all those aspects, the existing approaches for creating multimedia content abstractions share a number of

similar steps which allow to define a common process model. Based on the definition of a process model for multimedia content abstraction, we aim at defining a software framework supporting video browsing.

The application scenario is that of content management in the post-production phase of film and TV production. In post-production environments, users typically deal with large amounts of audiovisual material, such as newly shot scenes, archive material and computer generated sequences. A large portion of the material is unedited and often very redundant, for example, containing several takes of the same scene shot by a number of different cameras. Typically only few metadata annotations are available (e.g., which production, which camera, which date). The goal is to support the user in navigating and organizing these audiovisual collections, so that unusable material can be discarded, yielding a reduced set of material from one scene or location available for selection in the post-production steps.

Media production workflows are becoming increasingly flexible and distributed, involving many contributors located at different sites. This poses new challenges for managing audiovisual assets, as efficient access to content needs to be possible remotely and because people who could be consulted when looking for content are not in reach. Based on a previous desktop application for browsing audiovisual content [3] we propose a Web application that provides the same functionality, but allows to access content repositories remotely.

The contributions of this work are the following. We present a process model for multimedia content abstraction and implement a framework for video browsing based on this model. The framework can be easily extended to support browsing by any low-, mid-, or high-level feature. We present a desktop and a Web-based user interface for the framework. We evaluate both user interfaces using different evaluation approaches and compare the results between the two user interfaces and the different evaluation methods.

The rest of this paper is organized as follows. Section 2 discusses aspects of multimedia content abstraction and presents a general process model. In Section 3 we present an implementation of this process for interactive video browsing as well as a desktop and Web-based user interface in Section 5. The evaluation of the desktop and Web-based video browsing tool and its results are discussed in Section 6, and Section 7 concludes the paper.

2. Multimedia Content Abstraction

2.1. Aspects of the Problem. Multimedia content abstraction encompasses different ways of summarizing, condensing and skimming multimedia content. There are a number of aspects that discriminate these approaches proposed in literature. In the following we discuss some of them, focusing on those that influence the design of our software framework for video browsing. We do not attempt to provide a complete survey of related work here, for a comprehensive overview and comparison of video abstraction methods see for example [2].

Content abstraction can be done manually, automatically or semi-automatically (e.g., using user input to define examples of relevant content segments [4]). A basic aspect when creating the abstract is its **purpose**, which can be to objectively summarize the content conveying all of the original message or to deliberately bias the viewer (e.g., when creating a movie trailer, cf. [5]). In our case the purpose is to maximize the amount of information contained in the abstract.

Somewhat related to the purpose is the **context** of the abstract, which may be undefined and independent of the initial input of the user (e.g., when a user starts browsing), it can be defined by user input, or it can be predefined, for example, when abstracts are used for representing search results and the user's query is known [6]. Domain knowledge also contributes to the definition of the context, as it helps defining the relevance of content segments. Most video abstraction approaches for sports broadcasts exploit this knowledge (e.g., goal scenes in soccer games are relevant). The context in film and TV postproduction is given by the current production a user is working on. However, while this context is possibly defined in scripts and storyboards, it is not formalized in a way that is directly usable by content management tools.

A number of aspects are related to the media type of the content to be extracted. The **dimension** of the content may be a single media item (e.g., one video) or a collection of items (an example for the visualization of a content set is presented in [7]). In the latter case all items may be of the same or of different types (e.g., a mixed collection of still images and videos). The media type also determines whether the content set has a defined **order**. For example, a video or audio stream has an intrinsic temporal order, which is often kept in the abstract. In our case the dimension is given by the set of content related to a certain production, which is often 30 times or more of the duration of the final content.

One of the most important aspects is the **content structure**. In [8] the authors discriminate *scripted* (such as movies) from *unscripted* content (such as sports video, surveillance video, home videos). Of course, the boundaries between the two are very fuzzy. Another dimension of structure is *edited* versus *unedited* content. While some content is not intended to be edited (e.g., surveillance video), there exist rushes for both scripted and unscripted content. For edited scripted content the abstraction algorithm can attempt to detect and use the structure of the content (such as dialogs [5]), while for unscripted (and especially also unedited) content other approaches are required (e.g., [9]). Content structure does not only exist on the level of the single media item, but also on the level of the collection in the case of multiitem abstracts. In some cases the collection has a "macrostructure", such as a set of rushes produced according to a script. The content encountered in our use case is typically unedited, but depending on the production it can be structured or unstructured.

There is a big variety of approaches for the **presentation** of abstracts. It can be interactive or non-interactive,

sequential or hierarchical, and different media types and visualizations can be used. Typical ways of presentation are static visualizations of representative frames (using different visualizations such as story boards or comic book style [10]), hierarchical static or navigatable visualizations of representative frames (e.g., [11]) and video skims [12]. One aspect related to presentation is whether the abstraction system is distributed. Web-based browsing of video content has already been considered in early work on video retrieval (e.g., [13]). However, most of this work deals with browsing abstract of single video items as collections of video content were rarely accessible on the Web. In [14] authors propose search and browsing interfaces for the Open Video archive. Due to the fact that flexibility and interactivity of Web applications has been quite limited, most of these approaches are limited to static or animated key frames. Only recently it has become possible to provide the functionality available in many desktop applications for video browsing also on the Web.

Unified frameworks for multimedia content abstraction have been proposed, mostly to integrate content abstraction and retrieval (e.g., [8, 15]), but they are often limited to some types of media (e.g., only video), to only scripted or only unscripted content, or they only support certain presentation forms (such as skims [16]).

2.2. Process Model. In [3] we have proposed a multimedia content abstraction process that supports the creation of content abstractions independent of many of the aspects discussed in Section 2.1, that is, media type, context, presentation (interactive and noninteractive) and visualization, extending the generic five step process for video skimming and four step process in clustering-based approaches described in [2]. In the following, we briefly review the definition of the process and relate it to our video browsing use case.

Design. The first stage deals with the conceptualization of the content abstraction and makes basic decisions about its purpose and form. If the work is done manually, this involves a creative intervention by the user. If it is done automatically, many of these decisions may have been already taken by the developer of the application, and they are hard-wired or depend on the application's state and context. In the case of interactive video browsing for post-production, many design decisions of the abstract are taken when developing the application. The user just has control over some presentation aspects.

Clustering. In this stage, similarities within the content set are found and content segments that are related in terms of some feature are grouped. If selection has been performed before, the selected subset of content segments is used as an input, otherwise clustering is performed on the whole content set. Clustering is a key step in content abstraction, as it is crucial for the reduction of redundancy. The clustering step in the interactive video browsing tool is specific to the feature the user has selected for clustering, and each clustering step uses one feature. Content clustering and selection steps are repeated iteratively in this case.

Selection. This step selects relevant segments or groups of segments according to a defined set of criteria. If these criteria have been specified by the user or are known from the application context, the selection step can be performed before clustering (e.g., sports highlights extraction). In other cases, the selection step is performed after clustering, selecting relevant clusters instead of segments. In many automatic content abstraction processes the selection criteria are a result of clustering, for example, outliers such as unusual events in the content are found relevant to be included in the abstract (e.g., when summarizing surveillance video). The selection of content for interactive video browsing has two aspects: The decisions which kind of content to keep and which to discard are made by the user's interactions. The selection of representative content to visualize subsets of the content depend on automatic content analysis (e.g., key frame extraction) and feature specific algorithms.

Presentation. In order to visualize and/or auralize the selected groups of content segments, either new media items are created (e.g., mosaics of a shot [17], plots of a data space, time lines) or representative segments are used (e.g., representative frames for a set of video segments, a short clip). The media items representing groups of content segments are organized according to the layout of the presentation, forming a new multimedia document. The presentation in our interactive video browsing tool is a light table representation of selected key frames. The key frames can be kept in their original order or ordered by a feature value.

Consumption. If the result of the content abstraction is noninteractive (e.g., video skim, movie trailer), the consumption step only consists of viewing the document (and possibly navigation using the player controls). In the interactive case, such as in video browsing, the user selects a subset of the content segments and maybe also changes further parameters, thus altering the input for selection and clustering. The result of re-running the creation process is an updated presentation that better suits the user's needs and interests.

The basic workflow in the browsing tool, shown in Figure 1, is as follows: the user starts from the complete content set. By selecting one of the available features the content will be clustered according to this feature. Depending on the current size of the content set, a fraction of the segments (mostly a few percent or even less) is selected to represent a cluster. The user can then decide to select a subset of clusters that seems to be relevant and discard the others, or repeat clustering on the current content set using another feature. In the first case, the reduced content set is the input to the clustering step in the next iteration. The user can select relevant items at any time and drag them into the result list.

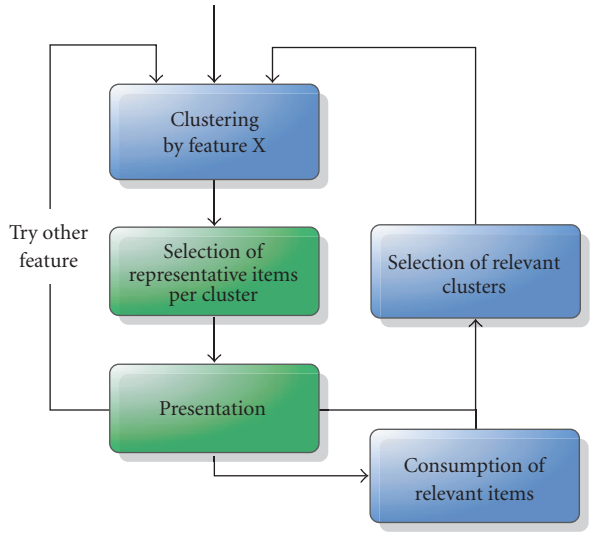


Figure 1: The basic workflow of the video browsing tool.

3. Implementing the Abstraction Process for Interactive Video Browsing

From the user's point of view, the basic difference between content browsing and search and retrieval is the limited need to know how to formulate a query and about what to expect from the content set. Thus, the content browsing tool must support the user in building a query step by step, by trying to add new restrictions and reducing the content set when applying the chain of restrictions built up so far (cf. the ostensive model of developing information needs [18]).

3.1. Process. The three core steps of the abstraction process, that is, **selection**, **clustering** and **presentation**, can be mapped to components in the software framework in a straight forward fashion. This is more difficult for the first (**design**) and last (**consumption**) step, as they are much more dependent on the specific application. In addition to the conceptual phases of the process described above there is the technical need for a **preprocessing** phase in the software implementation of the abstraction process. This step ingests material into the system, performs the required content analysis and annotation and prepares the data structures (e.g., indices) that are required by the following selection and clustering operations. The preprocessing phase is directly influenced by the design step of the process, as the decisions taken there determine the features and annotations needed and thus the content analysis operations that have to be performed. In the case of interactive video browsing, the **selection** and **clustering** steps are executed iteratively. In the **presentation** step the selection of representative media items or the creation of a visualization of a segment depends on the feature and is also tied to the clustering and selection approaches. The rest of the presentation step, as well as the consumption step, are implemented in the user interface components.

3.2. Components. This section describes the manifestation of the functionalities of the framework in software components. The framework defines interfaces for all these components. The feature specific components are implemented as plug-ins, allowing to easily add new features or change the implementation for a certain feature. Figure 2 shows the components of the framework.

The metadata repository is a transversal component for storing metadata and links to data throughout all steps of the process. The indexing service ingests content descriptions and builds additional efficient index structures. The summarizer, together with its plugins, implements the **selection** and **clustering** steps of the process for the different features.

The content analysis tools performing the actual feature extraction and producing the MPEG-7 descriptions that are imported by the indexing service as well as the features which are extracted are described in Section 4. The user interface components implementing the **presentation** step of the process are described in detail in Section 5.

3.2.1. Metadata Repository. The metadata repository is a basic infrastructure component for managing the media items under control of the video browsing application. Essence and derived essence created by automatic content analysis (such as for example representative frames) are stored in the file system. The complete metadata descriptions are stored as MPEG-7 documents in the file system as well. In addition, more efficiently searchable index structures are kept for those metadata items that are needed for clustering and selection. They are kept in a relational database. We currently use SQLite (http://www.sqlite.org), but if necessary, a more powerful database system could be integrated instead.

3.2.2. Indexing Service. The indexing service is responsible for ingest of content into the abstraction system. It does not perform content analysis itself, as legacy metadata or manual annotations might be available. The input to the indexing service are MPEG-7 descriptions conforming to the Detailed Audiovisual Profile [19]. The service watches a directory for new MPEG-7 descriptions or is triggered by a web service call. The indexing service processes the metadata descriptions and fills the index data structures. The core implementation of the service just performs feature-independent tasks such as registering new content, while plug-ins are invoked for all other tasks.

Feature specific indexing is performed by a set of indexing plug-ins. The plug-ins extract the features related information from the MPEG-7 documents and create the necessary database and/or index structure entries. A plug-in will also create additional tables in the database or index structures if they are not yet there (i.e., if the plug-in for the feature has not been used before). This increases the flexibility of the framework, as new indexing plug-ins can be easily registered with the indexing service and will be used for all further incoming documents. In order to speed up clustering in the summarizer, the indexer plug-ins for some features also create and update additional information

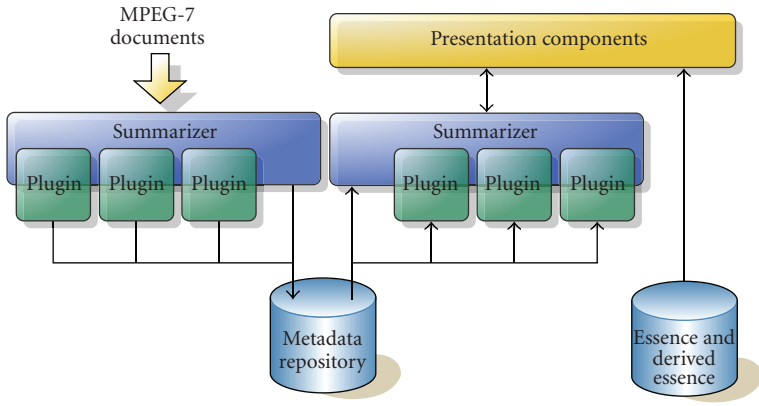


Figure 2: The components of the framework for video browsing.

such as a table of mutual similarities of descriptors in the documents indexed so far.

3.2.3. Summarizer. The summarizer is the component handling clustering, filtering and selection of representative media items. It accesses the data structures created and filled by the indexing service and has a generic interface towards the presentation layer in order to allow the use of different visualization and interaction paradigms. The functionality of the core implementation of the summarizer is mainly that of a broker, as—like in the indexing service—all feature-specific tasks are delegated to a set of plug-ins. The summarizer has a state that is defined by the current data set and its cluster structure. It also keeps a history of the clustering and selection operations carried out so far, as well as their parameters. This allows implementing undo and redo functionality in interactive applications, as well as storing the users' browsing trails in order to improve the clustering and selection algorithms.

Each plug-in provides the following functionality for one feature: the clustering algorithm and optionally algorithms for selecting a subset of the current data and for selecting/creating representative media items for a data set. The framework defines interfaces for all three types of algorithms. For the two latter ones, simple feature-independent default implementations are provided by the framework, but a plug-in can override them. It is possible to provide multiple plug-ins for one feature, for example, to experiment with different clustering algorithms.

3.2.4. Presentation Components. There are two implementations of the presentation components: a desktop application and a Web application. In the desktop application, the user interface (described in Section 5) is directly linked to the summarizer. Figure 3 illustrates the architecture of the Web application. The summarizer libraries offer their functionality to the Web-based version as Web services using gSoap (http://gsoap2.sourceforge.net). The Web-based video browsing tool is a Java Web application built with the Google Web Toolkit (http://code.google.com/webtoolkit/) and deployed in the servlet container Apache Tomcat

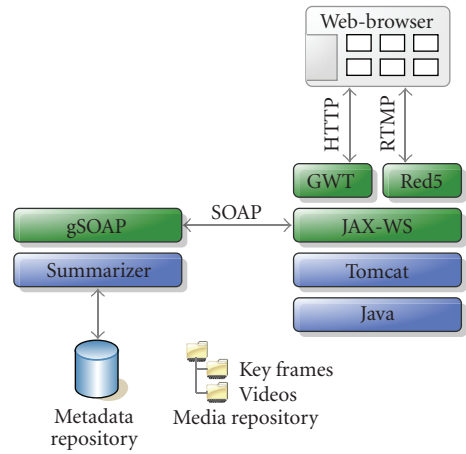


Figure 3: Architecture of the Web-based video browsing tool.

(http://tomcat.apache.org). To ensure high scalability we use default capabilities of the servlet container where each request is handled in a separate thread which can run on a own core of a processor. The Web service client is implemented with the Java API for XML Web Services (JAX-WS) (https://jax-ws.dev.java.net). Over the Web services cluster information and datasets are retrieved. The key frame images and videos are loaded directly from the media repository. The videos are streamed on demand with the Red5 (http://red5.org) Flash server to the client.

4. Feature Extraction

Feature extraction is performed by a content analysis framework using a dataflow graph approach [20]. The feature extraction is performed before ingest by the indexing service and produces one metadata description per video conforming to the MPEG-7 Detailed Audiovisual Profile [19].

First shot boundary detection and representative frame extraction are performed. The results of this step are used

as prerequisites for the extraction of other features discussed below and for visualization. As shot boundaries are natural limits of the occurrence of most visual features (such as camera movements, object occurrences), they are an important prerequisite for further visual feature extraction algorithms. For each shot, a number of representative frames is selected. The selection of representative frame positions is based on the visual activity in the material, that is, the more object and/or camera motion, the shorter the time interval between two representative frame positions.

Based on the shot structure the features described below are extracted. This are the features that are available for clustering in the browsing tool. Each of the subsections also describes the clustering algorithms that are used for the feature.

The content analysis tools support distributing content analysis tasks across different cores or machines in order to increase the throughput of the system. For the features listed below, the total processing takes about six times longer than realtime.

4.1. Camera Motion. The use of camera motion as a browsing feature is twofold: it is often used to guide the user's attention and express relevance of certain parts of the scene, for example, zooming on an object or person is an indicator of relevance, and in field sports, pans indicate the direction of the game. Secondly, it is an important selection criterion during editing, as visual grammar imposes constraints on the camera motion of sequences to be combined. The extraction algorithm (described in detail in [21]) is based on feature tracking, which is a compromise between spatially detailed motion description and runtime performance. The feature trajectories are then clustered by similarity in terms of a motion model and the cluster representing the global motion is selected. Camera motion is described on a sub-shot level. A new camera motion segment is created, when there is a significant change of the camera motion pattern (e.g., a pan stops, a zoom starts in addition to a tilt). For each of these segments, the types of motion present and a roughly quantized amount of motion are described.

We have implemented two clustering algorithms for camera motion. The first creates a fixed number of clusters, one for each type of camera motion (pan left, pan right, tilt up, tilt down, zoom in, zoom out, static). A camera motion segment is assigned to a cluster, if that type of camera motion is present in the segment, for example, if a segment contains a pan left and a zoom in, it is assigned to both clusters. The second clustering method tries to better model the actual data. Using the amounts of each type of motion, each camera motion segment is described by a vector in a three-dimensional feature space. The feature vectors are then clustered using the Mean Shift algorithm [22]. The algorithm determines the number of clusters and assigns each camera motion segment to one of them. Depending on the data, the clusters contain single camera motions or combinations and a textual label for the cluster is created (e.g., "moderate pan and strong zoom").

4.2. Visual Activity. Visual activity is a measure of the dynamics in a scene. Together with camera motion information, it is a measure for the local motion in a scene and can thus be used to discriminate quiet scenes from those with object motion. In this application we just measure the amplitude of visual change. The list of amplitude values is then median filtered to be robust against short term distortions and split into homogeneous segments. Each of these sub-shot segments is described by its average activity value. Clustering is performed using the k-means algorithm.

4.3. Audio Volume. Audio volume can for example be used to discriminate shots without any sound, calm shots of inanimate objects, interviews with a constant volume level and loud outdoor shots in city streets. As no content-based audio segmentation is available in the system, we use segments of a fixed length of 30 seconds. A list of audio volume samples is extracted for each of these segments by calculating the average volume of a 0.5 seconds time window. The list is then median filtered to be robust against short term distortions and split into homogeneous segments. Each of these sub-segments is described by its average volume value. Clustering is performed using the k-means algorithm.

4.4. Face Occurrence. The occurrence of faces is a salient feature in video content, as it allows inferring the presence of humans in the scene. The size of a face is also a hint for the role of the person, that is, a large face indicates that this person is in the center of attention. Our extractor is based on the face detection algorithm from OpenCV (http://opencvlibrary.sourceforge.net). In order to make the description more reliable and to eliminate false positives, which mostly occur for a single or a few frames, we only accept face occurrences that are stable over a longer time (we use a time window of about a second to check this). As a result we get a continuous segmentation into sub-shot segments with and without faces. There is no need for a specific clustering algorithm, as there are only two groups of segments (face and nonface).

4.5. Global Color Similarity. Global color similarity allows to group shots that depict visually similar content, for example, several takes of the same scene or different shots taken at the same location (if the foreground objects are not too dominant). To describe the color properties of a shot, the MPEG-7 ColorLayout descriptor [23] is extracted from each representative frame. The ColorLayout descriptor has the advantage of also taking the spatial color distribution of the image into account. In order to reduce the number of color descriptors to be processed, similar descriptors extracted from representative frames of the same shot are eliminated. Then the pair-wise similarities between all remaining descriptors of the content set are calculated and stored in a matrix. The similarity matrix is used as input for hierarchical clustering using the single linkage algorithm [24]. The cutoff value for the resulting tree is determined from the desired number of clusters.

4.6. Repeated Takes. In film and video production usually large amounts of raw material are shot and only a small fraction of this material is used in the final edited content. The reason for shooting that amount of material is that the same scene is often taken from different camera positions and several alternative takes for each of them are recorded, partly because of mistakes of the actors or technical failures, partly to experiment with different artistic options. The action performed in each of these takes is similar, but not identical, for example, has omissions and insertions, or object and actor positions and trajectories are slightly different. Identifying the takes belonging to the same scene and grouping them can thus significantly increase the efficiency of the work.

We use the approach proposed in [25] to identify repeated takes of the same scene. This algorithm uses a variant of the Longest Common Subsequence (LCSS) measure on a sequence of visual activity samples and color and texture features of regularly samples key frames to identify takes of the same scene. The detection results are described in the MPEG-7 document.

4.7. Multiple Views. Recently the production of multi-view video content is of growing importance, mainly driven by stereoscopic cinema. 3D television is also an emerging application area. For multi-view content the relation of clips between views is stored in the metadata description. In addition, key frames need to be extracted synchronously from all views. If necessary, the clips from different views can be automatically temporally aligned, using the method for repeated take detection [25] with a different parameterization.

The component that implements most of the support for multiview content is the indexing service. In addition plug-ins for handling multiview specific metadata have been added to the indexing service and the summarizer. The indexing service adds information about the relation of the streams to the database. Stream-specific metadata can be supported by the same indexing service plug-ins that handle single view content, while a new plug-in has been developed handling cross-stream metadata.

5. User Interfaces

The user interfaces of the desktop (Figure 4) and Web-based (Figure 5) version of the video browsing tool have been designed to be as similar as possible. As illustrated in these figures, the central component of the browsing tool's user interface is a light table (5). The light table shows the current content set and cluster structure using a number of representative frames for each of the clusters. The clusters are visualized by colored areas around the images. By clicking on an image in the light table view, a video player (a Flash video player in the case of the Web-based version) is opened and plays the segment of the video that is represented by that image. The workflow in the browsing tool is as follows: the user starts with selecting a dataset (1). By selecting one of the available features (2) the content will be clustered according to this feature (e.g., camera motion, visual activity, faces, or global color similarity). Depending on the current size of the content set, a fraction of the segments (mostly a few percent or even less) is selected to represent a cluster. The user can then decide to select a subset of clusters that seems to be relevant and discard the others (3), or repeat clustering on the current content set using another feature (2). In the first case, the reduced content set is the input to the clustering step in the next iteration. The cluster selection and the size adjustment of key frames are visualized differently in the desktop and Web-based versions. The clustered segments can be ordered by original temporal order or by the feature value.

On the left side of the application window the history (6) and the result list (7) are displayed. The history window automatically records all clustering and selection actions done by the user. By clicking on one of the entries in the history, the user can set the state of the summarizer (i.e., the content set) back to this point. The user can then choose to discard the subsequent steps and use other cluster/selection operations, or—in the desktop version—to branch the browsing path and explore the content using alternative cluster features. The result list (7) can be used to memorize video segments and to extract segments of videos for further video editing, for example, as edit decision list (EDL). The user can drag relevant key frames into the result list at any time, thus adding the corresponding segment of the content to it. The size of the images in the light table view can be changed dynamically (4) so that the user can choose between the level of detail and the number of visible images without scrolling.

In the desktop application, the temporal context of a key frame is shown by a time line of temporally adjacent key frames that is shown when the user moves the mouse over a frame. Figure 6 shows such as example in a view that is clustered by repeated takes of a scene.

In case of multiview content, the browsing tool allows clustering content by the view it originates from or by a scene shot from multiple views. The same features are also available for contextual similarity search on any of the results shown in the tool. Figure 7 shows an example of clustering multi-view content by the camera from which it has been shot, using synchronously extracted key frames from the views.

6. Evaluation

The evaluation of video browsing tools is still an open issue, with different evaluation approaches proposed in literature. In [26], we have reviewed approaches in the literature and compared different evaluation methods. Here we apply different of these methods to the desktop and Web-based version of the video browsing tool and compare their results for our video browsing tool.

6.1. Research Questions. Given the fact that there is no established method for the evaluation of multimedia browsing we have chosen to apply both TRECVID [27] style approaches as well as a survey taking the user experience into account and intend to compare their results. The *retrieval tasks* contain

Figure 4: Screenshot of the video browsing tool desktop application.

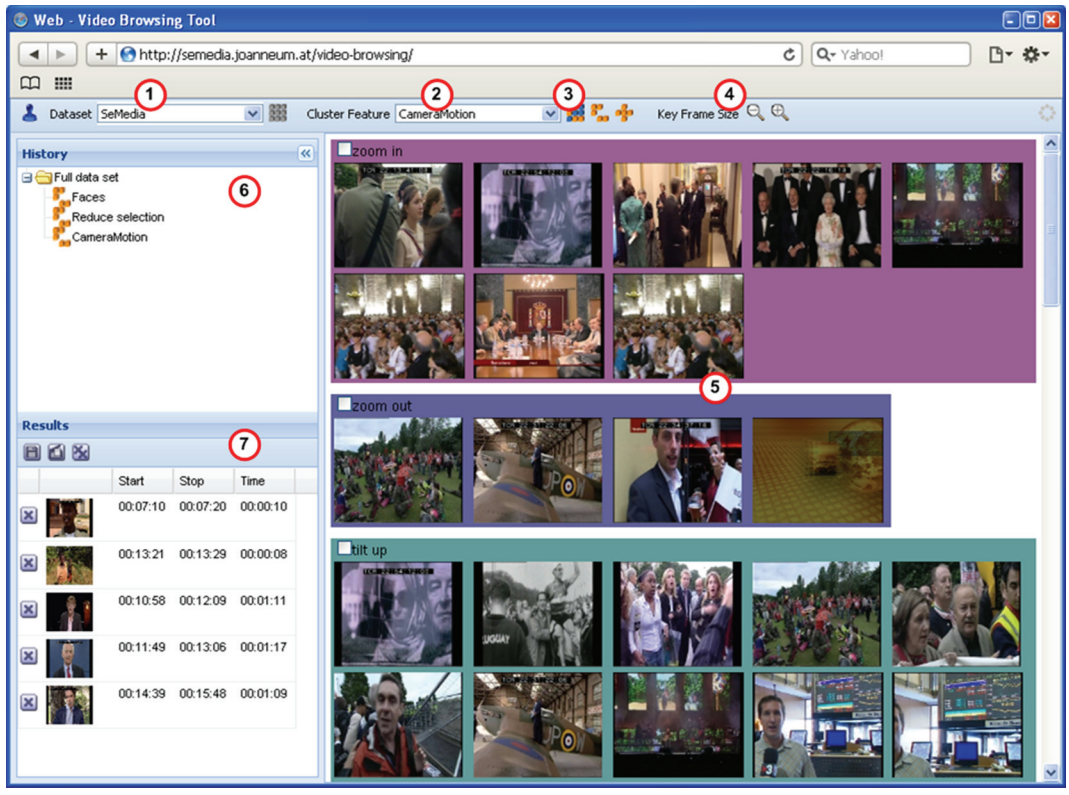


Figure 5: Screenshot of the Web-based video browsing tool.

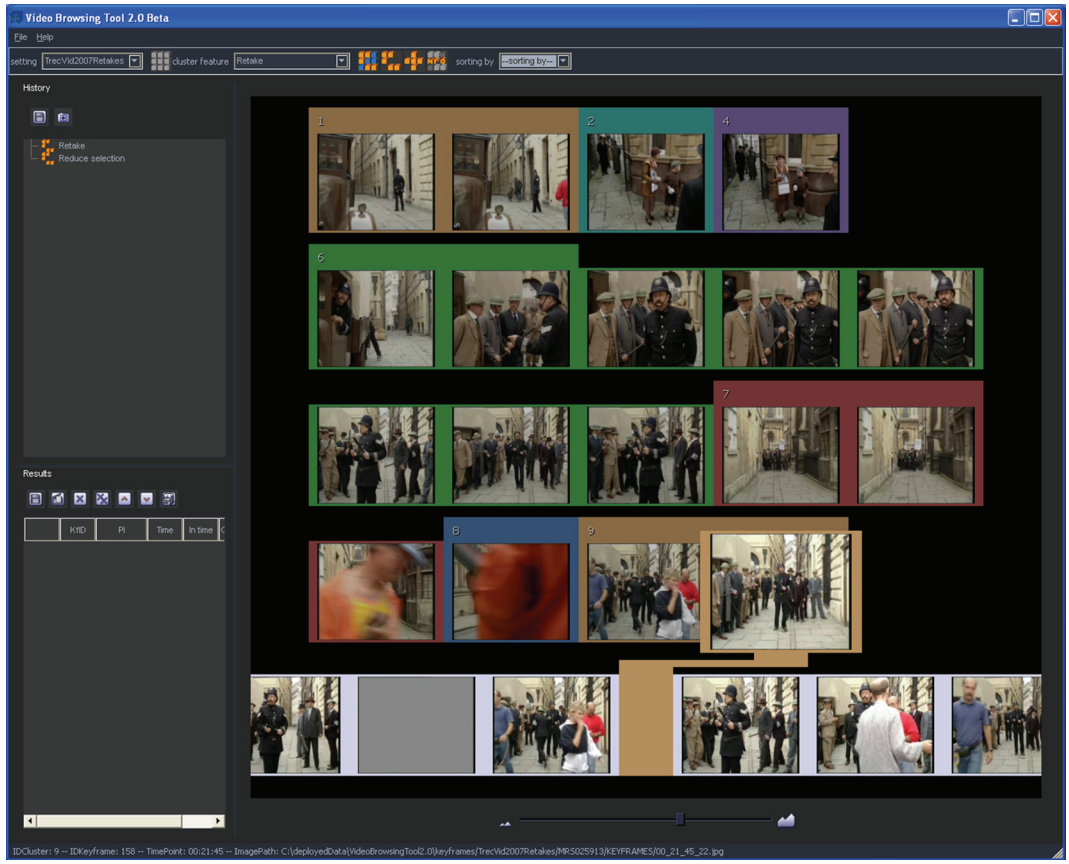


FIGURE 6: Screenshot of the video browsing tool: repeated takes. The images in the bottom row show the temporal context of the key frame under the mouse cursor.

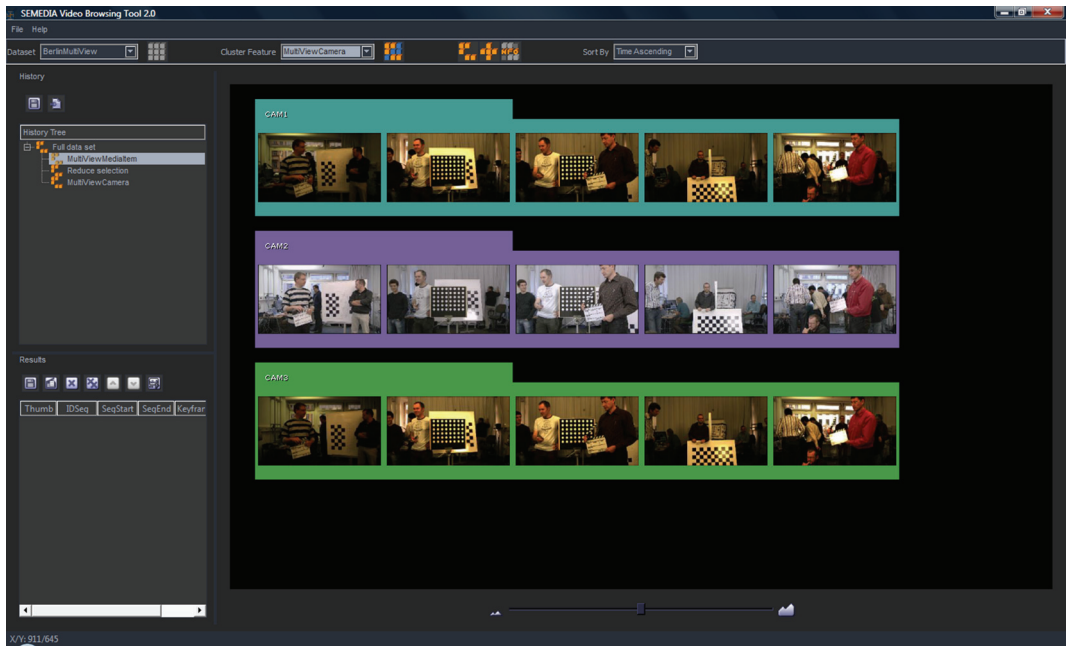


FIGURE 7: Screenshot of the video browsing tool: multi-view content.

a well defined need for video clips (motivated by scenarios in film and TV post-production), the *question answering tasks* are more goal oriented, leaving the description of the needed video clips more fuzzy. We designed corresponding pairs of retrieval and question answering tasks that target the same content sets. The *survey* asks about the experiences of the users when completing the two types of tasks. The retrieval and question answering tasks as well as the questionnaires are available at http://semedia.joanneum.at/.

In particular, we want to answer the following questions.

(i) Do different user groups achieve different results?

(ii) Is there a correlation between the results of the retrieval and question answering tasks and the assessment of the users in the post-task questions of the survey? We want to know whether the results from the different approaches yield similar or complementary results.

(iii) Can the post-task questions of the survey be answered independently? Surveys aim at giving a more holistic picture of the user experience. We are thus interested whether the questions asking about different aspects of the tool can be treated independently.

(iv) Is there a learning effect? Is it evident in the task results that users achieve better results when using the tool for some time, and does that correspond to the user's experience as expressed in the survey?

(v) Do users achieve a higher precision score when viewing the video before selecting a segment?

(vi) Is there a difference in the results achieved when using the desktop and Web-based version of the browsing tool?

The experiments to answer the first five questions are conducted with the desktop version of the browsing tool and the last question was evaluated with the Web version of the browsing tool.

6.2. Materials and Procedure. The survey is independent of the data set used. For the retrieval and question answering tasks two data sets are used. The TRECVID BBC Rushes 2006 data set is the one used in the TRECVID 2006 rushes exploitation task and consists of about 25 hours of rushes of travel documentaries (in French). The SEMEDIA data set is a part of the data collected by BBC (http://www.bbc.co.uk) and CCMA (http://www.ccma.cat) in the context of the SEMEDIA project (http://www.semedia.org) and consists of about 10 hours of edited news stories and complete news, sports and talk show programs (in English and Catalan).

There are slight differences in the evaluation procedure of the desktop application and the Web-based application. Therefore, we refer subsequently to *desktop evaluation* and *Web evaluation* to describe the differences.

6.2.1. Retrieval Tasks. Each of the retrieval tasks consists of a one line description of the synopsis of a video segment. The task is to use the browsing tool to locate all segments that match the given textual description. The results are collected in the result list of the browsing tool and saved at the end of the task. The result lists are then matched against a list of ground truth segments created before. The ground truth has been created based on the agreement of annotations from two annotators.

The retrieval tasks for the TRECVID BBC Rushes 2006 data set are the same as in the evaluation for the TRECVID 2006 rushes exploitation task described in [28], results can thus be compared.

6.2.2. Question Answering Tasks. The question answering tasks are only done in the evaluation of the desktop application. Each question answering task is a multiple choice question with six statements of which one or more are true. The question is a description of a scene, where each of the options is a statement about the scene. The questions are chosen so that they share the set of relevant video segments with a corresponding retrieval task.

For example, retrieval task 5 is *Find segments showing a football player scoring a goal*. The corresponding question is

Question 5: A football player scoring a goal . . .

(a) wears a green shirt,

(b) does so from a penalty,

(c) is shown in close up,

(d) is shown cheering at the sideline,

(e) wears a white shirt,

(f) is shown from the camera behind the goal.

6.2.3. Survey. The survey consists of three questionnaires: The pre-test questionnaire is completed once for each individual user who takes part in the evaluation after they are trained in the use of the browsing tool. The post-task questionnaire is completed after each task that each user finishes during the experiment. The questions of the post-task questionnaire are listed in Table 1. The post-test questionnaire is completed once for each participant after completing the last task. The questionnaire is largely based on the one used for the TRECVID 2004 interactive search task [29]. Some questions that were too specific to retrieval systems have been discarded and two questions specific to video browsing have been added to the post-test questionnaire.

6.2.4. Procedure. The evaluation session starts with an introduction of the browsing tool and an explanation of the evaluation procedure. Then the users have 10 minutes time for getting accustomed to using the browsing tool. Before starting to work on the tasks the users complete the pre-test part of the survey.

One evaluation session consists of a sequence of 4 retrieval tasks or a sequence of 4 question-answering tasks. The participants are evenly divided into 4 groups with varying assignment of task types and data sets in order to avoid an effect of the order on the results. In the evaluation

Table 1: Questions of the post-task questionnaire. Possible answers for each of the questions are: not at all, a little, fairly, quite a bit, very much.

TVB1	I was familiar with the topic of the query.
TVB3	I found that it was easy to find clips that are relevant.
TVB4	For this topic I had enough time to find enough clips.
TVB5	For this particular topic the tool interface allowed me to do browsing efficiently.
TVB6	For this particular topic I was satisfied with the results of the browsing.

of the Web-based version only retrieval tasks have been used. The working time for one task is 10 minutes including the time to complete the post-task questionnaire for each task. The users are allowed to ask staff for technical support about the use of the tool during the evaluation.

After the 4 tasks the users complete the posttest part of the survey. The total time for the session is thus about 60 minutes. Users can choose to do one or two sessions. In the latter case they work on a different type of task and a different data set in each of the sessions and complete only one pre-test survey in the first and one post-test survey in the second session.

6.3. Subjects. Desktop Evaluation: The tests have been performed with 19 users. In the pre-test part of the survey we have collected information about the subjects. According to the frequency the users use digital video retrieval systems, we introduced two groups: The first group consists of 11 subjects who never or rarely use digital video retrieval systems. The second group of 8 subjects represents more experienced users, who use digital video retrieval systems at least once a day.

Two thirds of the users search the Web or information systems more than once per day. More than half of the users were unfamiliar with the tool to be evaluated, only 10% were fairly or more familiar with it. Two thirds had no or little knowledge of the data sets used, only 17% were fairly or more familiar with all of the data.

Web evaluation: The evaluation of the Web-based video browsing tool has been performed with ten participants of our institute who are not involved in video browsing and have not participated in the evaluation of the desktop application. Only one subject is a little familiar with the TRECVID 2006 dataset, all others stated that they are not familiar with any of the datasets used in the evaluation. Eight of the subjects of this evaluation do not use any digital video retrieval system. Also eight subjects are not familiar with the video browsing tool (Web and desktop). Therefore it was not possible to create two user groups based on the experience of the users as in the desktop evaluation. Two people stated that they are a little familiar with the video browsing tool. Four search the Web very frequently, the others less frequently.

6.4. Results

6.4.1. Different User Groups. As we have two different user groups (experienced and inexperienced users) we want to determine whether the results for the groups are different. We try to reject the null hypothesis that the F1 measures, defined as $F1 = (2 * precision * recall)/(precision + recall)$, achieved by the two groups for the retrieval and question answering tasks have the same mean. For the retrieval tasks we have 24 samples from the inexperienced users (mean F1 0.26) and 12 from the experienced users (mean F1 0.23). For the question answering tasks we have 25 samples from the inexperienced group (mean F1 0.37) and 16 samples from the experienced user group (mean F1 0.45). We apply a two-tailed independent two-sample t-test which yields a P-value of .75 for the retrieval tasks and .48 for the question answering tasks, both at a significance level of 95%. The lower value for the question answering task seems to be mainly due to the lower number of samples. We can thus not reject the null hypothesis, that is, there are no significant differences between the two user groups.

6.4.2. Correlation between Methods. In order to compare the different evaluation approaches we analyze the correlation between the results. The assumption is that the F1 scores of a retrieval task and the corresponding question answering task are correlated, as well as the F1 measures with the answers to the questions TVB3-6 in the post-task questionnaire of the respective task (cf. Table 1).

The correlation coefficients between the F1 measures of the retrieval and question answering tasks are $r = -0.45$ and $\rho = -0.33$ (P-value .41). There is a slightly negative correlation between the results but no significant one. A t-test also shows at a significance level of 0.0001 that the two distributions have different means. We can conclude that retrieval and question answering tasks are not directly comparable, even if the users need a very similar result set to answer each of them.

A possible reason for the differences in results could be the imbalance in precision and recall. In the retrieval tasks, recall is typically lower than precision. Although users do not collect a result set in the question answering task, it can theoretically be seen as consisting of a retrieval task (collecting all necessary material) and an analysis of the collected material to answer the question. Low recall rates would of course decrease the ability to answer the question correctly. Another reason for the difference could be that users approach retrieval tasks ("collecting data") and question answering tasks ("fact finding") in a very different way.

Table 2 shows the correlation between the task results (again F1 measures) and the answers to the post-task questions of the respective task. The retrieval results are only correlated with question TVB6 at a significance level of 0.10,

that is, the user's satisfaction with the browsing result is positively correlated with the actual retrieval performance.

There is also only one strong correlation for the question answering task. The F1 measure is correlated with question TVB4 at a significance level of 0.10. But this is a negative correlation, that is, the users scored worse on the question answering tasks for which they felt to have more time. A possible explanation is that users feel stressed in cases where they encounter many video segments that match the query but think they have more time than when they only encounter few relevant ones.

6.4.3. Independence of Post-Task Questions. After each retrieval or question answering task the users answer the questions listed in Table 1. The correlation among the questions is shown in Table 3. There is a strong correlation among the questions TVB3, TVB4, TVB5 and TVB6, that can be accepted at a significance level of 0.10, in two cases even at a level of 0.01. These results show that it is difficult for the user to judge certain aspects separately (e.g., whether the tool was helpful in this case). Instead a general impression of the browsing experience is rated, including the satisfaction with the tool and with the results and the impression to have sufficient time.

The familiarity with the topic is not or only very weakly correlated with the other aspects. There is only a correlation with the perceived easiness of the task ($\rho = 0.67$ at significance level 0.10), that is, the task seems easier for users who are familiar with the topic. However, they do not feel that they have more time or achieve more satisfying results than others.

6.4.4. Learning Effect. We analyze this by looking for trends in the results achieved for the 4 tasks done in one session. As we have four different sequences of tasks, two different data sets and the tasks are done in different order by different users, the difficulty of individual tasks does not influence the trend. Figure 8 shows the scores of the post-task questions for the first through fourth task done by the participants. As expected, some of the measures (such as the familiarity with the search topic) do not show a clear pattern, but some seem to have a trend. If we fit a linear trend function to the data we get a clear trend for two of the questions: TVB4 (sufficient time, slope 0.22) and TVB6 (satisfaction with results, slope 0.19). The longer users work with the tool the higher is their satisfaction with the results and they perceive the working time as more sufficient.

The question is whether this trend can also be measured in the task results. When fitting a trend function to the results of the question answering tasks, we get slopes of -0.06 for precision and -0.04 for recall, that is, no clear trend can be seen, especially not a positive trend as in the survey answers. For the retrieval tasks the trend function for precision has a slope of 0.12 and that for recall of 0.01. The user's perception is supported by the precision values of the retrieval task, although the increase is not as strong as in the survey answers. The fact that the satisfaction is more related to precision than to recall can be explained as follows. The users know only about the video segments they have found, that is, not about the correct ones not found that would be measured by recall. Thus the perceived quality of the results depends on how well the segments in the result set match the query, which correlates with precision.

6.4.5. Higher Precision Score When Viewing the Video. Users have the option to view a video segment in the player before or after selecting it or to just drag the corresponding key frame to the result list without viewing the video. We can expect that viewing the video serves as a validation and thus the precision should be higher in cases where the video player has been used. Thus we try to reject the null hypothesis that the distribution of precision achieved without using the video player has a higher mean than that with using the video player. We have 46 samples, for 20 (44%) the video player has been used, the means are $\mu_{\text{withplayer}} = 0.58$ and $\mu_{\text{withoutplayer}} = 0.49$. We apply a independent two-sample t-test which yields a P-value of .25 for the one-tailed test, that is, we cannot reject the null hypothesis. It seems that users use the video player in cases where they are unsure while they add segments for which they are sure without using the player. Thus the precision for the segments added after viewing them is not significantly better.

6.4.6. Web Tool Evaluation. To determine whether the results of the retrieval tasks of the Web-based application and the desktop application are different, we have evaluated the series of measurements with two-tailed two-sample t-tests assuming different variances. We try to reject the null hypothesis in each test that the means of precision and recall of the Web and the desktop version are identical.

The first test (see also Table 4) consists of 38 samples of the Web evaluation and 34 samples of the desktop evaluation (users from our institute's staff). We get a P-value for the precision of .036 and a P-value for the recall of .222, both at a significance level of 0.05. Thus we can reject the null hypothesis for the precision, which means the results of the Web-based application are worse in contrast to the desktop application for this user group. On the other hand we cannot reject the null hypothesis for the recall values, which means there is no overall significant difference in this test setup.

The second test (see also Table 5) consists of 20 samples of the Web evaluation and 20 samples of the desktop evaluation (users from the SEMEDIA project). We get a P-value for the precision of .526 and a P-value for the recall of .387, both at a significance level of 0.05. This means we cannot reject the null hypothesis for both tests, that is, there are no significant differences of the Web-based application and the desktop application with this user group.

Figure 9 shows the scores of the post-task questions for the first through fourth task done by the participants at the Web evaluation. Only the satisfaction (TVB6) shows a slope of -0.1. This is in contrast to the desktop version where the users were more satisfied over the tasks (slope 0.19). All other questions (TVB1–TVB5) of this test do not show a clear pattern.

Table 2: Correlation between F1 measures of retrieval (R) and question answering (Q) task results and the post-task questionnaire (r denotes Pearson's product-moment correlation coefficient, ρ denotes Spearman's rank correlation coefficient and P the associated P-value).

		TVB1	TVB3	TVB4	TVB5	TVB6
R (F1)	r	−0.06	0.27	0.33	0.42	0.62
	ρ	−0.05	0.21	0.36	0.40	0.71
	P	.91	.61	.38	.33	.05
Q (F1)	r	0.23	−0.46	−0.80	−0.40	−0.46
	ρ	0.33	−0.41	−0.68	−0.16	−0.15
	P	.42	.31	.06	.70	.73

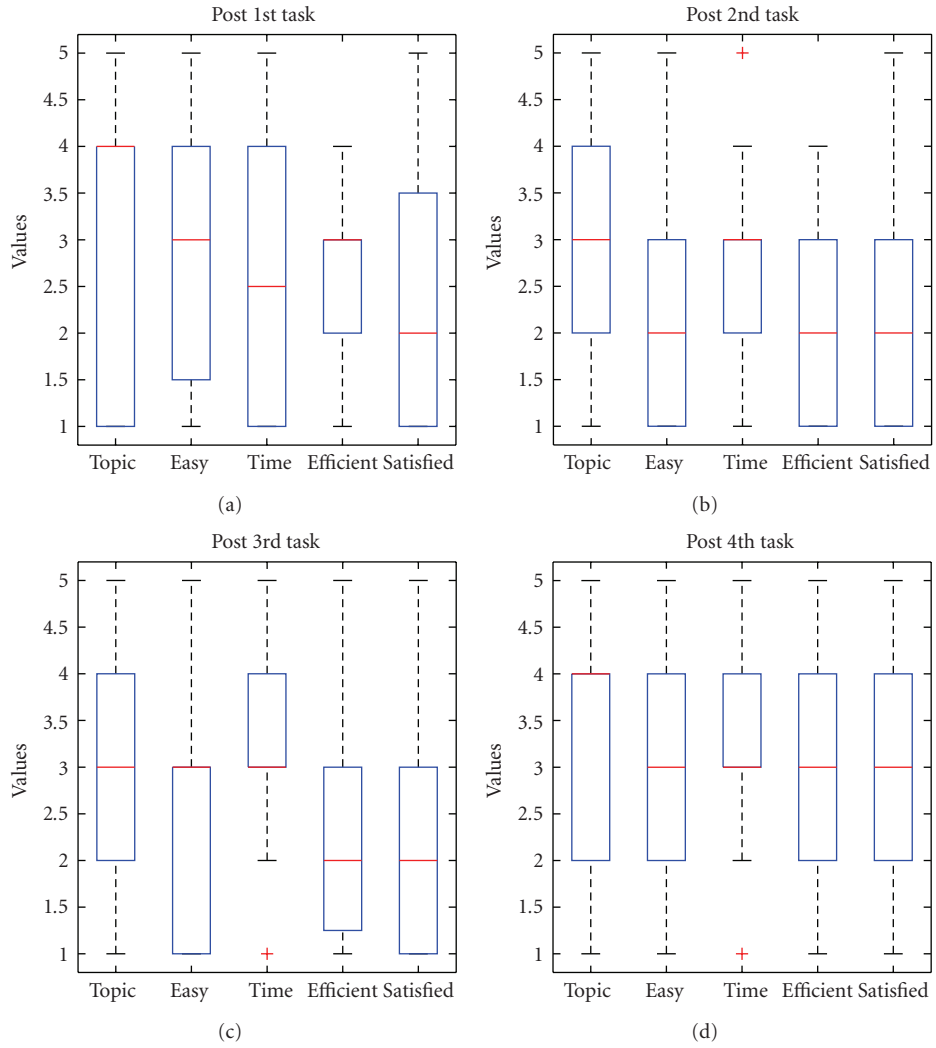


Figure 8: Results of the post-task question after each task done by the user using the desktop version.

Figure 10 illustrates the cumulated and normalized number of found items in the result list of all users during the working time. Remarkable are the tasks 1 and 6 which have both an approximating concave curve. At one third of the working time the users had about the half of the items in the result list and at the half of the working time about 70% of the items were in the result list. Furthermore, the users achieved the best results on these tasks (task 1: precision, 0.56 recall 0.30; task 6: precision, 0.65 recall 0.20).

6.4.7. Post-Test Questionnaire and User Feedback. In the post-test questionnaire we have collected information about the video browsing tool and general free text feedback of the users.

Desktop Evaluation. The half of the users stated that the response time was "fairly fast" and one third of the users are the opinion that it is "quite fast". The responses to "Learning how to use the system was easy" are as follows: not at all 4%, a little 17%, fairly 42%, quite a bit 21% and very much 13%.

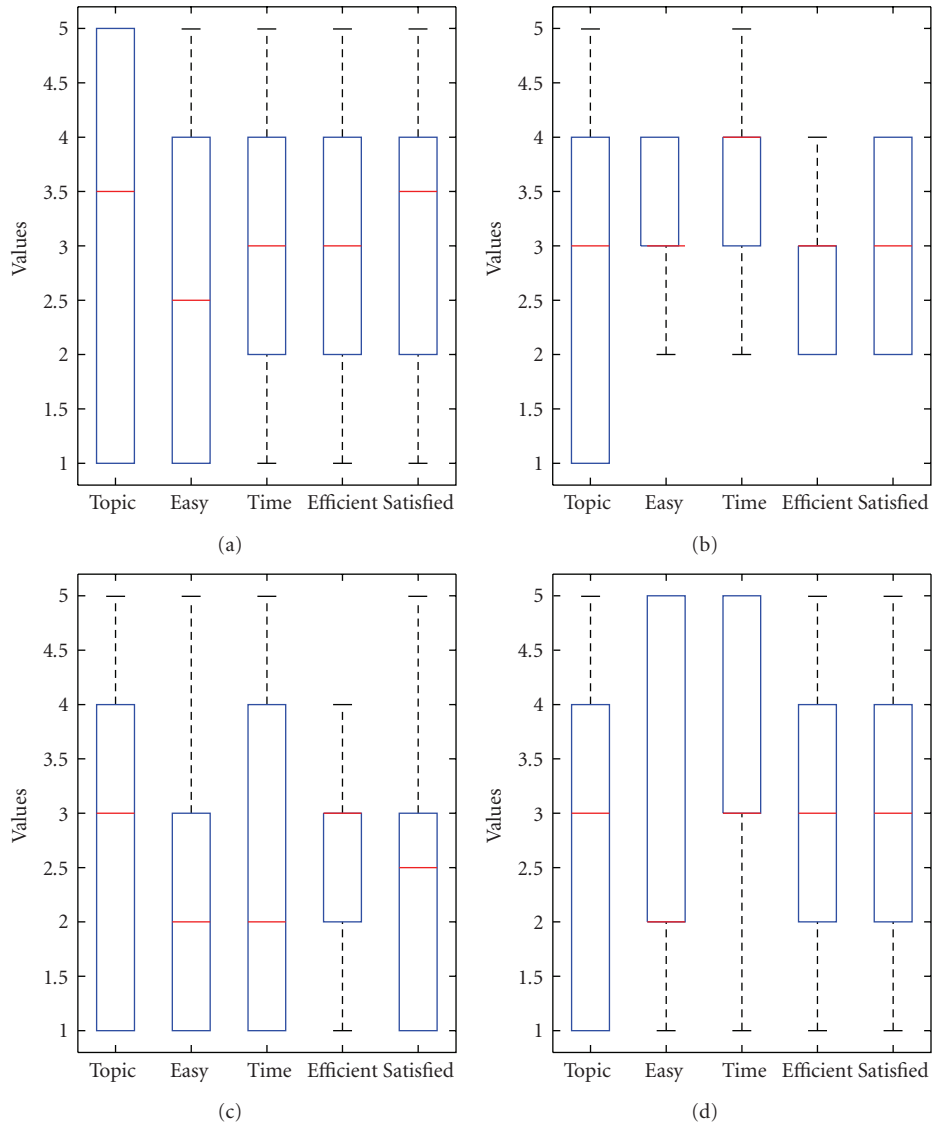


Figure 9: Results of the post-task question after each task done by the user using the Web-based version.

50% of the users answered that the system interface allowed to do the tasks efficiently is a little help, 42% answered that it is a fairly good help.

Web Evaluation. 40% of the Web users stated that the systems response time is "little fast", one third of the users choose "not fast at all" and the last third are the opinion that the system responds "fairly". The half of the users answered that it is quite easy to learn how to use the system. In contrast to the desktop version, 50% of the Web users are the opinion that the system interface "quite a bit" allows to do the retrieveal task, the other answers are: fairly 20%, a little 20% and not at all 10%.

According to the free text feedback of the users, the most annoying things of both versions are performance issues and that the cluster features sometimes produce not correct results (each 11 out of 46 answers), but 7 out of 43 answers about what users liked best of the system mention the clustering features. In 4 out of 43 answers the users wished additional features for clustering. Furthermore, 11 out of 43 answered that the interface is easy to use. Also 4 answers were about to display more metadata and information about videos, clusters and key frames.

7. Conclusion

Multimedia content abstraction approaches such as browsing applications and video summaries are of growing importance for dealing with multimedia collections. They are complementary to search and retrieval approaches and focus on problems where the formulation of a query is difficult due to the available metadata and/or the user's knowledge of the content set.

We have proposed a software implementation of a process model for multimedia content abstraction for a video browsing tool targeted at application in post-production.

Table 3: Correlation among questions of the post-task questionnaire (r denotes Pearson's product-moment correlation coefficient, ρ denotes Spearman's rank correlation coefficient and P the associated P-value). The questions TVB1 through TVB6 are listed in Table 1.

		TVB3	TVB4	TVB5	TVB6
TVB1	r	0.33	0.27	0.43	0.36
	ρ	0.67	0.50	0.54	0.33
	P	.07	.20	.17	0.43
TVB3	r		0.90	0.86	0.86
	ρ		0.86	0.86	0.64
	P		.01	.01	.08
TVB4	r			0.67	0.81
	ρ			0.64	0.74
	P			.09	.03
TVB5	r				0.84
	ρ				0.67
	P				.07

Table 4: Results of the comparison of precision (p) and recall (r) of the Web-based application and the desktop application using a two-tailed two-sample t-test assuming different variances, significance level 0.05.

	Web Evaluation		Desktop Evaluation (institute members)	
	p	r	p	r
Samples	38	38	34	34
Mean	0.342	0.156	0.525	0.206
Variance	0.11	0.026	0.148	0.032
P-value	.036	.222		

Table 5: Results of the comparison of precision (p) and recall (r) of the Web-based application and the desktop application using a two-tailed two-sample t-test assuming different variances, significance level 0.05.

	Web Evaluation		Desktop Evaluation (SEMEDIA partners)	
	p	r	p	r
Samples	20	20	20	20
Mean	0.349	0.2	0.423	0.275
Variance	0.114	0.031	0.146	0.115
P-value	.526	.387		

The browsing tool is an interactive application that allows to perform iterative clustering and selection in order to filter the content down to a manageable set of relevant items. Clustering can be performed using the features camera motion, visual activity, audio volume, face occurrences, global color similarity, repeated takes and relations in multi-view content. A number of representative frames are used to visualize a cluster. A desktop and a Web-based implementation of the client application have been presented. Concerning scalability, the tool is designed for content sets in the production workflow, which are expected to be around

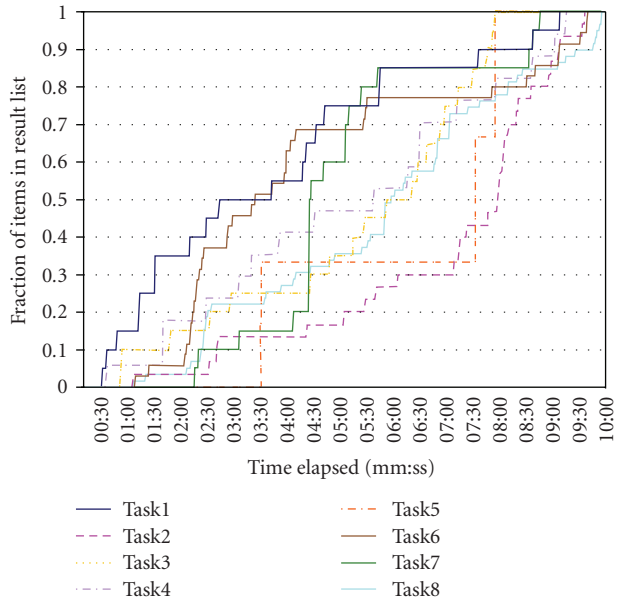


Figure 10: Cumulated and normalized number of items put into the result list of all users during the working time for the retrieval task.

100 hours per production. The response times for clustering on the complete content is not more than a few seconds for most of the features. Furthermore the increase in runtime with growing data sets is sublinear, that is, for a data set of eightfold size the time for clustering increases only by a factor between 1.3 and 3.9.

We have applied two TRECVID style fact-finding approaches and a user survey to the evaluation of a video browsing tool. We have analyzed the correlation between the results of the different methods, whether different aspects can be evaluated independently with the survey, if a learning effect can be measured with the different methods, and have compared the desktop and Web-based client applications.

In general, the results show (not unexpectedly) that especially the recall scores are rather low in such an application. This is definitely an issue that needs to be addressed in future work in video browsing.

We are also interested in comparing the different evaluation approaches for video browsing tools. We can conclude that the retrieval task correlates better with the user experience according to the survey than the question answering tasks. As retrieving relevant content is also closer to the real-world application of the tool than finding facts about the content, it seems to be the more appropriate evaluation method in this case, although it is a costly method due to the efforts for data set and ground truth preparation. Thus only retrieval tasks and a survey have been used for comparing the desktop and then Web-based client applications.

It turns out that the survey rather measures the general user experience while different aspects of the usability cannot be analyzed independently. This means that surveys are rather suitable for comparing the general usability of tools

for certain applications than for getting information about strengths and weaknesses of a certain tool.

Acknowledgments

The authors would like to thank their colleagues who contributed to the implementation of the components described here, especially Christian Schober, Harald Stiegler, and András Horti, as well as all people who took part in the evaluation sessions. The research leading to this paper has been partially supported by the European Commission under the contracts IST-2-511316-IP, "IP-RACINE—Integrated Project Research Area Cinema" (http://www.ipracine.org), FP6-045032, "SEMEDIA" (http://www.semedia.org), and FP7-215475, "2020 3D Media—Spatial Sound and Vision" (http://www.20203dmedia.eu/). BBC 2006 Rushes video is copyrighted. The BBC 2006 Rushes video used in this work is provided for research purposes by the BBC through the TREC Information Retrieval Research Collection.

References

[1] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *Journal of Visual Communication and Image Representation*, vol. 7, no. 4, pp. 345–353, 1996.

[2] B. T. Truong and S. Venkatesh, "Video abstraction: a systematic review and classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 3, no. 1, article 3, 2007.

[3] W. Bailer and G. Thallinger, "A framework for multimedia content abstraction and its application to rushes exploration," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR '07)*, pp. 146–153, Amsterdam, The Netherlands, July 2007.

[4] J. Oh and K. A. Hua, "An efficient technique for summarizing videos using visual contents," in *Proceedings of IEEE International Conference on Multi-Media and Expo (ICME '00)*, pp. 1167–1170, New York, NY, USA, 2000.

[5] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Video abstracting," *Communications of the ACM*, vol. 40, no. 12, pp. 55–62, 1997.

[6] M. G. Christel, A. G. Hauptmann, A. S. Warmack, and S. A. Crosby, "Adjustable filmstrips and skims as abstractions for a digital video library," in *Proceedings of the Forum on Research and Technology Advances in Digital Libraries (ADL '99)*, pp. 98–104, Baltimore, Md, USA, 1999.

[7] D. Ponceleon, "Hierachical brushing in a collection of video data," in *Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS '01)*, vol. 4, p. 116, IEEE Computer Society, Washington, DC, USA, 2001.

[8] Z. Xiong, R. Radhakrishnan, A. Divakaran, Y. Rui, and T. S. Huang, *A Unified Framework for Video Summarization, Browsing and Retrieval: With Applications to Consumer and Surveillance Video*, Academic Press, New York, NY, USA, 2005.

[9] P. Chiu, A. Girgensohn, W. Polak, E. Rieffel, and L. Wilcox, "A genetic algorithm for video segmentation and summarization," in *Proceedings of IEEE International Conference on Multi-Media and Expo (ICME '00)*, vol. 3, pp. 1329–1332, New York, NY, USA, 2000.

[10] S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video manga: generating semantically meaningful video summaries," in *Proceedings of the ACM International Multimedia Conference & Exhibition (ACMMM '99)*, pp. 383–392, Orlando, Fla, USA, November 1999.

[11] M. M. Yeung and B.-L. Leo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771–785, 1997.

[12] M. A. Smith and T. Kanade, "Video skimming for quick browsing based on audio and image characterization," Tech. Rep. CMU-CS-95-186, Carnegie Mellon University, Pittsburgh, Pa, USA, July 1995.

[13] J. R. Smith and S.-F. Chang, "Image and video search engine for the World Wide Web," in *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Sience and Technology (IE '97)*, vol. 3022 of *Proceedings of SPIE*, San Jose, Calif, USA, February 1997.

[14] G. Geisler, G. Marchionini, B. M. Wildemuth, et al., "Video browsing interfaces for the open video project," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI '02)*, pp. 514–515, ACM, New York, NY, USA, 2002.

[15] Y. Rui and T. S. Huang, "A unified framework for video browsing and retrieval," in *Image and Video Processing Handbook*, A. C. Bovik, Ed., pp. 705–715, Academic Press, New York, NY, USA, 2000.

[16] H. Sundaram, L. Xie, and S.-F. Chang, "A utility framework for the automatic generation of audio-visual skims," in *Proceedings of the ACM International Multimedia Conference and Exhibition (MULTIMEDIA '02)*, pp. 189–198, New York, NY, USA, 2002.

[17] M. Irani and P. Anandan, "Video indexing based on mosaic representations," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 905–921, 1998.

[18] I. Campbell and C. J. van Rijsbergen, "The ostensive model of developing information needs," in *Proceedings of the 2nd International Conference on Conceptions of Library Science (COLIS '96)*, pp. 251–268, Copenhagen, Denmark, 1996.

[19] W. Bailer and P. Schallauer, "The detailed audiovisual profile: enabling interoperability between MPEG-7 based systems," in *Proceedings of the 12th International Multi-Media Modelling Conference (MMM '06)*, H. Feng, S. Yang, and Y. Zhuang, Eds., pp. 217–224, Beijing, China, January 2006.

[20] H. Stiegler, "Module developers guide," Tech. Rep., JOANNEUM RESEARCH, Institute of Information Systems & Information Management, Graz, Austria, 2007.

[21] W. Bailer, P. Schallauer, and G. Thallinger, "JOANNEUM RESEARCH at TRECVID 2005—camera motion detection," in *Proceedings of the TRECVID Workshop*, pp. 182–189, Gaithersburg, Md, USA, November 2005.

[22] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[23] "Information technology-multimedia content description interface—part 3: visual," ISO/IEC 15938-3, 2001.

[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2001.

[25] W. Bailer, F. Lee, and G. Thallinger, "A distance measure for repeated takes of one scene," *Visual Computer*, vol. 25, no. 1, pp. 53–68, 2009.

[26] W. Bailer and H. Rehatschek, "Comparing fact finding tasks and user survey for evaluating a video browsing tool," in *Proceedings of the ACM Multimedia Conference, with Co-located Workshops and Symposiums (MM '09)*, pp. 741–744, Beijing, China, October 2009.

[27] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 321–330, Santa Barbara, Calif, USA, 2006.

[28] W. Bailer, C. Schober, and G. Thallinger, "Video content browsing based on iterative feature clustering for rushes exploitation," in *Proceedings of the TRECVID Workshop*, pp. 230–239, Gaithersburg, Md, USA, November 2006.

[29] A. Smeaton and P. Wilkins, "TRECVID 2004: Interactive search questionnaires," http://www-nlpir.nist.gov/projects/tv2004/questionnaires.html.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 836357, 20 pages
doi:10.1155/2010/836357

Research Article

Personalized Sports Video Customization Using Content and Context Analysis

Chao Liang,[1,2] Changsheng Xu,[1,2] and Hanqing Lu[1,2]

[1] *National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*
[2] *China-Singapore Institute of Digital Media, 119615, Singapore*

Correspondence should be addressed to Chao Liang, liangchao827@gmail.com

Received 2 September 2009; Revised 11 December 2009; Accepted 26 January 2010

Academic Editor: Jungong Han

We present an integrated framework on personalized sports video customization, which addresses three research issues: semantic video annotation, personalized video retrieval and summarization, and system adaptation. Sports video annotation serves as the foundation of the video customization system. To acquire detailed description of video content, external web text is adopted to align with the related sports video according to their semantic correspondence. Based on the derived semantic annotation, a user-participant multiconstraint 0/1 Knapsack model is designed to model the personalized video customization, which can unify both video retrieval and summarization with different fusion parameters. As a measure to make the system adaptive to the particular user, a social network based system adaptation algorithm is proposed to learn latent user preference implicitly. Both quantitative and qualitative experiments conducted on twelve broadcast basketball and football videos validate the effectiveness of the proposed method.

1. Introduction

The proliferation of advanced program production technology and multiple TV broadcast channels have contributed to an amazing growth of sports video content and its increasing popularity among the public. However, such increasing availability has not yet been accompanied by an improvement in its accessibility, which means that audiences can do nothing but passively watch the whole match edited by studio professionals once they choose it. Since interesting segments usually account for a small portion of the whole match; such passive watching mode not only impairs audiences' viewing experience but also wastes their time and money. To solve this problem, the ability to provide personalized video content in accordance to the features of individual viewers is of great importance.

Intuitively, viewers difference first comes through in their diverse preference towards semantic content where particular players and events are appearing in the video. For example, a Beckham's fan may mainly focus his attention on this football star than any other players, while an NBA's audience may prefer to watch the slam dunk than any other events. To meet these requirements, the source video has to be analyzed in a more refined scale and higher semantic level. More precisely, the video analysis should not merely tag some salient events with simple concepts, for example, shots in basketball or fouls in football, but annotate various events with detailed semantic description, including the involved player(s), event type(s), and result consequence.

Besides video content personalization, viewers individuality also reflects in their diverse customization modes and environments. Here, video customization mode regulates the concrete selection criteria of video segments. For viewers focusing on the particular player or events, segments that are semantically consistent to viewers' interest are better, while for the viewer interested in a game's global situation, segments that can best capture the main body of the match are more preferable. As for the customization environment, it denotes multiple constraints during the practical usage, for example, memory capacity, electrical quantity and transmission bandwidth, and so forth. All these physical conditions differ from person to person, and directly affect the final customization result.

Video personalization not only lies in customizing pointed content, but also embodies in system adaptation to the particular viewer. It is an important function for an intelligent system that can be made easier to use as the user continues to use it. User preference learning is an effective measure to tackle such problem. By analyzing the explicit or implicit feedbacks given by the user, an intelligent system can automatically infer the latent user preference and adaptively adjust its structure or/and parameters for future more convenient usage.

In this paper, we aim to propose a sports video personalization framework, through which users can enjoy refined video segments containing their favorite semantics from the lengthy sports match at anytime in anyplace. Both subjective content preference and objective environment constrains will be well balanced so that the optimal visual experience can be brought to the particular viewer. Particularly, we adopt basketball and football games as our initial sports genres because they are not only widely adopted study bed but also globally popular sports, which possess great values in both research and application. Moreover, since the proposed framework is generic, we believe that our approach can be easly extended to other sports domains.

The rest of the paper is organized as follows. Related work on sports video analysis, retrieval, summarization, and user preference learning are reviewed in Section 2. The problem formulation and proposed framework are described in Section 3. The technical details of sports video annotation, personalized video customization, and system adaptation are presented in Sections 4, 5, and 6, respectively. Experimental results are reported in Section 7. We conclude the paper with future work in Section 8.

2. Related Work

Extensive research efforts have been devoted to sports video analysis and application due to their wide viewership and enormous commercial potential. In this section, we give a brief review of related work on sports video annotation, retrieval, summarization, and user preference learning.

2.1. Sports Video Analysis and Annotation. Sports video analysis and annotation aim at the detection and recognition of semantic content. Most of the previous work is based on the audio [1–3], visual [4, 5], and textual [6, 7] features directly extracted from video content itself. The basic idea of such methods is to utilize heuristic rules [8] or machine learning algorithms [9, 10] to infer semantic events from various low-level [11] or mid-level [12] features. Since sports video is an integration of various information modalities, algorithms based on multimodal fusion are more likely to achieve robust and accurate event detection result than single-modality methods. For example, audiovisual features were successfully used to detect events in basketball [13], soccer [14], cricket [15], and tennis [16]; audiovisual-textual features were also collaborated in analyzing baseball [17], cricket [18], and golf [19] matches.

Nevertheless, due to the semantic gap between low-level features and high-level semantics, the content-based methods, no matter using single- or multimodality, can only annotate certain salient events with simple concepts, which cannot meet viewers' personalized appetites for specific players and events. In order to obtain more abundant high-level semantics, external textual information is introduced to facilitate video annotation and has achieved encouraging results. Babaguchi et al. [20] proposed a multimodal strategy using closed caption for event detection and video indexing. Xu and Chua [21] raised an integrative approach to align text events with match phase information to detect multiple events in soccer video. Xu et al. [22] used web broadcasting text from sports websites to detect event semantics and achieved inspiring results.

2.2. Sports Video Retrieval and Summarization. As for the direct applications of sports video analysis, retrieval and summarization represent two typical customization modes. Video retrieval can be regarded as a point query, where users focus on the particular person or event, while video summarization can be considered as a plane query, where users are more concerned about the entire situation of the match.

For methods using only low-level features, slow-motion replay portions and various highlight segments are competent candidates for video summarization. In [8], Ekin et al. summarized soccer video by classifying shot types and detecting slow-motion replay. In [23, 24], the whole sports video was divided into a sequence of play/break segments and highlights were assigned to the play segments for complete sports video summarization. For methods using textual features, video customization can effectively incorporate user preference and operate on the semantic level. Fleischman and Roy [25] proposed an unsupervised sports video retrieval approach by pairing repeated temporal visual patterns with associated closed caption text. Babaguchi et al. [26] proposed a personalized video retrieval and summarization framework based on the rich semantic description obtained from closed caption recognition.

Considering various environment limitations in the practical usage, such as network traffic and device memory, resource-constraint video customization also received wide attention from both academy [27, 28] and industry [29].

2.3. System Adaptation. Adaptation is an important mechanism of enabling an information system to provide personalized service to the particular user. Through learning user preference, a personalization system can adaptively adjust its structure and/or parameters to provide more pointed service.

In previous work, relevance feedback is a representative approach of explicit user preference learning. The main idea is using human-computer interaction to directly guide system adaptation so that it can provide focused service to the particular user. Zhang et al. [30] designed a relevance feedback strategy to retrieve suitable sports video clips to meet user request. By computing both the semantic and visual consistency of selected video segments, users'

personalized preference can be properly quantified and satisfied. Amir et al. [31] proposed a mutual relevance feedback method for multimodal query formulation in video retrieval. Based on the relevant shots marked by the user, the system can automatically identify useful search terms to refine the retrieval result.

Due to the time-consuming interactions in explicit feedback, implicit feedback is utilized to make the system adaptive to the user with less interruption. User profile analysis is a typical method of implicit user preference learning. Syeda-Mahmood and Ponceleon [32] adopted a hidden Markov model to predict users' internal states from their history browsing behaviors, and then generated a specified video preview for the particular viewer. Zimmerman et al. [33] used two kinds of implicit recommenders, the Bayesian classifier and the C4.5 decision tree, to learn users' preference from their viewing histories and fused multiple recommendations with a neural network to generate user favorite TV shows.

3. Problem Formulation and Framework

Video annotation serves as the foundation of personalized video customization. It takes the responsibility of connecting low-level audiovisual segments with high-level semantics. Compared with content-based annotation, approaches utilizing external textual information can provide more detailed semantic description of video content. Currently, two types of external textual sources, closed caption and web-casting text, are used for semantic video analysis. For the closed caption, although it holds the inherent advantage of video-text synchronization, it faces the challenge of accurate information extraction from irregular and variable spoken language, while for the web-casting text, the reverse applies, which means it has well-defined syntax structure but lacks of direct video-text correspondence. Most related work [22, 30] adopted the timestamp as a key link to connect these two media. However, these methods are usually confined to the lower timestamp recognition accuracy due to the noisy broadcast video. Moreover, the availability and concrete styles of timestamp are always decided by the program producer, which further limit the expansibility of such timestamp-based approaches.

With the detailed semantic annotation, two research challenges need to be addressed for personalized video customization: first, the incorporation of high-level semantics in video content selection, second, the balance between user preference and environment constraints. If we consider the customization problem from the optimization point of view, the first challenge defines an objective function to evaluate the semantic importance of video segments, while the second challenge models a constraint optimization problem on the basis of the above objective function by adding environment constrains. Most previous work focused on only one aspect of the above two problems, either semantic content selection [25, 26] or resource-constrained application [28, 29]; all of which lacked an integrated consideration of the above two challenges.

To practice the concept of "human-centered multimedia" [34], it is the responsibility of the information system rather than the user to adapt itself to provide more convenient service. On the side of the user, an ideal personalization system should learn his/her preference as accurately and unconsciously as possible. However, these two requirements are usually incompatible in the practical implementation. Specifically, explicit feedback [30, 31] gives accurate user preference but requires additional interaction, while implicit inference [32, 33] needs less operations but may result in incomplete learning.

In this paper, we will address the challenges existing in the previous approaches for semantic annotation, personalized customization, and system adaptation. Solutions toward the above problems in these three fields constitute an integrated system to provide personalized sports video customization. Compared with previous work in the related fields, the main contributions of our approach are summarized as follows.

(1) We propose an integrated framework to provide personalized sports video customization. With a comprehensive strategy on semantic annotation, personalized customization, and preference learning, users can conveniently customize their interested video segments concerning specific players or events.

(2) We propose a novel sports video annotation approach, where video content and web-casting text are aligned by their semantic correspondence along the temporal sequences. Since semantics is an intrinsic existence in multimedia, it is more generic and robust for cross-media analysis.

(3) We propose a user-participant multiconstraint 0/1 Knapsack model for personalized video customization, where user content preference and environment limitations can be well balanced and satisfied.

(4) We propose a social network based system adaptation algorithm to propagate local user interaction information along the video semantics network, from which complete user preference can be implicitly inferred and learned.

Figure 1 illustrates the framework of our proposed approach. First, a hierarchical semantic-matching method is employed to generate detailed video annotation in an off-line manner. Then, a user-participant content customization algorithm is proposed to provide real-time video content customization under resource-constraint environment. Meanwhile, to facilitate the above customization process, a concept network is built to capture the latent user preference for effective system adaptation.

4. Sports Video Annotation

Sports video annotation is responsible for the generation of detailed semantic description of video content. Recent work [22, 35] mainly focused on the timestamp-based video-text alignment, where timestamp's availability and recognition are two main bottlenecks restricting the application of

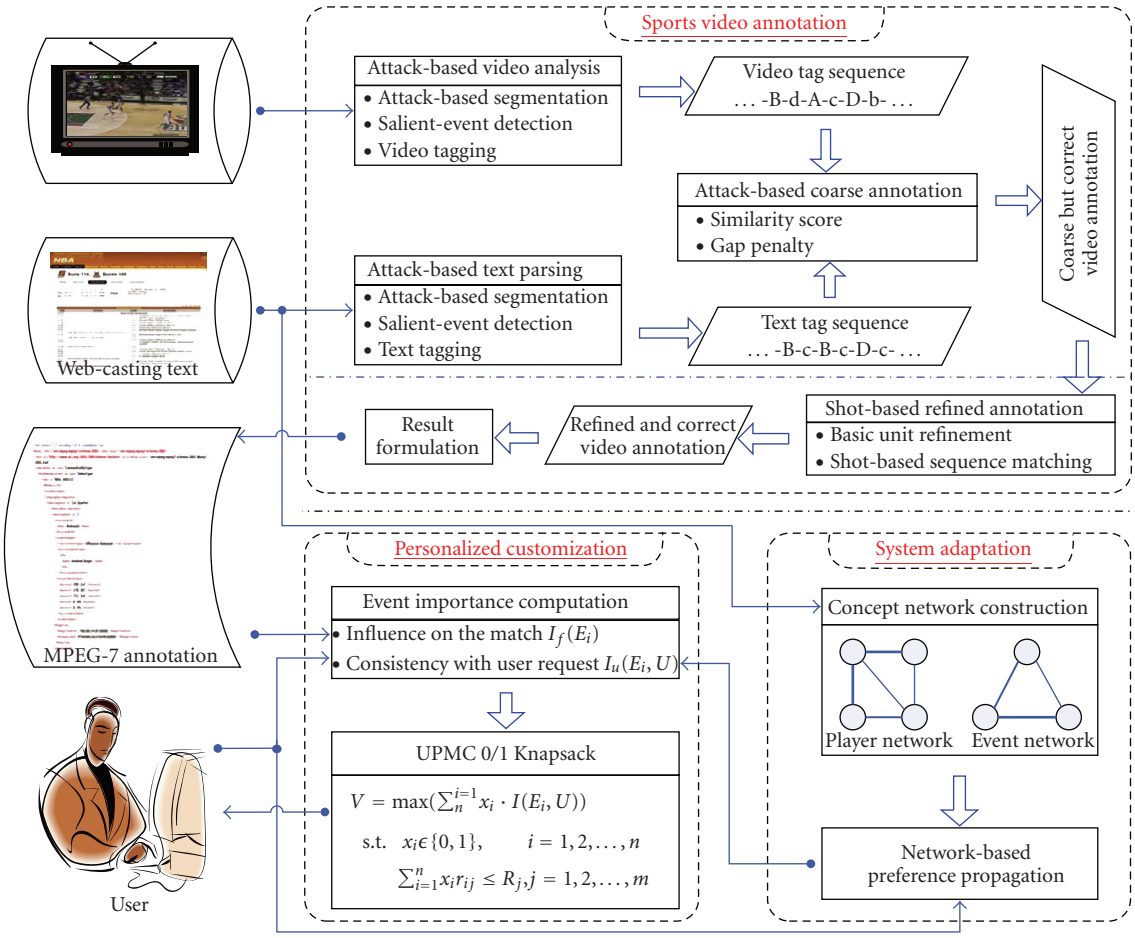


Figure 1: System framework of personalized sports video customization.

such method. To conquer these drawbacks, we propose a semantics-matching approach [36] to align the sports video and web-casting text based on their semantic correspondence. Since semantics is ubiquitous in various media and not susceptible to the postediting and transmitting, the proposed method is believed to be more generic to the cross-media analysis.

4.1. Attack-Based Coarse Alignment. To perform semantic matching between video and text, the first thing is to find a common abstract unit that can be reliably extracted from both media. Once such a unit is identified, the semantic bridge connecting the paired media can be effectively built. According to the sports video features, attack-based analysis method [37] provides a suited choice. Since the notion of attack exists in most opponent sports and has very clear semantics, a change in the attack side is a natural segmentation criterion for both game video and text.

4.1.1. Video Tagging. Video tagging module aims to generate a semantic tag sequence where each tag corresponds to an attack segment in the match. Here, attack is defined as a complete attempt of a team (player) in an opponent sport to score a point. It refers to a macroscopic process rather than a specific event; hence it includes not only obvious offensive events like shot or goal in an attack attempt, but also contains other nonoffensive events like foul or return pass during that process.

According to the above definition, consistent moving segment from one side of the court to the other usually corresponds to a complete attack unit. However, due to the fierce competition in the sports match, segmentation result is likely to be affected by a mass of blurring motion. To overcome this difficulty, we first smooth the horizontal camera motion [38] by the field zone information [21] so that blurring motion clips without obvious position change can be filtered out. Then, attack-based video segmentation is obtained with a sequence of boundaries at the start point of each remaining motion segment. A realistic example of the above process is illustrated in Figure 2, where blue solid line represents field zone information (1.5 denotes left field, −1.5 right field and 0 mid field) and green dash-dot line represents horizontal camera motion (1 denotes leftward moving, −1 rightward moving).

After attack-based video segmentation, a group of mid-level binary features including shot type transition (only long and nonlong shots are considered in our method) (*ST*) [39],

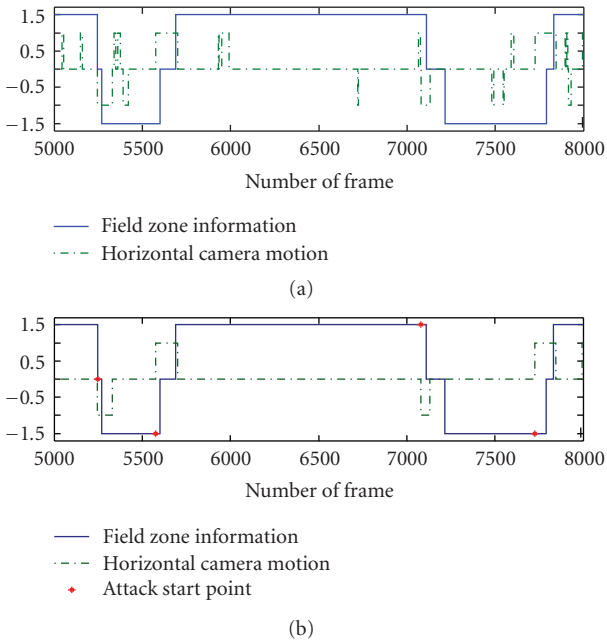


Figure 2: Attack-based sports video segmentation. (a) Initial field zone and horizontal camera motion information. (b) Smoothed horizontal camera motion and three attack segments (#1: 5210–5613; #2: 5613–7120; #3: 7120–7748).

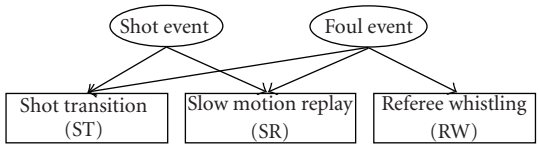


Figure 3: Naive Bayesian network for shot and foul events detection.

slow-motion replay (SR) [35], and referee whistling (RW) [13] are extracted from each attack segment and a heuristic Bayesian network (see Figure 3) is trained to detect shot and foul events as follows:

$$S^* = \underset{S}{\operatorname{argmax}} P(S \mid ST, SR, RW),$$
$$F^* = \underset{F}{\operatorname{argmax}} P(F \mid ST, SR, RW), \quad (1)$$

where S and F are binary variables corresponding to the existing states of shot and foul events in an attack segment and S^* and F^* are their inferred states. Finally, the detected attack direction and semantic events are further encoded by the combination of their binary status (see Table 1), and hence the whole video tagging process can be represented in a concise form as follows:

$$X^* = X(D, S^*, F^*) = \underset{X(D,S,F)}{\operatorname{argmax}} P(S, F \mid ST, SR, RW), \quad (2)$$

where X^* represents the final video tag encoded from the combination of attack direction (D) and the inferred semantic events (S^* and F^*).

4.1.2. Text Tagging. The utilization of textual information significantly facilitates video content analysis. Compared with caption text overlaid on the image [6] or encoded in the video [20], web-casting text [22] has following obvious advantages. (1) It is available in many famous sports websites such as ESPN (http://sports.espn.go.com/) and BBC (http://news.bbc.co.uk/sport2/hi/football/teams/) and can be timely accessed during or after the game. (2) It is organized in a well-defined structure that can be easily parsed. (3) It contains rich match description that are difficult to be obtained solely from content analysis (see Figure 4). Therefore, we utilize web-casting text in our method to facilitate semantic video annotation.

Similar to the content-based video tagging process, web-casting text analysis also aims to generate a semantic tag sequence where each tag corresponds to an complete attack attempt in the game text. Since web-casting text is tagged by sports professionals and provided by famous websites, its use of words and syntax structure are standard and fixed. Therefore, semantic events can be reliably detected by keyword-based searching. Table 2 lists the selected events and their related keywords used in our method; all of which are typically interesting events in the match and can be flexibly expanded according to users' preference.

With text event detection, attack-based text segmentation can be formulated as clustering adjacent text events into individual groups so that each group corresponds to a complete attack attempt in the match. By convention of composing the web-casting text, adjacent records with consistent attack directions always belong to the same attack process. Hence, we can cluster event records by analyzing the current attack side (CAS) in each text event. In opponent sports, the CAS is an important binary feature indicating which team is in the offense state when current text event is happening. Both one side's offense event and the other side's defense event correspond to the same CAS value. It can be reliably extracted from a text record by analyzing the player membership and event attack attribute (listed in Table 2). For example, given a text event saying "Michael Redd makes layup", we can know that it represents an offense event from the keyword "layup" and the CAS is the Bucks, the team that Michael Redd belongs to. Once all CAS features are extracted from text events, segment boundaries can be easily identified by sequentially comparing the adjacent two CAS features. To better understand the above process, the detailed algorithm flow and an example for CAS detection are given in Figure 5.

When using the above text segmentation method, one implementation detail needs to be addressed. As can be seen from Table 2, except rebound event, all other events having double attack attributes are related to the foul event. Since these events can happen in both offense and defense sides of an attack, their attack attributes are always difficult to be identified from a single text record. To solve this problem, we utilize their previous event's CAS feature and current event's related team to codetermine current event's attack attribute. Specifically, if current event is performed by a player belonging to the offense team in the previous event, its attack attribute is the offense, otherwise the defense. Such a rule is based on the assumption that the attack

Table 1: Code book used for video tagging.

Semantics	Attack direction		Semantic events	
Tag	Rightward	Leftward	Shot event	Foul event
"a"	true	false	false	false
"b"	true	false	true	false
"c"	true	false	false	true
"d"	true	false	true	true
"A"	false	true	false	false
"B"	false	true	true	false
"C"	false	true	false	true
"D"	false	true	true	true

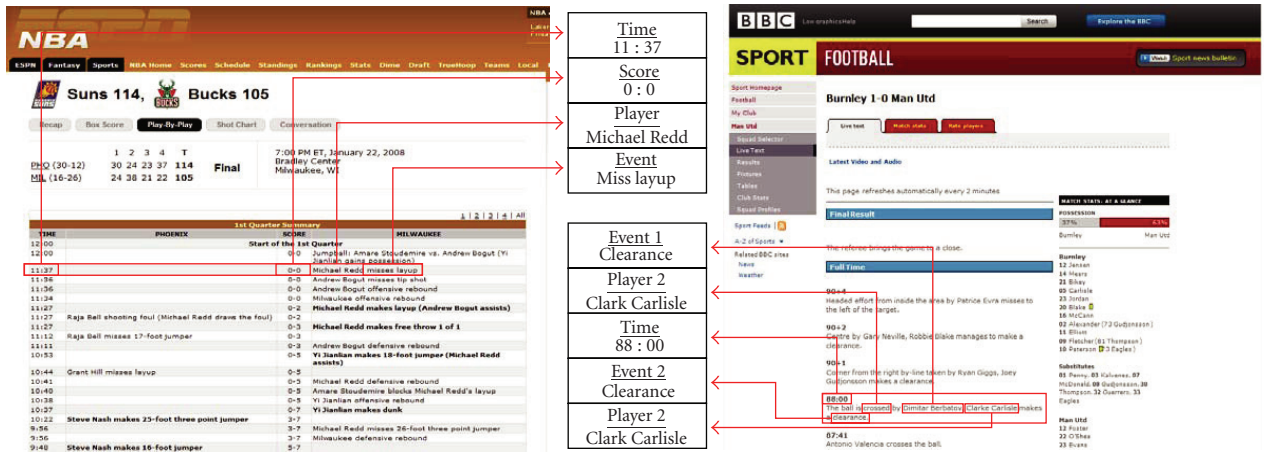


Figure 4: Web-casting text for (a) basketball and (b) football matches.

side is less likely to change in two adjacent events when the latter is a foul event (If not so, there must be an event in between causing the attack side change, which is contradictory with the assumption that two events are adjacent.). Our experiments also prove the rationality of such assumption.

With detected attack side in the match, the related attack direction can be easily inferred by comparing the attack segment ratios between two teams in the text and two directions in the video. Then, similar encoding process can be conducted to generate text semantic tag sequence.

Although video and text tagging share similar processing steps and output, we must stress that these two sequences are intrinsically different. For the former, each tag is a random variable and the finally derived video sequence is composed of the most likely semantic tags given an observation of mid-level features extracted from video attack segments. In contrast, the text sequence is a constant sequence with each tag being identified in a determinate way. Such a difference provides convenience to the tag similarity measurement in the following subsection.

4.1.3. Attack-Based Sequence Matching. The output of video analysis is a tag sequence with accurate attack boundaries (in terms of video shot) but inaccurate semantic tags (due to the semantic gap), while the output of text parsing is another tag sequence with accurate semantic tag (from keywords matching) but without video boundary information. Therefore, an intuitive way to annotate sports video on the level of attack is to align the accurate text tag sequence with its related video counterpart. Considering the semantic tag sequence, we propose a semantic-based Needleman-Wunsch algorithm [36] to match the probability video sequence with the constant text sequences. Compared with other algorithms, the proposed algorithm searches for the optimal alignment based on the global semantic correspondence, hence is more robust to local errors in the inaccurate video tag sequence.

The standard Needleman-Wunsch algorithm [40] is intrinsically a dynamic programming algorithm that initially aims to find the best protein or nucleotide sequence matching in bioinformatics. It is implemented in a multistage decision process, where the forward computation calculates the optimal local matching scores under various matching modes and the backward tracking identifies the global optimal sequence alignment. Consider a matching problem between two sequences, $\mathbf{v}$ and $\mathbf{t}$, which has m and n elements, respectively. In the forward computation, a score matrix $\mathtt{M}$ with m rows and n columns is first allocated, where each row corresponds to an element in $\mathbf{v}$ and each column to

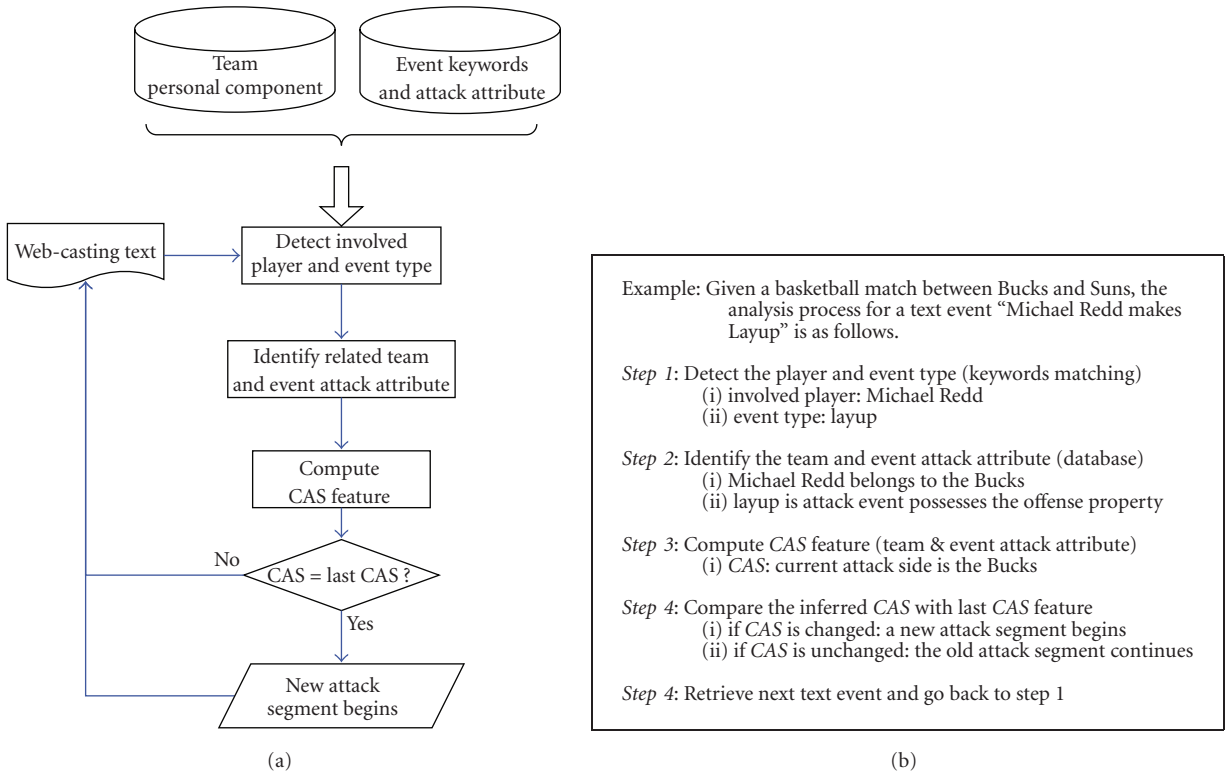


Figure 5: Attack-based text segmentation (a) algorithm flow and (b) an example.

Table 2: Event keywords and attack attributes.

Domains	Events	Related keywords	Attack attributes
Basketball	shot	makes layup, hook, jumper, free throw	offense
	miss	miss	offense
	block	blocks	defense
	stolen	steal, lose ball, bad pass	defense
	foul	foul	offense/defense
	rebound	offensive rebound, defensive rebound	offense/defense
Football	shot	shot, goal, head, scored	offense
	free kick	free kick	offense
	corner	corner	offense
	foul	foul	offense/defense
	card	red card, yellow card	offense/defense
	offside	offside	offside

an element in $\mathbf{t}$. The (i, j)th entry of the matrix $\mathtt{M}$ records a local optimal alignment score between subsequence $\mathbf{v}_1 \sim \mathbf{v}_i$ and $\mathbf{t}_1 \sim \mathbf{t}_j$. The forward computation is implemented in an iterative manner, where the value of $\mathtt{M}_{i,j}$ is decided by one of its three adjacent predecessors as follows:

$$\mathtt{M}_{i,j} = \max\left\{\mathtt{M}_{i,j-1} + \mathrm{Pg},\ \mathtt{M}_{i-1,j} + \mathrm{Pg},\ s_{i,j} + \mathtt{M}_{i-1,j-1}\right\}, \quad (3)$$

where $s_{i,j}$ is the similarity score between the ith video tag ($\mathbf{v}_i$) and the jth text tag ($\mathbf{t}_j$), Pg denotes the gap penalty given to an empty matching. In (3), the first two items correspond to the rightward (gap-element) and downward (element-gap) matching, where element in one sequence cannot find its counterpart in the other sequence; thus a gap matching generated. The third item in (3) denotes a diagonal (element-element) matching where the similarity between element $\mathbf{v}_i$ and $\mathbf{t}_j$ is computed. By comparing the final alignment scores generated from three directions, the highest score is assigned to $\mathtt{M}_{i,j}$ and the related direction is stored in the corresponding position of an equal-sized $(m \times n)$ backtracking matrix $\mathtt{B}$. As the whole score matrix is allocated, the related backtracking matrix is also filled. Starting from the bottom right of the backtracking matrix, the global optimal alignment path

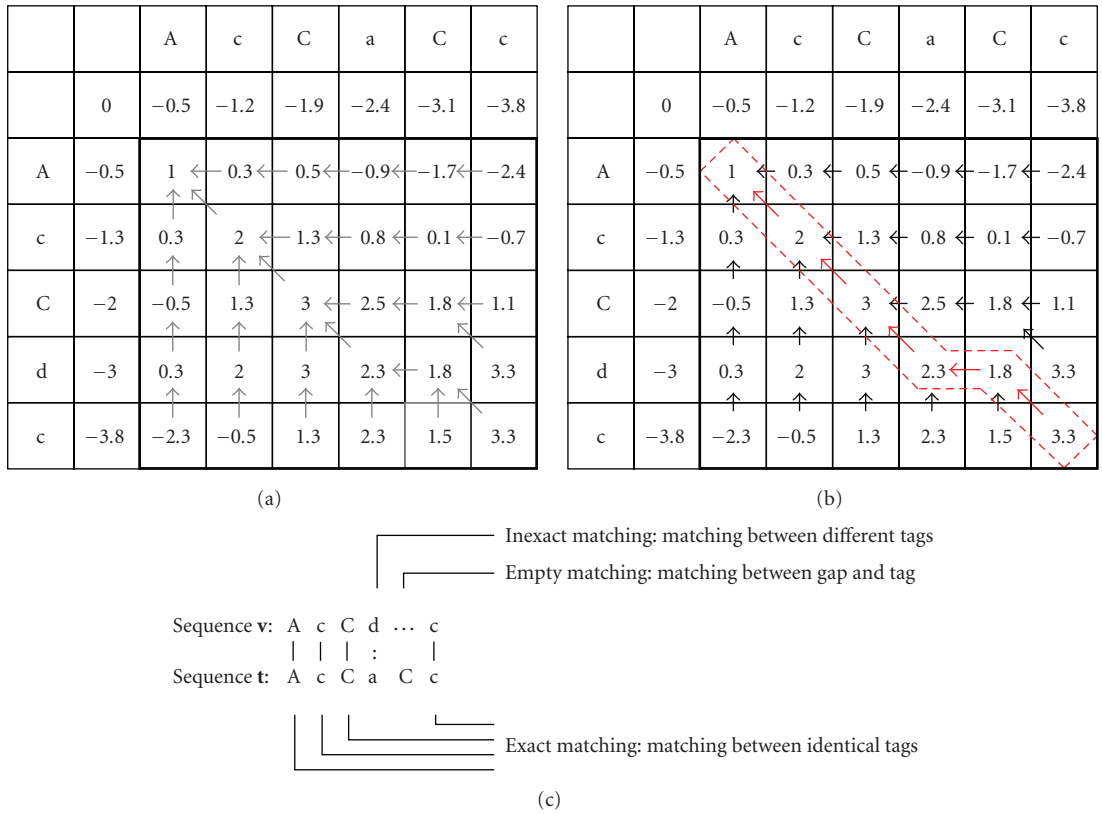


Figure 6: (a) Forward filling of score matrix and backtracking matrix, (b) backward tracing along best local alignment directories, and (c) sequence alignment result.

can be finally obtained through tracing back along the stored local matching directory. A graphical demonstration of above algorithm flow is illustrated in Figure 6, where numbers and direction in the thick dark black box in Figure 6(a) constitute the score and backtracking matrixes, respectively, and red arrows in Figure 6(b) indicate the final backtracking directory corresponding to the optimal sequences alignment. Three different matching types are obtained in Figure 6(c), where the mark "|" denotes the exact matching between identical elements and mark ":" denotes the inexact matching between different elements. As for the empty matching, no mark is presented.

From above discussion, the forward computation of score matrix serves as the core part of the Needleman-Wunsch algorithm. It affects the final alignment result by deciding the local alignment directions. In the field of bioinformatics, the sequence elements usually correspond to certain chemical substances, and thus the matching scores are given based on their chemical structures and properties. However, in the particular application of sports video and game text alignment, since each sequence element is a character tag representing the combination of high-level features and events, the alignment scores should be marked according to their semantic similarities.

Because the video tag is intrinsically a discrete random variable with its value ($\mathbf{v}_i$) corresponding to the largest probability in the distribution generated by Bayesian network, the semantic similarity between video tag $\mathbf{v}_i$ and text tag $\mathbf{t}_j$ can be indirectly measured by the conditional probability difference when the video tag variable taking the values of $\mathbf{v}_i$ and $\mathbf{t}_j$, given an observation of the mid-level features. In our approach, we formulate the similarity measure between video and text tags as follows:

$$s_{i,j} = \begin{cases} -\infty & \text{inconsistent directions,} \\ \dfrac{P\left(X = \mathbf{t}_j \mid ST, SR, RW\right)}{P(X = \mathbf{v}_i \mid ST, SR, RW)} & \\ = \dfrac{P\left(X = \mathbf{t}_j \mid ST, SR, RW\right)}{\max P(X \mid ST, SR, RW)} & \text{consistent directions} \end{cases} \tag{4}$$

where X is the video tag variable and ST, SR, RW represent audiovisual features described in Section 4.1.1. Since the attack direction can be reliably identified from both video and text, tags corresponding to inconsistent attack directions are not allowed to be aligned in the approach. For the direction consistent condition, the more proximal the inference probabilities between tags $\mathbf{v}_i$ and $\mathbf{t}_j$ are, the more likely $\mathbf{t}_j$ can be used to replace $\mathbf{v}_i$ to annotate the related video segment, in other words, the more likely video segment tagged as $\mathbf{v}_i$ can be aligned with text group tagged as $\mathbf{t}_j$. Moreover, since $\mathbf{v}_i$ corresponds to the maximum condition probability in

the video tagging process, the tags similarity $s_{i,j}$ can never be larger than 1, which can only be reached in the exact matching.

As for the gap penalty, it is defined as a function of the semantic events contained in the attack segment:

$$\mathrm{Pg} = [-0.5 - 0.25 \times (S^* + F^*)] \times \theta, \tag{5}$$

where S^* and F^* represent the most probable existing state of shot and foul events generated by (1) and θ is the affine gap cost defined as 1 for the first gap and 0.95 for others in our following experiments. According to above equation, the more events contained in an attack, the less likely it cannot find a corresponding text tag, hence the severer punishment will be given to its empty alignment, and vise versa. Table 3 describes the proposed semantics-based Needleman-Wunsch algorithm for the tag sequence alignment between broadcast sports video and web-casting text.

If we consider a timestamp as a tag, previous timestamp-based methods can be regarded as a special case of our approach. However, two important differences exist. First, tags in our method represent high-level semantics rather than low-level visual features, which is an intrinsic and generic link across multimedia. Second, global structure rather than individual equivalence is utilized to align video and text sequences, which greatly improve the method's robustness against local errors. Therefore, the proposed semantics-matching approach is considered to be more effective for the generic cross-media analysis.

4.2. Shot-Based Refined Alignment. The output of the attack-based alignment is a coarse but accurate annotation result, where the basic unit corresponds to an attack segment rather than a specific event. To obtain a more elaborate event detection result, which locating the semantic event in a specific video shot, we repeat previous sequence matching algorithm on the scale of each attack to generate shot-level video annotation. With the variance of matching granularity, the basic sequence units change from the attack segment into a shot in the video and an event record in the text. Since long shots are always adopted to depict the global situation when the match is in play, they are the only shot type used in refined alignment.

Although the shot-based refined alignment lacks direct semantic correspondence between sports video and web text, it indeed generates more elaborate annotation on the level of shot. Moreover, our experiments also show that such semantic weakness during the refinement process based on the coarse alignment result is not significant and hence totally acceptable.

Final video annotation result is stored in the standard MPEG-7 XML format for efficient retrieval and management. As shown in Figure 7, the XML file is organized as a hierarchical structure in accordance with its related sports match, where the whole match includes several quarters and each quarter contains a group of events. For each event in the match, the XML file records the detailed information including its involved player(s), event type and moment, current score and the corresponding video segment.

5. Personalized Video Customization

Personalized video customization aims to tailor proper video content to the particular user. Different from generic highlight presentation work [23, 24] where interesting video clips are fully decided by video content analysis, the proposed customization scheme is featured in the cooperation of video semantics and user preference in video content selection process. Moreover, considering the conflict between the mass video content and limited user environment, a balanced customization strategy is also addressed to maximize audience's viewing enjoyment under various context constraints.

5.1. Event Importance Computation. To evaluate the quality of selected video segments, event importance computation is indispensable. Both event influence on the match and its relevance to user request are taken into account. For influence computation, event rank and occurrence time are two main factors [41]. For relevance measurement, the semantic consistency on involved players and event types are considered.

(1) Event Rank. The rank of event is directly determined by its influence to the game state. Considering the difference between basketball and football match, two representative rank criteria are adopted to evaluate the event importance in these two sports types. Since shot event is common in basketball matches, the change of game leader and score gap are used as the indicator for event importance evaluation. While for the football match, influential events are confined in limited types and thus only shot and card events are ranked. With the rank list given in Table 4, the rank-based event importance $I_{\mathrm{ra}}(E_i)$ $(0 \le I_{\mathrm{ra}} \le 1)$ is defined as follows:

$$I_{\mathrm{ra}}(E_i) = 1 - \frac{R_i - 1}{3} \cdot \alpha, \tag{6}$$

where R_i $(0 \le R_i \le 4)$ represents the rank level of event E_i and α $(0 \le \alpha \le 1)$ denotes the adjustable parameter to control the effects of rank difference on event importance computation.

(2) Event Occurrence Time. In the sports match, events occurring at the end of the match are usually critical to both sides because little time is left to change the final result. In our approach, the event occurrence time based importance $I_t(E_i)$ $(0 \le I_t \le 1)$ is defined as follows:

$$I_t(E_i) = 1 - \frac{N - i}{N} \cdot \beta, \tag{7}$$

where N is the total event number, i is the index of E_i, and β $(0 \le \beta \le 1)$ is the adjustable parameter to control the effects of event occurrence time on event importance computation.

(3) Event Relevance. Different from the type rank and occurrence time that are decided by event itself, the event relevance reflects the semantic consistency between event content and user's preference. This item plays an important role in our

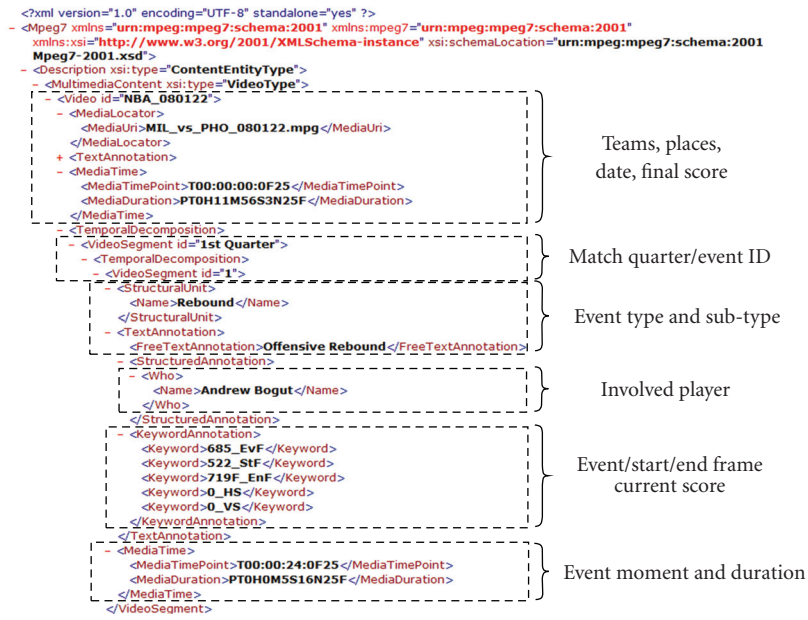


Figure 7: An example of MPEG-7 sports video description XML file.

Table 3: Sequence matching algorithm for video-text alignment.

Output	Alignment sequence $\mathbf{a} : \mathbf{a}_1 \sim \mathbf{a}_{\max(m,n)}$
Input	Video tag sequence $\mathbf{v} : \mathbf{v}_1 \sim \mathbf{v}_m$ and text tag sequence $\mathbf{t} : \mathbf{t}_1 \sim \mathbf{t}_n$
Step 1	Forward computation
Step 1.1	Allocate a score matrix $\mathtt{M}$ and a direction matrix $\mathtt{B}$.
	Both matrixes are with m rows and n columns.
Step 1.2	Fill all entries in $\mathtt{M}$ and $\mathtt{B}$ from the up left to bottom right corner.
	(i) For each $\mathtt{M}_{i,j}$, its value is calculated by (3), where $s_{i,j}$ and Pg are computed by (4) and (5).
	(ii) For each $\mathtt{B}_{i,j}$, its value is the optimal matching direction indicated by $\mathtt{M}_{i,j}$ in (3).
Step 2	Backward tracing
Step 2.1	Allocate an alignment sequence a with $\max(m, n)$ elements.
Step 2.2	Fill the entries of $\mathbf{a}$ in a reverse direction by tracing back along $\mathtt{B}$.
	(i) For each $\mathbf{a}_k$, its value is equal to $\mathtt{B}_{i,j}$, where (i, j) corresponds to the coordinate of the kth backtracking step in $\mathtt{B}$.
Step 3	Result explanation
Step 3.1	Three possible values (directions) may appear in $\mathbf{a}_k$, which are
	(i) diagonal, corresponding to the "$\mathbf{v}_i - \mathbf{t}_j$" alignment,
	(ii) downward, corresponding to the "empty-$\mathbf{t}_j$" alignment,
	(iii) rightward, corresponding to the "$\mathbf{v}_i$-empty" alignment.

Table 4: Event importance rank list.

Rank Level	Event description	
	Basketball	Football
1	Score events and their inductive foul events that can change the game leader	Score (goal events)
2	Score events and their inductive foul events that retain the game leader but change the scoring gap	Offense events and red card event
3	Offense events that failed to score a goal and common foul events	Yellow card event
4	All other events that are not in the rank 1 to 3	All other events

Remark: in above description, score event refers to attack event resulting in a score, while offense event refers to attack event without resulting in a score.

personalized customization because it effectively introduces the user's opinion in event importance computation. By increasing the importance score of semantically related events, the system can finally tailor the personalized video content to the particular user. With semantic video annotation, the event relevance based importance $I_{\text{re}}(E_i)$ $(0 \le I_{\text{re}} \le 1)$ can be defined as:

$$I_{\text{re}}(E_i) = \gamma^{\text{Dist}_{\text{player}}(E_i,U)} \cdot (1-\gamma)^{\text{Dist}_{\text{event}}(E_i,U)}, \tag{8}$$

where function $\text{Dist}_{\text{player}}$ and $\text{Dist}_{\text{event}}$ measure the semantic consistence between event E_i and user request U on the subjects of involved players and event types, and adjustable parameter γ $(0 \le \gamma \le 1)$ denotes the user preference between the above two subjects.

(4) Preference Learning. With user preference learning, system can provide more appropriate video content to the particular users as they continue to use it. This function is realized by adaptively adjusting the concept weight in accordance with user's input. The calculation of the preference learning-based event importance $I_p(E_i)$ $(0 \le I_p \le 1)$ will be discussed in Section 6.

Based on the above analysis, the event importance can be calculated as:

$$\begin{aligned} I(E_i,U) &= \lambda \cdot I_f(E_i) + (1-\lambda) \cdot I_u(E_i,U) \\ &= \lambda \cdot I_{\text{ra}}(E_i) \cdot I_t(E_i) + (1-\lambda) \cdot I_{\text{re}}(E_i,U) \cdot I_p(E_i), \end{aligned} \tag{9}$$

where λ $(0 \le \lambda \le 1)$ is the fusion parameter distributing the weights on event influence on the match $I_f(E_i)$ and its semantic consistency to the user request $I_u(E_i)$.

5.2. User-Participant MultiConstraint 0/1 Knapsack Model. With the above event importance evaluation scheme, user preference can be effectively incorporated into the video content selection process. However, compared with mass suited video data, user's available viewing conditions, for example, device memory and watching time, are usually not limitless. Hence, how to provide optimal video content under resource-constraint environment is of great practical importance and has been studied in [28, 42]. Merialdo et al. [42] raised a 0/1 Knapsack Problem to model the viewing-time-limited TV program personalization. However, due to the lack of semantic analysis of video content, only category interest rather than content preference was considered in their video personalization system. Wei et al. [28] proposed a Multichoice Multidimension Knapsack strategy to maximize the gross information under multiple environment constraints. Since their method requires to include every video segments (on various abstraction levels) to the final summarization, it is not appropriate for the target-specific video customization, where only video segments containing particular semantics are needed. Motivated by the optimization model used in previous work, we formulated our personalized video customization as a user-participant multiconstraint (UPMC) 0/1 Knapsack problem:

$$\begin{aligned} V &= \max\left(\sum_{i=1}^{n} x_i \cdot I(E_i,U)\right), \\ &\text{s.t.} \quad x_i \in \{0,1\}, \quad i = 1,2,\ldots,n \\ &\sum_{i=1}^{n} x_i r_{ij} \le R_j, \quad j = 1,2,\ldots,m, \end{aligned} \tag{10}$$

where $I(E_i,U)$ represents the integrative importance value of event E_i under specified user request U, r_{ij} be the jth resource consumption of the ith event, R_j be the client-side resource bound of the jth resource, x_i denotes the existence of the ith event in the selected optimal set, and n and m represent the number of events and resource types.

As can be seen from (9) and (10), the proposed video customization strategy adopts a constraint optimization model to well balance the user content preference against multiple resource limitations, from which only video segments possessing higher importance values can be selected into the final customization result. With different fusion parameters, two typical video customization modes, retrieval and summarization, can be treated in an unified manner and seamlessly switched to each other. Specifically, In the case of λ approximating to 0, event importance is mainly decided by user preference, thus only semantically consistent video segments can be assigned higher importance scores and finally presented to the user. While in the case of λ approximating to 1, event importance is largely up to the match itself, hence the selected events can reflect the global situation of the match. Moreover, with unbiased configuration of the fusion parameter ($\lambda = 0.5$), a new customization mode, the personalized video summarization can be realized. In this situation, users can query a video abstract about the specified player or event.

6. System Adaptation

Since user preference is relatively stable in a period of time, customization system with adaptation function is expected to respond to the particular user with more appropriate results but less required interactions. To achieve this goal, complete user preference should be learned as implicitly as possible. However, the completeness usually conflicts with the implicitness and hence previous work can hardly achieve an optimal balance between these two indexes. To conquer this difficulty, we propose a social network based approach to identify latent user preference without additional interactions. By building the social network of video semantics, user preference towards specific concepts can be effectively propagated along the network edge so that their latent attitudes toward other unspecific concepts can be implicitly inferred.

6.1. Concept Social Network. We borrow the idea of social network analysis (SNA) [43] to depict the concept relationship in sports video. A social network is a social

structure made of individuals called "nodes", which are connected by one or more specific types of interdependency. With a graphical representation of individuals' relationship, SNA can utilize the related property and theory in the graph theory to discover hidden structures/properties that cannot be directly perceived or measured [44]. According to the semantic sports video annotation, two parallel social networks for the player and event entities are defined as follows:

$$G_p = \left\langle V_p, E_p, W_p \right\rangle, \qquad G_e = \langle V_e, E_e, W_e \rangle, \tag{11}$$

where $V_p = \{v_{p_1}, v_{p_2}, \ldots, v_{p_n}\}$ represents the set of players appearing in the match, $E_p = \{e_{p_{i,j}} \mid e_{p_{i,j}} = 0 \text{ or } 1\}$ is a binary matrix indicating the relationship existence between players i and j, and $W_p = \{w_{p_{i,j}}\}$ denotes the strength of the relationship between players i and j. Similar explanations can be obtained from event network G_e.

To build above weighted concept network, the quantization of concepts' relationship is critical. In the scenario of sports matches, the relation between concepts is developed when they appear in the similar match context. Here, the match context refers to the event type when we consider the player network and refers to the player when we consider the event network, in other words, the player and event are mutual match context of each other. Therefore, the more often two players appear in the same events, the closer relationship is built between them in the player network, and we can quantify this relationship as the number of cooccurrence in the same match context between two players. Similar conclusion can be also drawn from the event network.

With the obtained semantic sports video annotation, a match can be viewed as a bipartite graph in Figure 8(a), where the square nodes denote events, the circle nodes denote players and the edge between a pair of event and player nodes represents their cooccurrence in the same text event. For a match with m events and n players, the above bipartite graph can be represented as a matrix $\mathtt{A} = [a_{ij}]_{m\times n}$, where the entry a_{ij} represent the cooccurrence times of the ith event and jth player in the same text events. The jth column, $\mathbf{a}_j^\top = \{a_{1j}, a_{2j}, \ldots a_{mj}\}$, of matrix $\mathtt{A}$ represents the cooccurrence times between the jth player and other m events, and the ith row, $\bar{\mathbf{a}}_i = \{\bar{a}_{i1}, \bar{a}_{i2}, \ldots, \bar{a}_{in}\}$ of matrix $\mathtt{A}$ represents the cooccurrence times between the ith event and other n players. Based on the event-player bipartite graph, we can build the concept social networks as follows:

$$\begin{aligned}
&\mathtt{E} = \mathtt{A}\mathtt{A}^\top = \left[E_{ij}\right]_{m\times m},\\
&\quad \text{where } E_{ij} = \begin{cases} \sum_{k=1}^{n} \bar{a}_{ik}\bar{a}_{jk} = \bar{\mathbf{a}}_i\bar{\mathbf{a}}_j^\top, & \text{when } i \neq j,\\ 0, & \text{when } i = j, \end{cases}\\
&\mathtt{P} = \mathtt{A}^\top\mathtt{A} = \left[P_{ij}\right]_{n\times n},\\
&\quad \text{where } P_{ij} = \begin{cases} \sum_{k=1}^{m} a_{ki}a_{kj} = \mathbf{a}_i^\top\mathbf{a}_j, & \text{when } i \neq j,\\ 0, & \text{when } i = j, \end{cases}
\end{aligned} \tag{12}$$

where E and P correspond to the event and player networks, respectively.

In the example of Figure 8, the generated social network between events and players are illustrated in Figures 8(b) and 8(c). The thicker an edge is, the more similar of two concepts in their match context. Take the edge between e_1 and e_3 in Figure 8(b) as an example, it is the thickest one among all three edges, meaning these two events usually occur with the same players (p_3 and p_4) in the match. Hence, for users who like event e_1, they may be also interested in e_3 because they are potentially fond of p_3 and p_4. This analysis can be applied to the player network where its match context refers to the cooccurring events.

6.2. User Preference Learning. With the help of social network analysis, complete user preference can be effectively inferred from the finite customization process. For the convenience of the following discussion, we introduce a concept-weight pair, $\langle C_k, W_k \rangle$, to describe the user preference degree W_k of the kth concept (C_k) in the match. In addition, due to the symmetric processing and similar conclusion of the event and player concepts, we will not differentiate these two kinds of concepts unless it is necessary.

Initially, the weights of all concepts are set to 0. When the kth concept is used as a keyword in video customization, it is regarded as an obvious interesting concept to the user. Hence, the new weight W_k^{new} of the kth concept whose last weight is W_k^{old} is given by the following equation:

$$W_k^{\text{new}} = \phi + W_k^{\text{old}}, \tag{13}$$

where $\phi > 0$ is a constraint increment of concept preference.

As for other concepts that are not specified by the user, we further divide them into two classes, which are concepts that are connected with the kth concept in the social network and those are not. For the former, we distribute the weight increment ϕ to these latent concepts according to their link strength to the kth concept as follows:

$$W_i^{\text{new}} = \phi \cdot \frac{L_{ki}}{\sum_l L_{kl}} + W_i^{\text{old}}, \tag{14}$$

where L_{kl} represents the weight of the edge between the kth and lth concepts in the social network. For the rest concepts, which are neither specified by the user nor connected with the specified concept, their preference weights are decreased as follows:

$$W_j^{\text{new}} = \eta \cdot W_j^{\text{old}}, \tag{15}$$

where $1 > \eta > 0$ is a constant damping factor of concept preference.

With equations (13)–(15), user preference to all concepts in the match are identified. To avoid the weight divergence, a normalization process is adopted:

$$W_k = \frac{W_k^{\text{new}}}{\max\left\{W_k^{\text{new}}\right\}}, \tag{16}$$

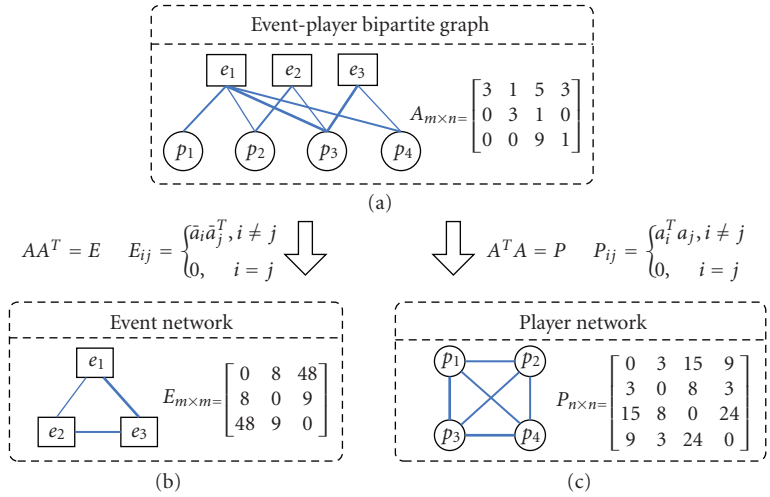


Figure 8: Graphical example of player and event network construction.

so that the normalized concept weight is not larger than 1. The corresponding preference learning based event importance $I_p(E_i)$ $(0 \le I_p \le 1)$ can be computed as:

$$I_p(E_i) = \frac{1}{2} \cdot \left(W_{\text{player}}(E_i) + W_{\text{event}}(E_i)\right), \tag{17}$$

where $W_{\text{player}}(E_i)$ and $W_{\text{event}}(E_i)$ correspond to the normalized preference weights of the player and event concepts in event E_i.

7. Experimental Result

In order to evaluate the proposed method, we conduct our experiment on more than 1000-minute broadcast sports video, including 2 NBA 2005-2006 and 4 NBA 2007-2008 basketball matches, 2 Euro-Cup 2004 and 4 World Cup 2006 football matches. The corresponding web-casting texts are obtained from the ESPN and BBC sports websites. In average, there are about 400 text events happening in one 100-minute basketball match and 50 text records in one 90-minute football match.

7.1. Sports Video Annotation. In this part, attack-based video and text segmentation, coarse and refined video annotation, and comparative experiment with timestamp-based method are reported to validate the effectiveness of the proposed semantic-matching algorithm.

To evaluate the obtained segmentation result, the "purity" index proposed in [45] is adopted in our experiment. Given a sequential data, a ground truth segmentation $\mathbf{S} = \{(s_1, \triangle t_1), \ldots, (s_g, \triangle t_g)\}$, and an automatic segmentation $\mathbf{S}^* = \{(s_1^*, \triangle t_1^*), \ldots, (s_g^*, \triangle t_g^*)\}$, the purity π is defined as:

$$\pi = \left(\sum_{i=1}^{g} \frac{\tau(s_i)}{T} \sum_{j=1}^{a} \frac{\tau^2\left(s_i, s_j^*\right)}{\tau^2(s_i)}\right) \cdot \left(\sum_{j=1}^{a} \frac{\tau\left(s_j^*\right)}{T} \sum_{i=1}^{g} \frac{\tau^2\left(s_i, s_j^*\right)}{\tau^2\left(s_j^*\right)}\right), \tag{18}$$

where $\tau(s_i, s_j^*)$ is the length of overlap between the scene segmentation s_i and s_j^*, $\tau(s_i)$ is the length of the scene s_i, and T is total length of all scenes. In each parenthesis, the first term is the fraction of recording for which a segment accounts, and the second term is a measure of how much a given segment is split into small fragments. The purity value ranges from 0 to 1. The larger the value is, the closer the result approaches the ground truth.

With the manually labeled ground truth segmentation, the attack-based video and text segmentation results are given in Figure 9. The consistent advantage of text segmentation result over that of sports video reflects the semantic intuitionism of the attack-based segmentation. The individual errors in the web text are mainly caused by the omission of some text records when the attack side is changed. As for the comparatively lower purity in video segmentation, large-scale back passings and inaccurate field zone information (especially in football matches) are primary reasons. However, despite some inaccuracies, the attack-based video and text segmentations in both basketball and football matches are generally feasible for the following semantic video annotation.

Leave-one-cross validation is adopted to evaluate the video annotation result. For each sports type, five matches are used to train the Bayesian network for salient event detection and the left one is used to test the annotation algorithm. With the attack-based segmentation, the coarse video-text alignment result is listed in Table 5, where the inference accuracy denotes the occupation of correctly detected tags in total video tags and alignment accuracy represents the occupation of all correctly matched tags in total video tags.

As shown in Table 5, the average inference accuracy is only about 77% for basketball matches and 86% for football matches, which reflects the negative effects of the semantic gap in content-based event detection. However, in contrast to the limited inference accuracy, the coarse alignment accuracy is still satisfactory (around 98% for basketball and 95% for

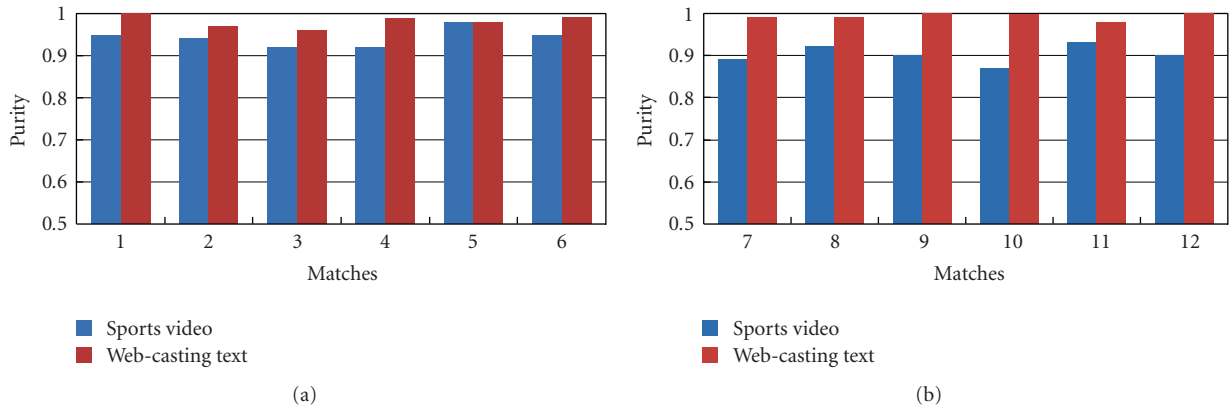


Figure 9: Attack-based sports video and web-casting text segmentation result in (a) basketball and (b) football matches. In both figures, the horizontal axes denote the match ID and the vertical axes represent the purity value of the segmentation result.

Table 5: Attack-based coarse alignment result.

No.	Sports type	Total video tags	Exact matching	Inexact matching	Inference accuracy	Alignment accuracy
1	basketball	192	143	44	74% (143/192)	97% (187/192)
2	basketball	196	152	42	78% (152/196)	99% (194/196)
3	basketball	188	133	49	71% (133/188)	97% (182/188)
4	basketball	176	144	26	75% (144/176)	97% (170/176)
5	basketball	187	150	33	80% (150/187)	98% (183/187)
6	basketball	185	140	38	76% (140/185)	96% (178/185)
7	football	55	45	8	82% (45/55)	96% (53/55)
8	football	50	43	6	86% (43/50)	98% (49/50)
9	football	53	44	5	83% (44/53)	93% (49/53)
10	football	53	47	3	88% (47/53)	95% (50/53)
11	football	51	45	4	88% (45/51)	96% (49/51)
12	football	45	39	2	87% (39/45)	91% (41/45)

football matches resp.). This result demonstrates the strong robustness of the proposed sequence matching algorithm and can be attributed to the semantics-based tags similarity measure and the global optimization strategy, where different tags can be effectively aligned with reference to the global matching status.

Attack-based video-text alignment generates a coarse but accurate video annotation, where the basic unit corresponds to an attack in the match. To obtain a more elaborate annotation, a refined alignment is carried out to locate each text event within a shot. Figure 10 gives the refined annotation result of 6 events in basketball and 7 events in football matches. Due to the lack of semantic relation, the shot-based alignment is not as accurate as the attack-based one. However, with the help of accurate coarse alignment, the following shot-based refinement can still annotate most events in an acceptable accuracy. As for the lower precision/recall rate of the shot event in basketball matches, it is mainly due to the irregular photography in free throw events where short shots rather than long shots were adopted by cameramen. Similar results also appear in the free kick and foul events in football matches, where the camera transition during those events is frequent and dynamic.

To further demonstrate the robustness of our proposed approach, a comparative experiment of our approach with timestamp-based method [35] is conducted on the evaluation data and their balanced F-measure results are shown in Figure 11. In the ideal condition, timestamp-based approach can achieve very high event detection accuracy if the timestamp can be correctly recognized. However, according to our experiments on both basketball and football matches, the above advantage is either not obvious (Figure 11(b)) or even in-existent (Figure 11(a)). This result can be explained from two aspects: First, timestamp cannot be always robustly located and recognized in the practical noisy broadcast video, which affects the event location precision of the timestamp-based method; Second, the basketball match has plenty of clock pauses, which make the timestamp-based method confused to locate the accurate event segment during that period. As for the lower detection accuracy of the proposed approach on football matches, it is mainly caused by the performance degradation when the sequence

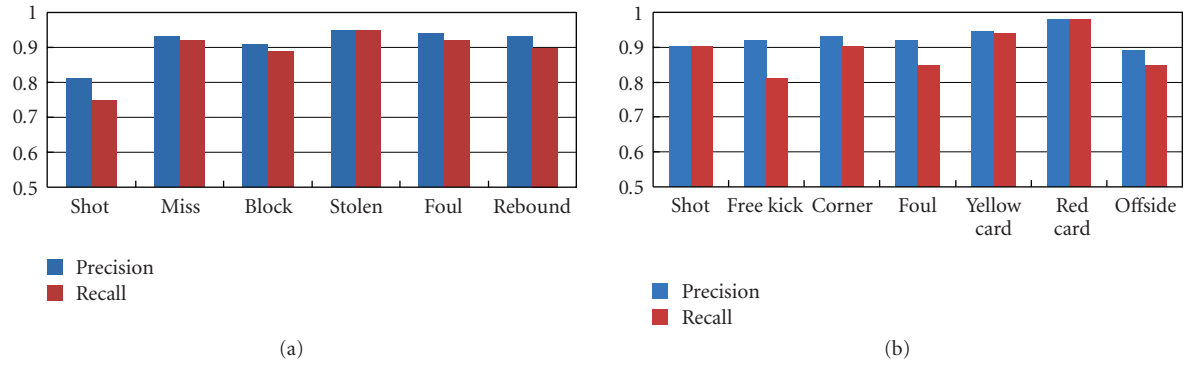


Figure 10: Refined event detection result on (a) basketball and (b) football matches.

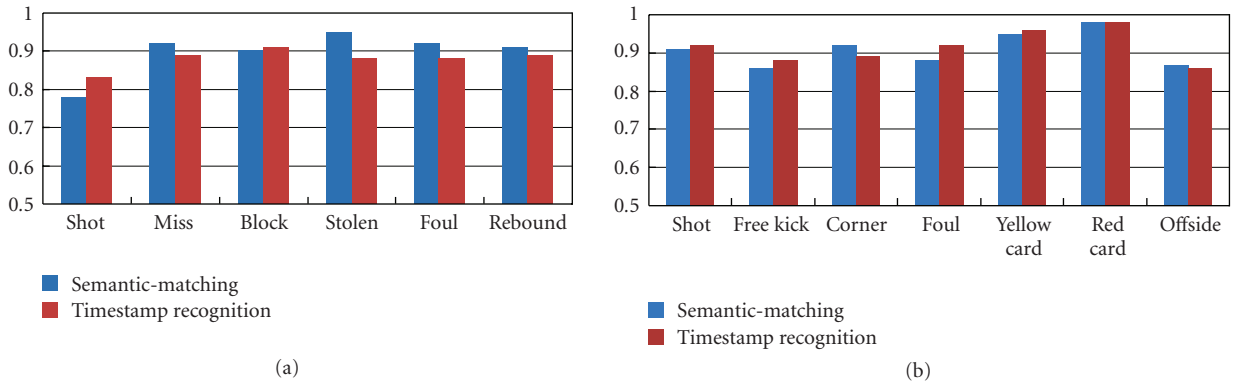
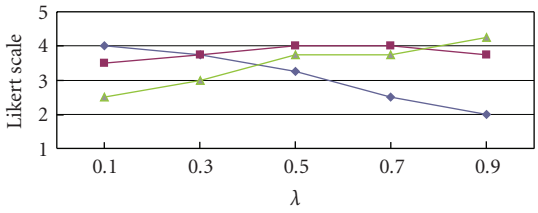


Figure 11: Comparative event detection experiments between semantics-matching and timestamp recognition methods on (a) basketball and (b) football matches.



Consistency	4	3.75	3.25	2.5	2
Conciseness	3.5	3.75	4	4	3.75
Converge	2.5	3	3.75	3.75	4.25

Figure 12: User evaluation of personalized video customization.

matching algorithm is applied to two sequences with obvious length difference (multiple long shots vs. single event) in the shot-level sequence matching.

7.2. Personalized Video Customization. Before we give the experimental results of personalized video customization, we first mention the setting of the adjustable parameters for event importance computation. As can be seen from (6) and (7), α and β are control coefficients to decide the effects of event rank and occurrence time on event importance computation. In our following experiments, both coefficients are set to 1, meaning that event rank and occurrence time fully affect the event semantic significance. In addition, the adjustable parameter γ in (8) is set to 0.5, denoting the equal weights for event type and involved player. As for the preference learning, the constant gain and damping is set to 0.1 and 0.99, respectively, implying the concept preference degree should be increased or decreased gradually. Based on the concept network, the Dist function used in (8) is computed as follows:

$$\mathrm{Dist}(E_i, U) = \begin{cases} 0, & \text{direct connection,} \\ 1 - \dfrac{L_{ki}}{\sum_l L_{kl}}, & \text{indirect connection,} \\ 1, & \text{no connection,} \end{cases} \tag{19}$$

where E_i and U represent the semantic event and user request, respectively, L_{ki} denotes the semantic context similarity between the kth and ith concepts in social network, where the kth concept is customized by U and the ith concept is contained in E_i. The "direct connection" refers to the case when E_i and U are consistent on the player or event type, and the "indirect connection" refers to the case when concept in E_i is not contained in U but indirectly connected with U in the concept network. As for the "no connection" case, neither direct nor indirect relationship can be found between E_i and U.

Table 6: Customized video segments under different fusion parameters.

λ	ID	Moment	Player	Event	Duration (s)
0.1	45	02 : 39	Steve Nash	three point	3.16
	33	06 : 41	Steve Nash	layup	3.08
	11	10 : 22	Steve Nash	jumper	3.40
0.5	54	01 : 10	Grant Hill	three point	3.65
	45	02 : 39	Steve Nash	three point	3.16
	33	06 : 41	Steve Nash	layup	3.08
0.9	54	01 : 10	Grant Hill	three point	3.65
	45	02 : 39	Steve Nash	three point	3.16
	50	01 : 54	Charlie Villanueva	jumper	3.10

Table 7: Learned user preference on player concepts.

Users	Player concept learning results (on 3 football and 2 basketball matches)				
	1st	2nd	3 rd	4th	AP
No. 1	Zidane (12, 3, 4)	Totti (14, 2, 1)	Henry (6, 12, 8)	Piero (10, 1, 3)	(0.83, 0.73)
No. 2	Messi (16, 1, 1)	Gonzalez (8, 6, 7)	Schweinsteiger (10, 4, 5)	Ballack (10, 2, 4)	(0.85, 0.67)
No. 3	Zidane (15, 1, 2)	Ronaldo (13, 2, 4)	Ronaldinho (12, 3, 3)	Carlos (8, 9, 8)	(0.86, 0.60)
No. 4	Redd (15, 2, 1)	Nash (14, 3, 2)	Yi (9, 6, 6)	Bogut (5, 7, 8)	(0.77, 0.75)
No. 5	Kobe (12, 3, 2)	Yao (12, 2, 1)	Gasol (9, 5, 7)	Odom (4, 8, 11)	(0.78, 0.70)

Due to the intrinsic subjectivity of personalized video customization, we carry out a user study to evaluate the performance of our customization system. Our study involves 12 volunteers, who are all first-time users of the customization system and have certain knowledge about basketball and football games. For each user, after watching 3 or 4 integrated matches, he/she is asked to use the system to customize their favorite video clips with various personalized preference, such as players, events and time, under different fusion parameters.

To validate the effectiveness of the proposed personalized customization algorithm, a specially designed questionnaire is handed out to the participants to get their feedbacks to the generated video content. Motivated by the work in [46], we define a new "3C" criterion in the questionnaire as follows:

(1) Consistency. Whether the generated video clip is consistent with user request on content semantics, such as involved players and event types.

(2) Conciseness. Whether the generated video clip capture the main body of the match without including irrelevant events.

(3) Coverage. Whether the generated video clip covers all important events happening in the match under current viewing time limit.

All above three indexes need to be answered on a five-grade Likert scale [47] where 1 denotes strongly reject, 2 reject, 3 marginally accept, 4 accept, and 5 strongly accept.

As can be seen from Figure 12, when the fusion parameter λ increases from 0.1 to 0.9, average user evaluation on result consistency gradually declines from 4 to 2. Meanwhile, the evaluation on result coverage behaves oppositely, rising from 2.5 to 4.25. This contrast reveals the important role that λ plays in balancing the game content and user preference in the video customization process. When λ is small, the algorithm will emphasize more on user's request, hence more semantically related events are selected. On the contrary, when λ is big, say equals to 0.9, the result is largely up to the match status and thus the globally interesting events are presented. Table 6 lists the customization results under different fusion parameters in response to the user request "selecting 10 seconds highlight about Steve Nash's shot event in the first quarter of NBA 2008 Suns vs. Bucks". As shown in the table, the selected segments are highly related to user request when λ equals to 0.1. As λ is growing from 0.5 to 0.9, the customized video clips gradually change to other more important shot events in the end the match. In those cases, user request is not strictly obeyed and followed, for example, the involved player.

7.3. System Adaptation. In this section, we investigate whether the proposed system can adaptively acquire user's

Table 8: Learned user preference on event concepts.

Users	Event concept learning results (on 3 football and 2 basketball matches)				
	1st	2nd	3rd	4th	AP
No. 6	Shot (10, 2, 1)	Free Kick (4, 4, 6)	Corner (8, 5, 5)	Red Card (2, 6, 7)	(0.57, 0.62)
No. 7	Goal (2, 6, 8)	Shot (5, 1, 2)	Offside (4, 5, 4)	Yellow Card (5, 3, 1)	(0.73, 0.81)
No. 8	Foul (5, 3, 4)	Corner (10, 1, 1)	Yellow Card (3, 5, 5)	Shot (1, 6, 7)	(0.73, 0.67)
No. 9	Shot (12, 1, 1)	Foul (10, 2, 2)	Stolen (5, 6, 5)	Rebound (9, 3, 4)	(0.92, 0.88)
No. 10	Shot (10, 2, 1)	Miss (5, 5, 6)	Foul (8, 1, 2)	Block (8, 3, 5)	(0.95, 0.82)

personalized preference from the customization process. For the convenience of result evaluation, a set of preselected concepts, including players and events, are given to users so that they can focus attention on these semantics in their following operations. Each user is assigned a sports match with 4 emphasis concepts, and is asked to use the proposed system for one or two hours, evoking the procedure of user preference leaning. After about 50 rounds of interaction, we sort the concepts in accordance with their preference weights, and list the ranks of those preselected concepts in the sorted sequence. To properly measure the system learning ability, we adopt the average precision (AP) index used in information retrieval to objectively depict the system's learning ability with user interaction.

Tables 7 and 8 show the experimental results of learned user preference on player and event concepts in football and basketball matches. In both tables, each row corresponds to a user preference learning result on one match, and each cell of its central part is filled with an emphasized concept name and a triple group, marked as (#F, #R1, #R2), where #F, #R1 and #R2 denote the concept queried frequency and its final rankings with and without network-based preference propagation. Similarly, the AP result corresponding to each match is a pair of numbers, marked as (#AP1, #AP2), representing the AP result of the customization system with and without concept network analysis. Take the up-left cell in Table 7 as an example, the user beforehand selected the most interesting player is "Zinedine". After 12 queries with that keyword, the derived weight rankings are 3 and 4, respectively, and the related AP results on the four preselected concepts are 0.83 and 0.73, respectively, corresponding to the learning algorithm with and without using concept network.

The experimental results indicate that 75% and 60% of the predecided player concept and 60% and 55% of those for event concepts were placed in the top-4 rank of the learned concept sequence with and without concept social network analysis (CSNA), respectively. Both of which reflect the effectiveness of the proposed user preference learning algorithm, especially the network-based preference propagation strategy. For the player concept learning in Table 7, CSNA acts more effectively on football matches than basketball matches, which is mainly due to the player context difference. Specifically, the selected football players are highly connected in attack events than that of basketball players (because the basketball players are usually involved in both offense and defense events in the match), thus it is more easily for the proposed CSNA algorithm to locate those football forwards than those in basketball matches. Similar explanations can be also applied to the event learning result in Table 8. In summary, the proposed user preference learning algorithm can effectively acquire user preference from their customization operations, and concept network analysis further improves the system performance especially when users appetite is fixed and specified.

8. Conclusion

Video personalization is an important mechanism to provide particular viewers with their favorite content. Considering the diverse content preference and environment limitations, detailed video semantics should be fully mined and reasonably evaluated so that suited video segments can be collected for the specific user. This is a challenging task and its difficulty embodies in every submodules including video annotation, personalized customization and system adaptation.

In this paper, we presented an integral framework for personalized sports video customization. In the off-line annotation, a hierarchical video-text matching method is raised to align the multimedia information based on their semantic correspondence, which generates both refined and accurate video content description. In the on-line customization, a user-participant multiconstraint 0/1 Knapsack model is proposed to realize semantic content retrieval and summarization under resource-limited condition. To facilitate the above on-line customization process, a concept network based system adaptation algorithm is designed to implicitly infer the complete user preference.

The approach's complexity focuses on the semantic video annotation. Compared with the simple web-text analysis, intensive video processing, for example, camera motion estimation and replay shot detection, costs the majority of

computation resources. Moreover, since semantic-matching algorithm needs the integral video and text sequences as its inputs, the alignment result will not come out unless the whole match is over. For these reasons, video annotation is performed in an off-line manner in our approach. However, once the semantic annotation is obtained, video customization and system adaptation (concept weight adjustment) only involve some mathematical calculations, and thus can achieve real-time processing. Both quantitative and qualitative experiments conducted on more than 1000 minutes sports video validate the effectiveness of the proposed approach.

Another point needs to be addressed is the expansibility of the proposed framework. Since the on-line customization and adaptation are irrelevant with specific sports genres, the expansibility is mainly up to the video annotation. Although our work is implemented on the field sports, like football and basketball, we think the basic idea and framework of the semantic matching is general and can be easily adapted to other opponent sports. The evidences supporting our thought come from two sides: First, live casting text is now a standard service in famous sports websites, such as ESPN and BBC, which covers the majority of popular sports genres in our daily lives. Hence, there is no lack of textual descriptions for sports video annotation. Second, the notion of "attack" exists in most opponent sports and has very clear semantics, hence its robust detection in other sports types is also feasible (although may be not exactly the same as our current methods for football and basketball matches). Therefore, with some necessary modifications according to the specific environment, the basic framework of semantic matching can still be applied in other sports genres.

In the future, we plan to improve current work in both theoretic and application aspects. In semantic video annotation, Bayesian network (BN) is adopted to tag the semantic content of video segment and sequence matching algorithm is applied to align the video and text tag sequences. Such a method ignores the temporal dependence of neighboring semantic tags and thus may impair the final matching precision. To cover this shortage, more advanced sequential models, like hidden Markov model (HMM), or dynamic Bayesian network (DBN), can be employed to model the text facilitated video annotation, where video-text alignment can be treated as a hidden state inference problem and solved with Veterbi-like inference algorithm. On the other hand, video personalization in this paper mainly focuses on the content selection. However, as a practical system, problems such as video encoding and decoding in different communication channels and video displaying on various devices are also of great importance. To the end user, video content selection, data transmission, and terminal displaying constitute an integral solution for personalized video customization.

Acknowledgment

This work is supported by Natural Science Foundation of China (no. 90920303).

References

[1] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 105–115, Los Angeles, Calif, USA, 2000.

[2] M. Xu, N. C. Maddage, C. S. Xu, M. S. Kakanhalli, and Q. Tian, "Creating audio keywords for event detection in soccer video," in *Proceeding of the IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. 281–284, Baltimore, Md, USA, 2003.

[3] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, "Audio events detection based highlight extraction from baseball, golf and soccer games in a United Framework," in *Proceeding of the IEEE International Conference on Acuoustics, Speech and Signal Processing (ICASSP '03)*, vol. 5, pp. 632–635, Hong Kong, April 2003.

[4] L.-Y. Duan, M. Xu, and Q. Tian, "Semantic shot classification in sports video," in *Storage and Retrieval for Media Database*, vol. 5021 of *Proceedings of SPIE*, pp. 300–313, January 2003.

[5] Y.-P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 133–146, 2000.

[6] D. Zhang and S.-F. Chang, "Event detection in baseball video using superimposed caption recognition," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 315–318, Juan-les-Pins, France, December 2002.

[7] J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, and W. Nunziati, "Semantic annotation of soccer videos: automatic highlights identification," *Computer Vision and Image Understanding*, vol. 92, no. 2-3, pp. 285–305, 2003.

[8] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transactions on Image Processing*, vol. 12, no. 7, pp. 796–807, 2003.

[9] A. S. David and E. O. C. Eoel, "Event detection in Field sports video using audiovisual features and a support vector machine," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1225–1233, 2005.

[10] C.-L. Huang, H.-C. Shih, and C.-Y. Chao, "Semantic analysis of soccer video using dynamic Bayesian network," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 749–760, 2006.

[11] L. Xing, Q. Ye, W. Zhang, Q. Huang, and H. Yu, "A scheme for racquet sports video analysis with the combination of audio-visual information," in *Visual Communications and Image Processing*, vol. 5960 of *Proceedings of SPIE*, pp. 259–267, 2005.

[12] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, and C.-S. Xu, "A mid-level representation framework for semantic sports video analysis," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 33–44, Berkeley, Calif, USA, 2003.

[13] M. Xu, L. Duan, C. Xu, M. Kankanhalli, and Q. Tian, "Event detection in basketball video using multi-modalities," in *Proceeding of IEEE Pacific Rim Conference on Multimedia (PCM '03)*, vol. 3, pp. 1526–1530, Singapore, December 2003.

[14] K. Wan and C. Xu, "Efficient multimodal features for automatic soccer highlight generation," in *Proceedings of the International Conference on Pattern Recognition (ICPR '04)*, vol. 3, pp. 973–976, Cambridge, UK, August 2004.

[15] M. H. Kolekar and S. Sengupta, "A hierarchical framework for generic sports video classification," in *Proceedings of the 7th Asian Conference on Computer Vision (ACCV '06)*, vol. 3852

of *Lecture Notes in Computer Science*, pp. 633–642, Springer, Hyderabad, India, January 2006.

[16] M. Xu, L.-Y. Duan, C.-S. Xu, and Q. Tian, "A fusion scheme of visual and auditory modalities for event detection in sports video," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '03)*, vol. 3, pp. 189–192, Hong Kong, 2003.

[17] M. Han, W. Hua, W. Xu, and Y. Gong, "An integrated baseball digest system using maximum entropy method," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 347–350, Juan-les-Pins, France, December 2002.

[18] M. H. Kolekar and S. Sengupta, "Event-importance based customized and automatic cricket highlight generation," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 1617–1620, July 2006.

[19] C. Jung and J. Kim, "Player information extraction for semantic annotation in golf videos," *IEEE Transactions on Broadcasting*, vol. 55, no. 1, pp. 79–83, 2009.

[20] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 68–75, 2002.

[21] H. Xu and T.-S. Chua, "The fusion of audio-visual features and external knowledge for event detection in team sports video," in *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR '04)*, pp. 127–134, New York, NY, USA, October 2004.

[22] C. S. Xu, J. Wang, K. Wan, Y. Li, and L. Duan, "Live sports event detection based on broadcast video and web-casting text," in *Proceedings of the 14th Annual ACM International Conference on Multimedia (MM '06)*, pp. 221–230, Santa Barbara, Calif, USA, October 2006.

[23] D. Tjondronegoro, Y.-P. P. Chen, and B. Pham, "Integrating highlights for more complete sports video summarization," *IEEE Multimedia*, vol. 11, no. 4, pp. 22–37, 2004.

[24] A. Ekin and A. M. Tekalp, "Generic play-break event detection summarization and hierarchical sports video analysis," in *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME '03)*, vol. 1, pp. 167–172, Baltimore, Md, USA, July 2003.

[25] M. Fleischman and D. Roy, "Situated models of meaning for sports video retrieval," in *Proceeding of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL '07)*, pp. 37–44, Rochester, NY, USA, April 2007.

[26] N. Babaguchi, K. Ohara, and T. Ogura, "Learning personal preference from viewer's operations for browsing and its application to baseball video retrieval and summarization," *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 1016–1025, 2007.

[27] W. Gao, Q.-M. Huang, S. Q. Jiang, and P. Zhang, "Sports video summarization and adaptation for application in mobile communication," *Journal of Zhejiang University: Science A*, vol. 7, no. 5, pp. 819–829, 2006.

[28] Y. Wei, S. M. Bhandarkar, and K. Li, "Video personalization in resource-constrained multimedia environments," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 902–911, Augsburg, Germany, September 2007.

[29] B. L. Tseng, C. Y. Lin, and J. R. Smith, "Video summarization and personalization for pervasive mobile devices," in *Storage and Retrieval for Media Database*, vol. 4676 of *Proceedings of SPIE*, pp. 359–370, 2002.

[30] Y. Zhang, X. Zhang, C. Xu, and H. Lu, "Personalized retrieval of sports video," in *Proceedings of the International Workshop on Multimedia Information Retrieval (MIR '07)*, pp. 313–322, Augsburg, Germany, September 2007.

[31] A. Amir, M. Berg, and H. Permuter, "Mutual relevance feedback for multimodal query formulation in video retrieval," in *Proceedings of International Workshop on Multimedia Information Retrieval (MIR '05)*, pp. 17–24, Singapore, November 2005.

[32] T. Syeda-Mahmood and D. Ponceleon, "Learning video browsing behavior and its application in the generation of video previews," in *Proceedings of the 9th ACM International Conference on Multimedia (MM '01)*, pp. 119–128, Ottawa, Canada, September 2001.

[33] J. Zimmerman, K. Kurapati, A. L. Buczak, D. Schaffer, S. Gutta, and J. Martino, "TV personalization system: design of a TV show recommender engine and interface," in *Personalized Digital Television: Targeting Programs to Individual Viewers*, pp. 27–51, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[34] A. Jaimes, N. Sebe, and D. Gatica-Perez, "Human-centered computing: a multimedia perspective," in *Proceedings of the 14th Annual ACM International Conference on Multimedia (MM '06)*, pp. 855–864, Santa Barbara, Calif, USA, September 2006.

[35] C. Xu, J. J. Wang, H. Q. Lu, and Y. F. Zhang, "A novel framework for semantic annotation and personalized retrieval of sports video," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 421–436, 2008.

[36] C. Liang, Y. Zhang, C. S. Xu, J. Q. Wang, and H. Q. Lu, "A hierarchical semanticsmatching approach for sports video annotation," in *Proceedings of the 10th Pacific Rim Conference on Multimedia (PCM '09)*, vol. 5879 of *Lecture Notes in Computer Science*, pp. 684–696, Bangkok, Thailand, December 2009.

[37] L. Wang, M. Lew, and G. Xu, "Offense based temporal segmentation for event detection in soccer video," in *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR '04)*, pp. 259–266, New York, NY, USA, October 2004.

[38] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 497–501, 2000.

[39] C. Xu, Y. F. Zhang, G. Y. Zhu, Y. Rui, H. Q. Lu, and Q. M. Huang, "Using webcast text for semantic event detection in broadcast sports video," *IEEE Transaction on Multimedia*, vol. 10, no. 7, pp. 1342–1355, 2008.

[40] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[41] N. Babaguchi, Y. Kawai, T. Ogura, and T. Kitahashi, "Personalized abstraction of broadcasted American football video by highlight selection," *IEEE Transactions on Multimedia*, vol. 6, no. 4, pp. 575–586, 2004.

[42] B. Merialdo, K. T. Lee, D. Luparello, and J. Roudaire, "Automatic construction of personalized TV News programs," in *Proceedings of the ACM International Multimedia Conference*, pp. 323–331, Orlando, Fla, USA, October 1999.

[43] J. Scott, *Social Network Analysis: A Handbook*, Sage, Newbury Park, Calif, USA, 1991.

[44] C.-Y. Weng, W.-T. Chu, and J.-L. Wu, "RoleNet: movie analysis from the perspective of social networks," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 256–271, 2009.

[45] A. Vinciarelli and S. Favre, "Broadcast news story segmentation using social network analysis and hidden Markov models," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 261–264, Augsburg, Germany, September 2007.

[46] L. He, E. Sanocki, A. Gupta, and J. Grudin, "Auto-summarization of audio-video presentations," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 489–498, Orlando, Fla, USA, October 1999.

[47] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 22, no. 140, pp. 1–55, 1932.

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2010, Article ID 486487, 18 pages
doi:10.1155/2010/486487

Research Article

Multimodal Indexing of Multilingual News Video

Hiranmay Ghosh,[1] Sunil Kumar Kopparapu,[2] Tanushyam Chattopadhyay,[3] Ashish Khare,[1] Sujal Subhash Wattamwar,[1] Amarendra Gorai,[1] and Meghna Pandharipande[2]

[1] *TCS Innovation Labs Delhi, TCS Towers, 249 D&E Udyog Vihar Phase IV, Gurgaon 122015, India*
[2] *TCS Innovation Labs Mumbai, Yantra Park, Pokhran Road no. 2, Thane West 400601, India*
[3] *TCS Innovation Labs Kolkata, Plot A2, M2-N2 Sector 5, Block GP, Salt Lake Electronics Complex, Kolkata 700091, India*

Correspondence should be addressed to Hiranmay Ghosh, hiranmay.ghosh@tcs.com

Received 16 September 2009; Revised 27 December 2009; Accepted 2 March 2010

Academic Editor: Ling Shao

The problems associated with automatic analysis of news telecasts are more severe in a country like India, where there are many national and regional language channels, besides English. In this paper, we present a framework for multimodal analysis of multilingual news telecasts, which can be augmented with tools and techniques for specific news analytics tasks. Further, we focus on a set of techniques for automatic indexing of the news stories based on keywords spotted in speech as well as on the visuals of contemporary and domain interest. English keywords are derived from RSS feed and converted to Indian language equivalents for detection in speech and on ticker texts. Restricting the keyword list to a manageable number results in drastic improvement in indexing performance. We present illustrative examples and detailed experimental results to substantiate our claim.

1. Introduction

Analysis of public newscast by domestic as well as foreign TV channels for tracking news, national and international views and public opinion is of paramount importance for media analysts in several domains, such as journalism, brand monitoring, law enforcement and internal security. The channels representing different countries, political groups, religious conglomerations, and business interests present different perspectives and viewpoints of the same event. Round the clock monitoring of hundreds of news channels requires unaffordable manpower. Moreover, the news stories of interest may be confined to a narrow slice of the total telecast time and they are often repeated several times on the news channels. Thus, round-the-clock monitoring of the channels is not only a wasteful exercise but is also prone to error because of distractions caused while viewing extraneous telecast and consequent loss of attention. This motivates a system that can automatically analyze, classify, cluster and index the news-stories of interest. In this paper we present a set of visual and audio processing techniques that helps us in achieving this goal.

While there has been significant research in multimodal analysis of news-video for their automated indexing and classification, the commercial applications are yet to mature. Commercial products like BBN Broadcast monitoring system (http://www.bbn.com/products_and_services/bbn_broadcast_monitoring_system/) and Nexidia rich media solution (http://www.nexidia.com/solutions/rich_media) offer speech analytics-based solution for news video indexing and retrieval. None of these solutions can differentiate between news programs from other TV programs and additionally cannot filter out commercials. They index the complete audio-stream and cannot define the story boundaries. Our work is motivated towards creation of a usable solution that uses multimodal cues to achieve a more effective news video analytics service. We put special emphasis on Indian broadcasts, which are primarily in English, Hindi (Indian national language), and several other regional languages.

We present a framework for multimodal analysis of multilingual news telecasts, which can be augmented with tools and techniques for specific news analytics tasks, namely delimiting programs, commercial removal, story boundary

detection and indexing of news stories. While there has been significant research in tools for each of the tasks, an overall framework for news telecast analysis has not yet been proposed in literature. Moreover, automated analysis of Indian language telecasts raises some unique challenges. Unlike most of the channels in the western world, Indian channels do not broadcast "closed captioned text", which could be gainfully employed to index the broadcast stream. Thus, we need to rely completely on audio-visual processing of the broadcast channels. Our basic approach is to index the news stories with relevant keywords discovered in speech and in form of "ticker text" on the visuals. While there are several speech processing and OCR techniques, we face significant challenges in using them for processing Indian telecasts. The major impediments are (a) low resolution (768×576) of the visual frames and (b) significant noise introduced in the analog cable transmission channels, which are still prevalent in India. We have introduced several preprocessing and postprocessing stages to audio and visual processing algorithms to overcome these difficulties. Moreover, the speech and optical character recognition (OCR) technologies for different Indian languages (including Indian English) are under various stages of development under the umbrella of TDIL project [1–5] and are far from a state of maturity. All these factors lead to difficulties in creating a reliable transcript of the spoken or the visual text. We have improved the robustness of the system by restricting the audio-visual processing tasks to discover a small set of keywords of domain interest. These keywords are derived from Really Simple Syndication (RSS) feeds pertaining to the domain of interest. Moreover, these keywords are continuously updated as new feeds arrive and thus, they relate to news stories of contemporary interest. This alleviates the problem of long turn-around time associated with manual updates of the dictionaries, which may fail to keep pace with a fast changing global scenario. We create a multilingual keyword list in English and Indian languages to enable keyword spotting in different TV channels, both in spoken and visual forms. The multilingual keyword list helps us to automatically map the spotted keywords in different Indian languages to their English (or any other language) equivalents for uniform indexing across multiple channels.

The rest of the paper is organized as follows. We review the state-of-the-art in news video analysis in Section 2. Section 3 provides the system overview. Section 4 describes the techniques adopted by us for keyword extraction from speech and visuals from multilingual channels in details. Section 5 provides an experimental evaluation of the system. Finally, Section 6 concludes the paper and provides direction for future work.

2. Related Work

We provide an overview of research in news video analytics in this section to put our work in context. There has been much research interest in automatic interpretation, indexing and retrieval of audio and video data. Semantic analysis of multimedia data is a complex problem and has been attempted with moderate success in closed domains, such as sports, surveillance and news. This section is by no means a comprehensive review on audio and video analytic techniques that has evolved over the past decade, as we concentrate on automated analysis of broadcast video.

Automated analysis, classification and indexing of news video contents have drawn the attention of many researchers in recent times. A video comprising visual and audio components leads to two complementary approaches for automated video analysis. Eickeler and Mueller [6] and Smith et al. [7] propose classification of the scenes into a few content classes based on visual features. A motion feature vector has been computed from the differences in the successive frames and HMM's have been used to characterize the content classes. In contrast, Gauvain et al. [8] proposes an audio-based approach, where the speech in multiple languages has been transcribed and the constituent words and phrases have been used to index the contents of a broadcast stream. Later work attempts to merge the two streams of research and proposes multimodal analysis, which is reviewed later in this section.

A typical news program on a TV channel is characterized by unique jingles at the beginning and the end of the newscast, which provide a convenient means to delimit the newscast from other programs [9]. Moreover, a news program has several advertisement breaks, which need to be removed for efficient news indexing. Several methods have been proposed for TV Commercial (We have used "commercial" and "advertisement" interchangeably in this paper.) detection. One simple approach is to detect the logos of the TV channels [10], which are generally absent during the commercials, but this might not hold good for many contemporary channels. Sadlier et al. [11] describes a method for identifying the ad breaks using "black" frames that generally precedes and succeeds the advertisements. The black frames are identified by analyzing the image intensity of the frames and audio intensity at those time-points. While American and European channels generally use black frames for separation of commercials and programs, it is not so for other geographical regions, including India [12]. Moreover, the heuristics used to ignore the extraneous black frames appearing at arbitrary places within programs are difficult to generalize. Hua et al. [13] have used the distinctive audio-visual properties of the commercials to train an SVM based classifier to classify video shots into commercial and noncommercial categories. The performance of such classifiers can be enhanced with application of the principle of temporal coherence [12]. Six basic visual features and five basic audio features derived context-based features have been used in [13] to classify the shots using SVM and further postprocessing.

The time-points in a streamed video can be indexed with a set of keywords, which provide the semantics of the video-segment around the time-point. Most of the American and European channels accompanied with closed caption text, which are transcripts of the speech, are aligned with the video time-line and provides a convenient mechanism for indexing a video. Where closed captioned text is not available, speech recognition technology needs to be used.

There are two distinct approaches to the problem. In phoneme-based approach [14], the sequence of phonemes constituting the speech is extracted from the audio track and is stored as metadata in sync with the video. During retrieval, a keyword is converted to a phoneme string and this phoneme string is searched for in the video metadata [15]. In contrast, [16] proposes a speaker independent continuous speech recognition engine that can create a transcript of the audio track and align it with the video. In this approach the retrieval is based on the keywords in text domain. The difference is primarily in the way the speech data is transcribed and archived. In the phoneme-based storage, there is no language dictionary used and the speech data is represented by a continuous string of phonemes. While in the later case a pronunciation dictionary is used to convert short phoneme sequences into known dictionary words and the actual phoneme sequence is not retained. Phone level approach is generally more error-prone than word-based approaches because the phoneme recognition accuracies are very poor, typically 40–50%. Moreover, word-based approach provides more robust information retrieval results [17] because in the word-based storage, a speech signal is tagged by at least 3 best (often referred to as *n*-best) phonemes (instead of only one phoneme) at each instance and the word dictionary is used to resolve which sequence of phonemes to use to be able to correlate the speech with a word in the dictionary. Additional sources of information that can be used for news video indexing constitute output from Optical Character Recognition (OCR) on the visual text, face recognizer and speaker identification [18].

Once the advertisement breaks are removed from a news-program, the latter needs to be broken down into individual news stories for further processing. Chua et al. [19] provide a survey of the different methods used based on the experience of TRECVID 2003, which defined news story segmentation as an evaluation task. One of the approaches involve analysis of speech [20, 21], namely, end-of-sentence identification and text tiling technique [22] which involves computing lexical similarity scores across a set of sentence and has been used earlier for story identification in text passages. Purely text-based approach generally yields low accuracy, motivating use of audio-visual features. Identification of anchor shots [23], cue phrases, prosody, and blank frames in different combinations are used together with certain heuristics regarding news production grammar in this approach. A third approach uses machine learning approach where an SVM or a Maximum Entropy classifier classifies a candidate story boundary point based on multimodal data, namely, audio, visual, and text data surrounding the point. While, some of these approaches use a large number of low-level media features, for example, face, motion, and audio classes, some others [24] proposes abstracting low level features to mid-level to accommodate multimodal features without significant increase in dimensionality. In this approach, a shot is preclassified to semantic categories, such as anchor, people, speech, sports, and so forth, which are then combined with a statistical model such as HMM [25]. The classification of shots also helps in segmenting the corpus into subdomains, resulting in more accurate models and hence, improved story-boundary detection. Besacier et al. [26] report use of long pause, shot boundary, audio change (speaker change, speech to music transition, etc.), jingle detection, commercial detection and ASR output for story boundary detection. TRECVID prescribes use of F1 Score [27], the harmonic mean of precision and recall, as a measure of the accuracy. An accuracy of F1 = 0.75 for multimodal story boundary detection has been reported in [22].

Further work on news video analysis extends to conceptual classification of stories. Early work on the subject [23] achieves binary classification shots to a few predefined semantic categories, like "indoors" versus "outdoor", "nature" versus "man-made", and so forth. This was done by extracting the visual features of the key-frames and using a SVM classifier. Higher level inferences could be drawn by observing co-occurrence of some of these semantic levels, for example, occurrence of "sky", "water", "sand", and "people" on a video frame implied a "beach scene". Later work has found that the performance of concept detection is significantly improved by use of multimodal data, namely audio-visual features and ASR transcripts [24]. A generic approach for multimodal concept detection that combines outputs of multiple unimodal classifiers by ensemble fusion has been found to perform better than early fusion approach that aggregates multimodal features into a single classifier. Colace et al. [28] introduced a probabilistic framework for combining multimodal features for classifying the video shots in a few predefined categories using Bayesian Networks. The advantage of Bayesian classifiers over binary classifiers is that the former not only classifies the shots but also ranks the classification. While judicious combination of multimodal improves the performance of concept detection, it has also been observed that use of query-independent weights to combine multiple features performs worst than text alone. Thus, the above approaches for shot classification could not scale beyond a few predefined conceptual categories. This prompts use of external knowledge to select appropriate feature-weights for specific query classes [18]. Harit et al. [29] provide a new approach to use an ontology that can be used to reason with media properties of concepts and to dynamically derive a Bayesian Network for scene classification in a query context. Topic clustering, or clustering news-videos at different times and from different sources is another area of interest. An interesting open question has been the use of audio-visual features in conjunction with text obtained from automatic speech recognition in discovering novel topics [24]. Another interesting research direction is to investigate video topic detection in absence of Automatic Speech Recognition (ASR) data as in the case of "foreign" language news video [24].

3. Framework for Telecast News Analysis

We envisage a system where a large number of TV broadcast channels are to be monitored by a limited number of human monitor. The channels are in English, Hindi (National language of India), and a few other Indian regional

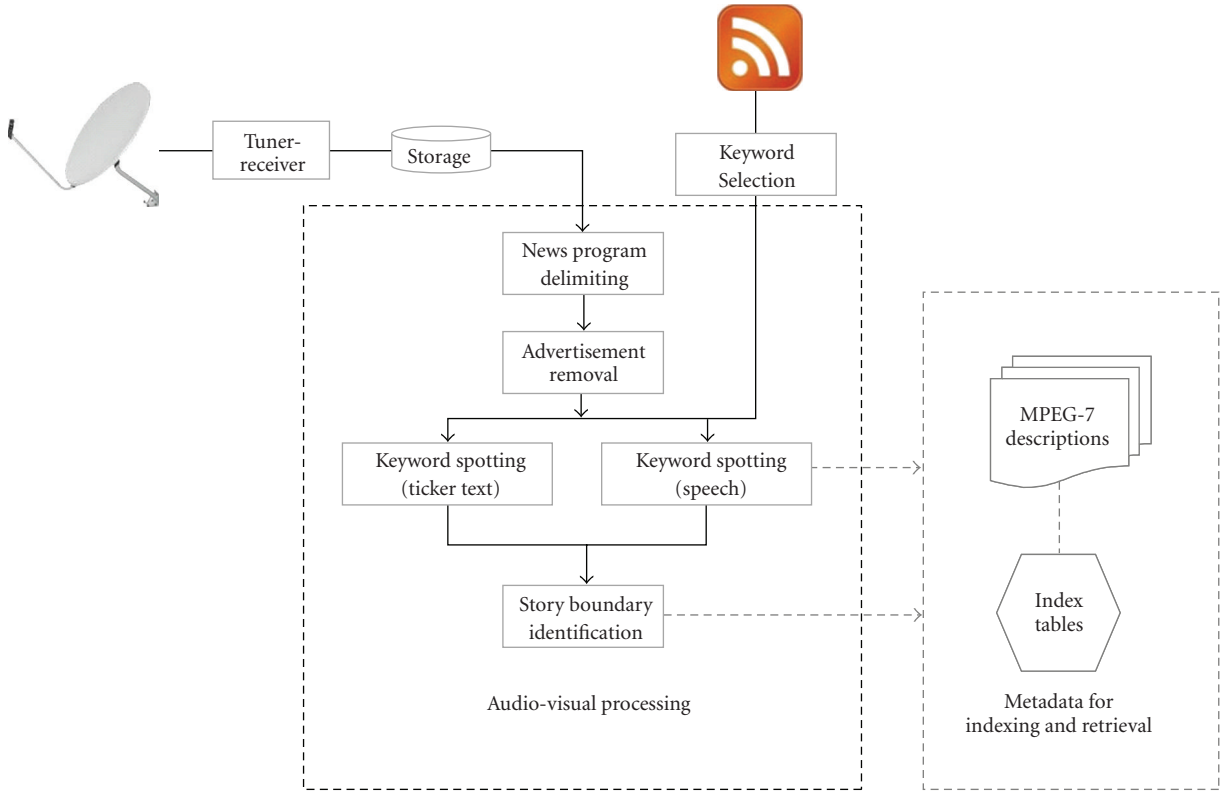


Figure 1: System architecture.

languages. Many of the channels are news channels but some are entertainment channels, which have specific time-slots for news. The contents of the news channels contain weather reports, talk shows, interviews and other such programs besides news. The programs are interspersed with commercial breaks. The present work focuses on indexing news and related programs only.

Figure 1 depicts the system architecture. At the first step of processing, the broadcast streams are captured from Direct to House (DTH) systems and are decoded. They are initially dumped on the disk in chunks of manageable size. These dumps are first preprocessed to identify the news programs. While the time-slots for news on the different channels are known, the accurate boundaries of the programs are identified with the unique jingles that characterize the different programs on a TV channel [9]. The next processing step is to filter out the commercial breaks. Since the black frame-based method does not work for most of the Indian channels, we propose to use a supervised training method [13] for this purpose. At the end of this stage, we get delimited news programs devoid of any commercial breaks.

The semantics of the news contents are generally characterized by a set of keywords (or key phrases) which occur either in the narration of the newscaster or in the ticker text [30] that appears on the screen. The next stage of processing involves indexing the video stream with these extracted keywords. Many American and European channels broadcast transcript of the speech as closed captioned text, which can be used for convenient indexing of the news stream. Since there is no closed captioning available with Indian news channels, we use image and speech processing techniques to detect keywords from both visual and spoken audio track. The video is decomposed into constituent shots, which are then classified into different semantic categories [7, 28], for example, field-shots, news-anchor, interview, and so forth—this classification information is used in the later stages of processing. We create an MPEG-7 compliant content description of the news video in terms of its temporal structure (sequence of shots), their semantic classes and the keywords associated with each shot. An index table of keywords is also created and linked to the content description of the video. The next step in processing is to detect the story boundaries. We propose to use multi-modal cues, visual, audio, ASR output, and OCR data, to identify the story boundaries. We select some of the methods described in [19]. Late fusion method is preferred because of lower dimensionality of features in the supervised training methods and better accuracy [24]. Once the story boundaries are known, analysis of keywords spotted in the story leads to their semantic classification.

In rest of this paper, we deal with the specific problem of indexing the multilingual Indian newscasts with keywords identified in the visuals (ticker text) and in the audio (speech) and improving the indexing performance of news stories with multimodal cues.

4. Keyword-Based Indexing of News Videos

This stage involves indexing of a news video stream with a set of useful keywords and key-phrases (We will use the "keywords" and "key-phrases" interchangeably further in this section.). Since closed captioned text is not available with Indian telecasts, we need to rely on speech processing to extract the keywords. Creating a complete transcript of the speech as in [8] is not possible for Indian language telecasts because of limitations in the speech recognition technology. A pragmatic and more robust alternative is to spot a finite set of contemporary keywords of interest in different Indian languages in the broadcast audio stream. The keywords are extracted from a contemporary RSS feed [31]. We complement this approach with spotting the important keywords in the ticker text that is superimposed on the visuals on a TV channel. While OCR technologies for many Indian languages used for ticker text analysis are also not sufficiently robust, extraction of keywords from both audio and visual channels simultaneously, significantly enhances the robustness of the indexing process.

4.1. Creation of a Keyword File. RSS feeds, made available and maintained by websites of the broadcasting channels or by purely web-based news portals, captures the contemporary news in a semistructured XML format. They contain links to the full-text news stories in English. We select the common and proper nouns in the RSS feed text and the associated stories as the keywords. These proper nouns (typically names of people and places) are identified by a named entity detection module [32] while the common nouns can be identified using frequency count. A significant advantage of obtaining a keyword list from the RSS feeds is the currency of the keywords because of dynamic updates of the RSS feeds. Moreover, the RSS feeds are generally classified into several categories, for example, "business-news" and "international", and it is possible to select the news in one or a few categories that pertains to analyst's domain of interest. Restricting the keyword list to a small number helps in improving the accuracy of the system, especially for keyword spotting in speech.

The English keywords so derived, form a set of concepts, which need to be identified in both speech and visual forms from different Indian language telecasts. While there are some RSS feeds in Hindi and other Indian Languages (For instance, see http://www.voanews.com/bangla/rss.cfm (Bangla), http://feeds.feedburner.com/oneindia-thatstelugu-all (Telugu) and http://feeds.feedburner.com/oneindia-thatshindi-all (Hindi).), aligning the keywords from independent RSS feeds proves to be difficult. We derive the equivalent keywords in Indian languages from the English keywords, each of which is either a proper or a common noun. We use a word level English-to-Indian language dictionary to find the equivalent common noun keywords in an Indian language. We use a pronunciation lexicon (A lexicon is an association of words and their phonetic transcription. It is a special kind of dictionary that maps a word to all the possible phonemic representations of the word.) for transliterating proper names in a semi-automatic matter as suggested in [15]. It is to be noted that (a) the translation of the keyword in English is possible only when the keyword is present in the dictionary else it is transliterated and (b) transliteration of nouns in Indian languages are phonetic and hence there are no transliteration problems that are more visible in a nonphonetic language like English.

```
<RULE NAME="KeyWord">
  <L PROPNAME="keyword">
    <CONCEPT NAME= "Afghanistan">
      <ENG KEY= "Afghanistan">Afghanistan</ENG>
      <BEN KEY= "Afganistan"> আফগানিs ান </ BEN>
      <HIN KEY= "Afganistan">अफगानिस्तान </ HIN>
      <TEL KEY= "Afganistan" >అఫగానిస్తన </ TEL>
    </CONCEPT>
    <CONCEPT NAME= "Rajshekhar">
      <ENG KEY= "Rajshekhar">Rajshekhar</ENG>
      <BEN KEY= "Rajshekhar">বাজশেখব</BEN>
      <HIN KEY= "Rajshekhar">राजशेखर </HIN>
      <TEL KEY= "Rajshekhar"> రాజశేఖర్ </TEL>
    </CONCEPT>
    <CONCEPT NAME= " Terrorist">
      <ENG KEY= "Terrorist ">Terrorist </ENG>
      <BEN KEY= "Santrasbaadi "> সন্ত্রাসবাদী </BEN>
      <HIN KEY= "Atankabaadi ">आतंकबादी </HIN>
      <TEL KEY= "Atankavaadi "> అతన్కవాది </TEL>
    </CONCEPT>
  </L>
</RULE NAME>
```

Figure 2: Keyword list structure.

Finally, the keywords in English and their Indian language equivalents and their pronunciation keys are stored as a multilingual dynamic keyword list structure in XML format. This becomes an active keyword list for the news video channels and is used for both keyword spotting in speech and OCR. We show a few sample entries from a multilingual keyword list file in Figure 2. The first two entries represent proper nouns, the name of a place (Afghanistan) and a person (Rajashekar), respectively. The third entry (terrorist) corresponds to a common noun. In Figure 2 every concept is expressed in three major Indian languages, Bangla, Hindi, and Telugu, besides English. We use ISO 639-3 codes (See http://www.sil.org/iso639-3/.) to represent the languages. KEY entries represent pronunciation keys and are used for keyword spotting in speech. The words in Indian languages are encoded in Unicode (UTF-8) and are used as dictionary entries for correcting OCR mistakes. Each concept is associated with a NAME in English, which is returned when a keyword (speech or ticker text) in any of the languages is spotted either in speech or ticker-text, thus resulting in a built-in machine translation.

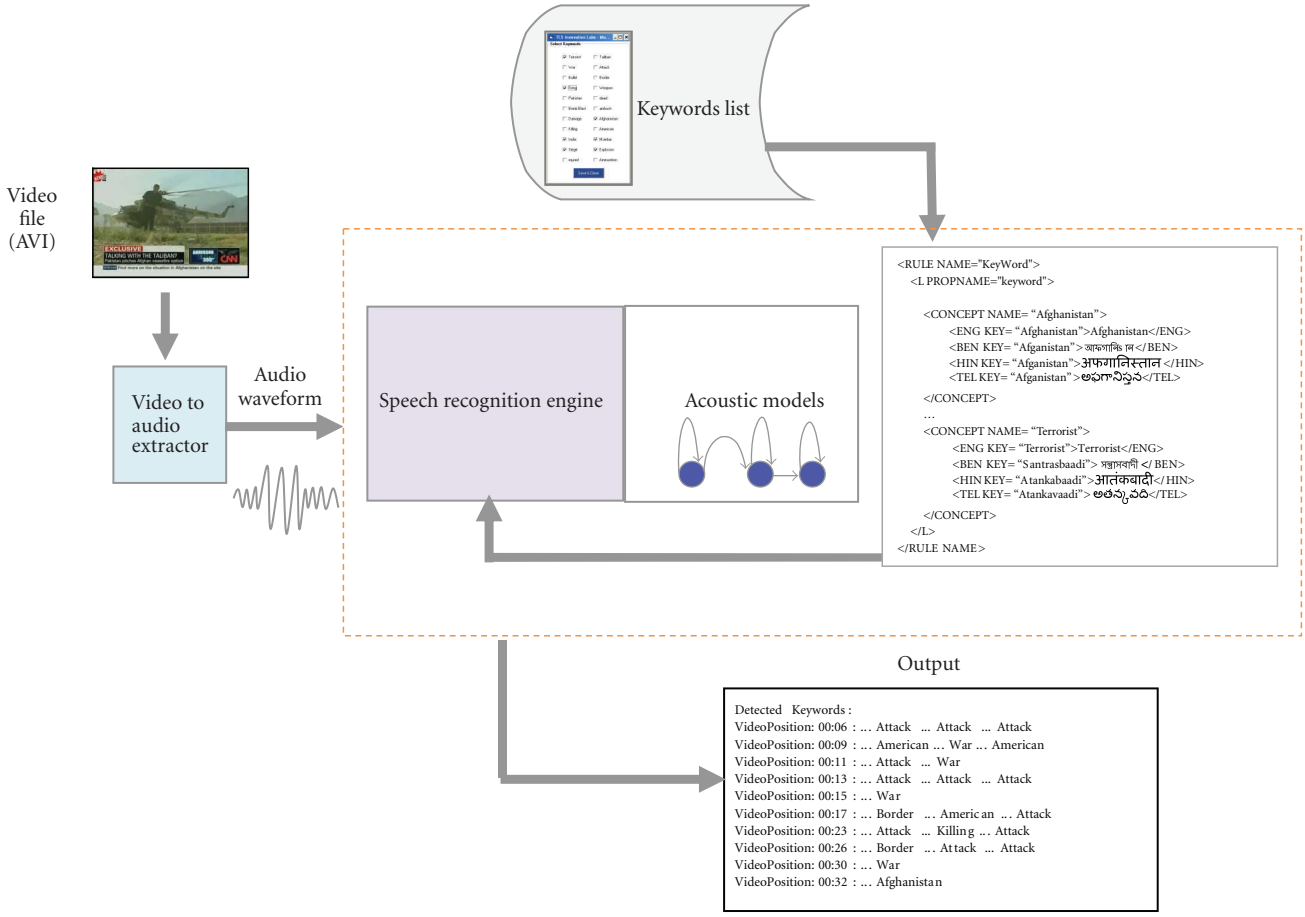


Figure 3: Typical block diagram of a keyword spotting system.

4.2. Keyword Spotting and Extraction from Broadcast News. Audio keyword spotting system essentially enables identification of words or phrases of interest in an audio broadcast or in the audio track of a video broadcast. Almost all the audio keyword spotting systems take the acoustic speech signal (a time sequence, $x(t)$) as input and use a set of (N) keywords or key phrases ($\{K_i\}_{i=1}^{N}$), as reference to spot the occurrences of these keywords in the broadcast [33]. A speech recognition engine ($S : x(t) \rightarrow x(s)$; $x(s)$ is a string sequence $\{s_k\}_{k=1}^{N}$), which is generally speaker independent and large vocabulary, is employed and is ideally supported by the list of keywords that need to be spotted (if $x(s) \in \{K_i\}_{i=1}^{N}$; then S, the speech recognition engine, is deemed to have spotted a keyword). Internally, the speech recognition engine has a built in pronunciation lexicon which is used to associate the words in the keyword list with the recognized phonemic string from the acoustic audio.

A typical functional keyword spotting system is shown in Figure 3. The block diagram shows as a first step the audio track extraction from a video broadcast. The keyword list is the list of keywords or phrases that the system is supposed to identify and locate in the audio stream. Typically this human readable keyword list is converted into a speech grammar file (FSG (finite state grammar) and CFG (context free grammar) are typically grammar used in speech recognition literature.). The speech recognition engine (in Figure 3) makes use of the acoustic models and the speech grammar file to ear mark all possible occurrences of the keywords in the acoustic stream. The output is typically the recognized or spotted words and the time instance at which that particular keyword occurred.

An audio KWS system for broadcast news has been proposed in [34]. The authors suggest the use of utterance verification (using dynamic time warping), out-of-vocabulary rejection, audio classification, and noise reduction to enhance the keyword spotting performance. They experimented on Korean news based on 50 keywords. More recent works include searching multilingual audiovisual documents using the International Phonetic Alphabet (IPA) [35] and transcription of Greek broadcast news using the HMM toolkit (HTK) [36]. We propose a multichannel, multilingual audio KWS system which can be used as a first step in broadcast news clustering.

In a multi channel, multilingual news broadcast scenario the first step towards coarse clustering of broadcast news can be achieved through audio KWS. As mentioned in earlier section broadcast news typically deals with people (including organizations and groups) and places; this makes broadcast news very rich in proper names which have to be spotted in audio. Notice that these words to be spotted

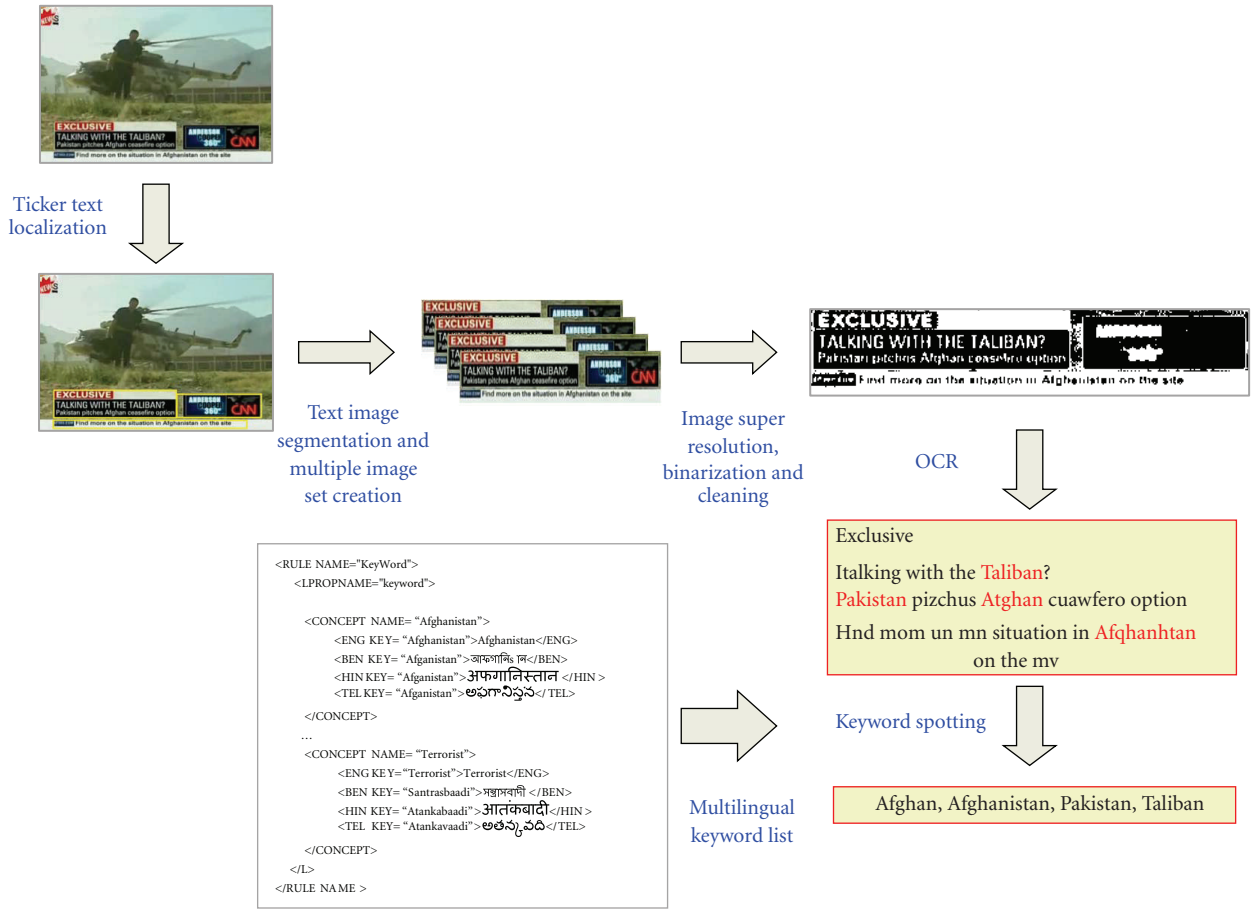


Figure 4: Keyword extraction from ticker text.

are largely language independent, the language independence comes because most of the Indian proper names are pronounced similarly in different Indian languages, implying that the same set of keywords or grammar files can be used irrespective of language of broadcast. In some sense we do not need to (a) identify the language being broadcast and (b) maintain a separate keyword list for different language channels. However, there is a need for a pronunciation dictionary for proper names. Creating a pronunciation lexicon of proper names is time consuming unlike a conventional pronunciation dictionary containing commonly used words. Laxminarayana and Kopparapu [15] have developed a framework that allows a fast method of creating a pronunciation lexicon, specifically for Indian proper names, which are generally phonetic unlike in other languages, by constructing a cost function and identifying a basis set using a cost minimization approach.

4.3. Keyword Extraction from News Ticker Text. News Ticker refers to a small screen space dedicated to presenting headlines or some important news. It usually covers a small area of the total video frame image (approximately 10–15%). Most of the news channels use two-band tickers, each having a special purpose. For instance, the upper band is generally used to display regular text pertaining to the story which is currently on air whereas "Breaking News" or the scrolling ticker on the lower band relates to different stories or displays unimportant local news, business stocks quotes, weather bulletin, and so forth. Knowledge about the production rule of specific TV channel or program is necessary to segregate the different types of ticker texts. We attempt to identify the desired keywords specified in the multilingual keyword list in the upper band, which relates to the current news story in different Indian channels.

Figure 4 depicts an overview of the steps required for keyword spotting in the ticker text. As the first step, we detect the ticker text present in the news video frame. This step is known as text localization. We identify the groups of video frames where ticker text is available and mark the boundaries of the text (highlighted by yellow colored boxes in the figure). The knowledge about the production rules of a channel helps us selecting the ticker text segments relevant to the current news story. In the next step, we extract these image segments from the identified groups of frames. Further, we identify the image segments containing the same text and combine the information in these images to obtain a high-resolution image using image super-resolution technique. We binarize this image and apply touching character segmentation as an image cleaning step.

These techniques help improve the recognition rate of OCR. Finally, the text images are processed by OCR software and desired keywords are identified from the resultant text using the multilingual keyword list. The following subsections give detailed explanation of these steps.

4.3.1. Text Localization in News Video Frames. The text recognition in a video sequence involves detection of the text regions in a frame, recognizing the textual content and tracking the ticker news video in successive frames. Homogeneous color and sharp edges are the key features of texts in an image or video sequence. Peng and Xiao [37] have proposed color-based clustering accompanied with sharp edge features for detection of text regions. Sun et al. [38] propose a text extraction by color clustering and connected component analysis followed by text recognition using a novel stroke verification algorithm to build a binary text line image after removing the noncharacter strokes. A multiscale wavelet-based texture feature followed by SVM classifier is used for text detection in image and video frames [39]. An automatic detection, localization and tracking of text regions in MPEG videos are proposed in [40]. The text detection is based on wavelet transform and modified k-means classifier. Retrieval of sports video databases using SIFT feature-based trademark matching is proposed by [41]. The SIFT based approach is suitable for offline processing in video database but is not a feasible option in real time MPEG video streaming.

The classifier-based approaches have a limitation that if the test data pattern varies from the data used in learning, robustness of the system gets reduced. In the proposed method we have used the hybrid approach where we localize the candidate text regions initially using the compressed domain data processing and process the region of interest in pixel domain to mark the text region. This approach has a benefit over other in two aspects namely robustness and time complexity.

Our proposed methodology is based on the following assumptions.

(1) Text regions have significant contrast with background color.

(2) News ticker text is horizontally aligned.

(3) The components representing texts region has strong vertical edges.

As stated above we have used compressed domain features and time domain features to localize the text regions. The steps involved are as follows.

(1) Computation of Text Regions Using Compressed Domain Features. In order to determine the text regions in the compressed domain, we first compute the horizontal and vertical energies at the sub block (4×4) level and mark the subblocks as text or nontext assuming that the text regions generally possess high vertical and horizontal energies. To mark the high energy regions we first divide the entire video frame into small blocks each of size 4×4 pixels. Next, we apply integer transformation on each of the blocks. We have selected Integer transformation in place of DCT to avoid the problem of rounding off and complexity of floating point operation. We compute the horizontal energy of the subblock by summing the absolute amplitudes of the horizontal harmonics (C_{U0}) and the vertical energy of the subblock by summing the absolute amplitudes of the vertical harmonics (C_{0V}). Then we compute the average horizontal text energy ($E_{\text{Avg\_Hor}}$) and the average vertical text energy ($E_{\text{Avg\_Ver}}$) for each row of subblocks. Lastly we mark candidate rows if both ($E_{\text{Avg\_Hor}}$) and ($E_{\text{Avg\_Ver}}$) exceed threshold value α, where α is calculated as $\mu_E + a\sigma_E$ where "a" is empirically selected by analyzing the mean and standard deviation of energy values observed over a large number of Indian broadcast channels.

(2) Filter Out the Low Contrast Components in Pixel Domain. Human eye is more sensitive in high-contrast regions compared to the low-contrast regions. Therefore, it is reasonable to assume that the ticker-text regions in a video are created with significant contrast with background colour. This assumption is found to be valid in most of the Indian channels. At the next step of processing, we remove all low-contrast components from the candidate text regions identified in the previous step. Finally, the candidate text segments are binarized using Otsu's method [42].

(3) Morphological Closing. The text components sometimes get disjointed depending on the foreground and background contrast and the video quality. Moreover, non textual regions appear as noise in the candidate text regions. A morphological closing operation is applied with rectangular structural elements with dimension of 3×5 to eliminate the noise and indentify continuous text segments.

(4) Confirmation of the Text Regions. Initially we run a connected component analysis for all pixels after morphological closing to split the candidate pixels into n number of connected components. Then we eliminate all the connected components which do not satisfy shape features like size and compactness (Compactness is defined as the number of pixel per unit area.).

Then we compute the mode for x and y coordinates of top left and bottom right coordinates of the remaining components. We compute the threshold as the mode of the difference between the median and the position of all the pixels.

The components, for which the difference of its position and the median of all the positions are less than the threshold, are selected as the candidate texts. We have used Euclidean distance as a distance measure.

(5) Confirmation of the Text Regions Using Temporal Information. At this stage, the text segments have been largely identified. But, some spurious segments are still there. We use heuristics to remove spurious segments. Human vision psychology suggests that eyes cannot detect any event within 1/10th of a second. Understanding of video content requires

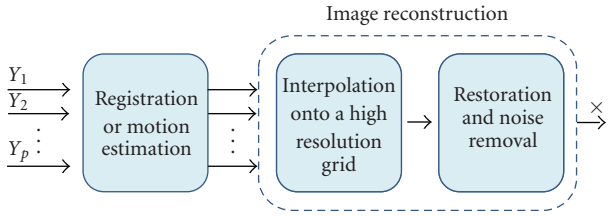


Figure 5: Stages of image super resolution.

at least 1/3rd of a second, that is, 10 frames in a video with frame-rate of 30 FPS. Thus, any information on video meant for human comprehension must persist for this minimum duration. It is also observed that the noise detected as text does not generally persist for significant duration of time. Thus, we eliminate any detected text regions that persists for less than 10 frames. At the end of this phase, we get a set of groups of frames (GoF) containing ticker text. The information together with the coordinates of the bounding boxes for the ticker text are recorded at the end of this stage of processing.

4.3.2. Image Super Resolution and Image Cleaning. The GoF containing ticker text regions cannot be directly used with OCR software because the size of the text is still too small and lacks clarity. Moreover, the characters in the running text are often connected and need to be separated from each other for reliable OCR output.

To accomplish this task we interpolate these images to a higher resolution by using Image Super Resolution (SR) techniques [43, 44] and subsequently perform touching character segmentation as image cleaning process in order to address these problems. The processing steps are given below.

(1) Image Super Resolution (SR). Figure 5 shows different stages of a multiframe image SR system to produce an image with a higher resolution (X) from a set of images $(Y_1, Y_2, \ldots, Y_p)$ with lower resolution. We have used SR technique presented in [45], where information from a set of multiple low resolution images is used to create a higher resolution image. Hence it becomes extremely important to find images with the same ticker text. We perform pixel subtraction of both the images in a single pass. We now count the number of nonblack pixels by using intensity scheme $(R, G, B) < (25, 25, 25)$. We then normalize this count by dividing it by total number of pixels and record this value. If this value exceeds statistically determined threshold "β", we declare the images as nonidentical otherwise we place both the images in the same set. As shown in Figure 5, multiple low resolution images are fed to an image registration module which employs frequency domain approach and estimates the planar motion which is described as function of three parameters: horizontal shift (Δx), vertical shift (Δy), and the planar rotation angle (Φ). In Image Reconstruction stage, the samples of the different low-resolution images are first expressed in the coordinate frame of the reference image. Then, based on these known samples, the image values are interpolated on a regular high-resolution grid. For this purpose bicubic interpolation is used because of its low computational complexity and good results.

Transcription	śivō rakṣatu gīrvāṇabhāṣārasāsvādatatparān
Bengālī	শিবো রক্ষতু গীর্বাণভাষারসাস্বাদতৎপরান্
Devanāgarī	शिवो रक्षतु गीर्वाणभाषारसास्वादतत्परान्
Gujarātī	શિવો રક્ષતુ ગીર્વાણભાષારસાસ્વાદતત્પરાન્
Gurmukhī	ਸ਼ਿਵੋ ਰਕ੍ਸ਼ਤੁ ਗੀਰ੍ਵਾਣਭਾਸ਼ਾਰਸਾਸ੍ਵਾਦਤਤ੍ਪਰਾਨ੍
Oṛiyā	ଶିବଃ। ରକ୍ଷତୁ ଗୀର୍ବାଣଭାଷାରସାସ୍ୱାଦତତ୍ପରାନ୍
Tamiḻ	ஷிவோ ரக்ஷது கீர்வாணபாஷாரஸாஸ்வாததத்பராந்
Tĕlugu	శివో రక్షతు గీర్వాణభాషారసాస్వాదతత్పరాన్
Kannaḍa	ಶಿವೋ ರಕ್ಷತು ಗೀರ್ವಾಣಭಾಷಾರಸಾಸ್ವಾದತತ್ಪರಾನ್
Malayāḷam	ശിവോ രക്ഷതു ഗീർവാണഭാഷാരസാസ്വാദതത്പരാൻ
Grantha	ശിവൊ രക്ഷതാ ഗീർവ്വാണഭാഷാരസാസ്വാദതത്പരാൻ

Figure 6: Samples of a few major Indian scripts (Source: http://www.myscribeweb.com/Phrase_sanskrit.png.).

(2) Touching Character Segmentation. We binarize the high-resolution image by Otsu's method [42] containing ticker text. We generally find some of the text characters touching each other in the binarized image because of noise that can adversely affect the performance of the OCR. Hence, we follow up this step with segmentation of touching characters for improved character recognition.

For Touching Character Segmentation, we initially find the average character width for all the characters in the region of interest (ROI) by $\mu_{\mathrm{WC}} = (1/n)\sum_{i=1}^{n} \mathrm{WC}_i$ where "n" is the number of characters in the ROI and "WC_i" is the character width of the ith component. We then compute the threshold for character length and the components with a width greater than that threshold are marked as candidate touching characters. The threshold for character length is computed as $(T_{\mathrm{WC}} = \mu_{\mathrm{WC}} + 3 * \sigma_{\mathrm{WC}})$. We have used $(3 * \sigma_{\mathrm{WC}})$ to ensure higher recall. For our purpose threshold is nearly 64. Then we split them into number of possible touches. The number of touches in a candidate component is computed as the ceiling value of the ratio between actual width and the threshold value, that is, $n_i = [\mathrm{WC}_i/T_{\mathrm{WC}}] + 1$. In some Indian languages (like Bangla and Hindi), the characters in a word are connected by a unique line called *Shirorekha,* also called the "head line". Touching character segmentation for such languages is preceded by the removal of *shirorekha*, which makes character segmentation more efficient.

4.3.3. OCR and Dictionary-Based Correction. The higher quality image obtained as a result of last stage of processing is processed with OCR software to create a transcript of the ticker text in the native language of the channel. The transcript is generally error-prone and we use the multilingual keyword list in conjunction with an approximate string matching algorithm for robust recognition of the desired keywords in the transcript. There are telecasts in English, Hindi (the national language), and several regional languages in India. Many of the languages use their own

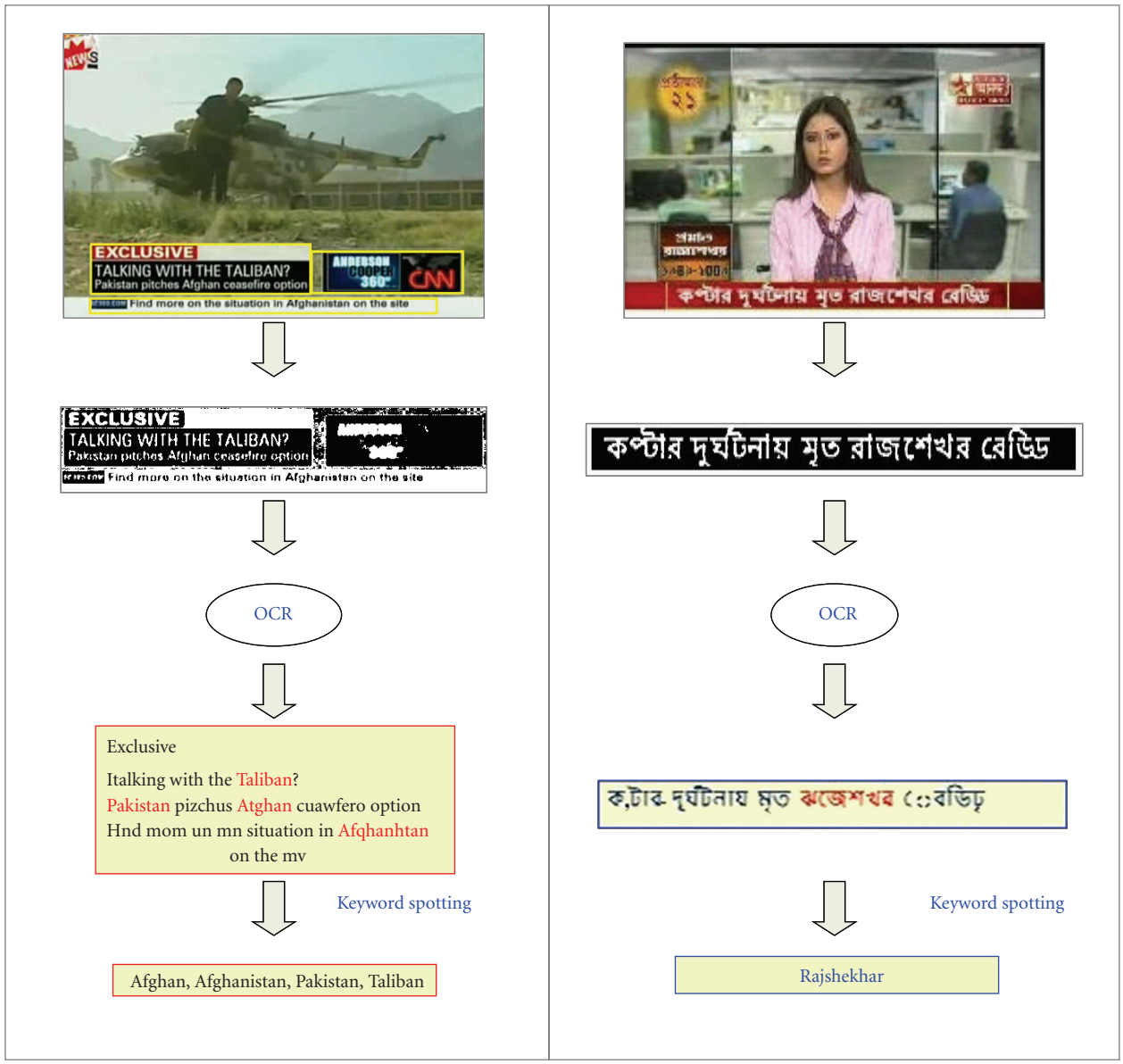


Figure 7: Keyword Identification from English and Bangla news channel.

scripts. Samples of a few major Indian scripts are shown in Figure 6.

The development of OCR in many of these Indian languages is more complex than English and other European languages. Unlike these languages, where the number of characters to be recognized is less than 100, Indian languages have several hundreds of distinct characters. Nonuniformity in spacing of characters and connection of the characters in a word by *Shirorekha* in some of the languages are other issues. There has been significant progress in OCR research in several Indian languages. For example, in Hasnat et al. [46], Lehal [1], and Jawahar et al. [2], word accuracy over 90% has been attained. Still, many of the Indian languages lack a robust OCR and are not amenable to reliable machine processing. For selecting a suitable OCR to work with English and Indian languages, we looked for the highly ranked OCRs identified at The Fourth Annual Test of OCR Accuracy [47] conducted by Information Science Research Institute (ISRI (http://www.isri.unlv.edu/ISRI/)). Tesseract [48] (More information on Tesseract and download packages are available at http://code.google.com/p/tesseract-ocr/.), an open source OCR, finds a special mention because of its reported high-accuracy range (95.31% to 97.53%) for the magazine, newsletter, and business letter test-sets. Besides English, Tesseract can be trained with a customized set of training data and can be used for regional Indian languages. Adaptation of Tesseract for Bangla has been reported in [46]. Thus, we find Tesseract to be a suitable OCR for creating transcripts of English and Indian language ticker text images extracted from the news videos.

Table 1: Results for keyword spotting in speech with master keyword list.

Story id	Instances of keywords present	Keywords found		Retrieval performance		
		True positives	False Positives	Recall (%)	Precision (%)	F-measure (%)
[1]	[2]	[3]	[4]	[5]	[6]	[7]
				[3]/[2] ∗ 100	[3]/([3] + [4]) ∗ 100	2 ∗ [5] ∗ [6]/([5] + [6])
English Channels						
E001	12	2	5	16.67	28.57	21.05
E002	40	10	6	25.00	62.50	35.71
E003	13	2	3	15.38	40.00	22.22
E004	67	8	12	11.94	40.00	18.39
E005	91	6	7	6.59	46.15	11.54
E006	51	7	8	13.73	46.67	21.21
E007	7	1	3	14.29	25.00	18.18
E008	7	1	3	14.29	25.00	18.18
E009	29	10	6	34.48	62.50	44.44
Overall (English)	*317*	*47*	*53*	*14.83*	*47.00*	*22.54*
Bangla Channels						
B001	7	1	0	14.29	100.00	25.00
B002	14	2	5	14.29	28.57	19.05
B003	13	2	1	15.38	66.67	25.00
B004	13	1	7	7.69	12.50	9.52
B005	29	2	7	6.90	22.22	10.53
Overall (Bangla)	*76*	*8*	*20*	*10.53*	*28.57*	*15.38*
Overall	**393**	**55**	**73**	**13.99**	**42.97**	**21.11**

Table 2: Results for keyword spotting in speech with constrained keyword list.

Story id	Instances of keywords present	Keywords found		Retrieval Performance		
		True positives	False Positives	Recall (%)	Precision (%)	F-measure (%)
[1]	[2]	[3]	[4]	[5]	[6]	[7]
				[3]/[2] ∗ 100	[3]/([3] + [4]) ∗ 100	2 ∗ [5] ∗ [6]/([5] + [6])
English Channels						
E001	12	5	4	41.67	55.56	47.62
E002	40	15	3	37.50	83.33	51.72
E003	13	4	1	30.77	80.00	44.44
E004	67	17	6	25.37	73.91	37.78
E005	91	14	8	15.38	63.64	24.78
E006	51	12	5	23.53	70.59	35.29
E007	7	1	0	14.29	100.00	25.00
E008	7	1	0	14.29	100.00	25.00
E009	29	12	4	41.38	75.00	53.33
Overall (English)	*317*	*81*	*31*	*25.55*	*72.32*	*37.76*
Bangla Channels						
B001	7	3	0	42.86	100.00	60.00
B002	14	3	1	21.43	75.00	33.33
B003	13	4	1	30.77	80.00	44.44
B004	13	1	2	7.69	33.33	12.50
B005	29	8	3	27.59	72.73	40.00
Overall (Bangla)	*76*	*19*	*7*	*25.00*	*73.08*	*37.25*
Overall	**393**	**100**	**38**	**25.45**	**72.46**	**37.66**

Table 3: Results for keyword spotting in ticker text with master keyword list.

Story id	No. of distinct ticker texts	Total instances of keywords present	Keywords found			
			On raw frame	On localized text region	After image super-resolution	After dictionary based correction
[1]	[2]	[3]	[4]	[5]	[6]	[7]
English Channels						
E001	5	41	17	19	24	29
E002	4	26	8	9	16	20
E003	4	23	9	10	13	16
E004	6	40	18	19	25	31
E005	4	31	10	13	17	22
E006	7	46	21	23	28	34
E007	4	21	8	9	12	17
E008	1	1	1	1	1	1
E009	5	19	9	9	11	14
Subtotal—English	40	248	101	112	147	184
Retrieval performance—English (%)			*40.73*	*45.16*	*59.27*	*74.19*
Bangla Channels						
B001	3	7	0	0	2	4
B002	3	7	1	1	2	4
B003	5	9	3	3	6	7
B004	3	6	1	1	2	3
B005	5	11	4	4	5	7
Subtotal—Bangla	19	40	9	9	17	25
Retrieval performance—Bangla (%)			*22.5*	*22.5*	*42.5*	*62.5*
Overall retrieval performance (%)			**38.19**	**42.01**	**56.94**	**72.57**

Despite preprocessing of the text images and high accuracy of Tesseract, the output of the OCR phase contains some errors because of poor quality of the original TV transmission. While it is difficult to improve the OCR accuracy, reliable identification of a finite set of keywords is possible with a dictionary-based correction mechanism. We calculate a weighted Levenshtein distance [49] between every word in the transcripts with the words in corresponding language in the multilingual keyword list and recognize the word if the distance is less than a certain threshold "β". The weights in computing the Levenshtein distance is based on visual similarity of the characters in an alphabet, for example, comparison of "l" (small L) and "1" (numeric one) has a lower weight than two other characters, say "a" and "b". We also put a higher weight for the first and the last letters in a word, considering that OCR has a lower error-rate for them because of the spatial separation (on one side) of these characters. Figure 7 shows examples of transcription and keyword identification from news channels in English and Bangla. We map the Bangla keywords to their English (or any other language) equivalents for indexing using the multilingual keyword file.

5. Experimental Results and Illustrative Examples

We have tested the performance of keyword-based indexing with a number of news stories recorded from different Indian channels in English and in Bangla, which is one of the major Indian languages. The news stories chosen pertained to two themes of national controversy, one involving the comments from a popular cricketer and the other involving a visa-related scam. These stories had been recorded over two consecutive dates. Each of the stories is between 20 seconds and 4 minutes in duration. RSS feeds from "Headlines India" (http://www.headlinesindia.com/) on the same dates have been used to create a master keyword-file with 137 English keywords and their Bangla equivalents. In order to test the improvement in accuracy with restricted domain-specific keyword set, we created a keyword file collected from "India news" category, to which the two stories belonged to. This restricted keyword-file contained 16 English keywords and their Bangla equivalents. The restricted keyword set formed was a subset of the master keyword set.

Sections 5.1 and 5.2 present performance of audio and visual keyword extraction, respectively. Section 5.3 present

Table 4: Results for keyword spotting in ticker text with constrained keyword list.

Story id	No. of distinct ticker texts	Total instances of keywords present	Keywords found			
			On raw frame	On localized text region	After image super-resolution	After dictionary-based correction
[1]	[2]	[3]	[4]	[5]	[6]	[7]
English Channels						
E001	5	36	15	17	22	27
E002	4	23	6	7	14	18
E003	4	23	9	10	13	16
E004	6	35	17	19	24	28
E005	4	31	10	13	17	22
E006	7	39	19	21	25	31
E007	4	18	7	8	11	16
E008	1	1	1	1	1	1
E009	5	16	7	7	9	12
Subtotal—English	40	222	91	103	136	171
Retrieval performance—English (%)			*40.99*	*46.40*	*61.26*	*77.03*
Bangla Channels						
B001	3	6	0	0	2	4
B002	3	6	1	1	2	4
B003	5	7	3	3	5	6
B004	3	4	1	1	2	3
B005	5	11	4	4	5	7
Subtotal—Bangla	19	34	9	9	16	24
Retrieval performance—Bangla (%)			*26.47*	*26.47*	*47.06*	*70.59*
Overall retrieval performance (%)			**39.06**	**43.75**	**59.38**	**76.17**

the overall indexing performance on combining audio and visual cues. Section 5.4 presents a few illustrative examples that explain the results.

5.1. Keyword Spotting in Speech. Table 1 presents the results for keyword spotting in speech in the same set of news-stories observed with the master list of keywords. Column [2] represents the number of instances when any of the keywords occurred in the speech. We call keyword spotting to be successful, when a keyword is correctly identified in the time neighborhood (within a +/− 15 ms window) of the actual utterance. Column [3] indicates the number of such keywords for each news story. Column [4] indicates when a keyword is mistakenly identified, though it was actually not uttered at that point of time. We compute the retrieval performances recall, precision and F-measure (Harmonic mean of precision and recall) in columns [5]–[7].

We note that the overall retrieval performance is quite poor, more so for Bangla. It is not surprising because we have used a Microsoft speech engine that is trained for American English. The English channels experimented with were *Indian* channels and the accent of the narrators were quite distinct. We performed the same experiments with the constrained set of keywords. Table 2 presents the results in detail. We note that both recall and precision has significantly improved with the constrained set of keywords, which were primarily proper nouns. The retrieval performance for Bangla is now comparable to that of English. This justifies the use of a dynamically created keyword list for keyword spotting, which is a key contribution in this paper. We note that the precision is quite high (72%), implying that the false positives are low. However, the recall is still pretty low (25%). We will show how we have exploited redundancy to achieve a reliable indexing despite poor recall at this stage.

5.2. Keyword Spotting in Ticker Text. Table 3 depicts a summary of results for ticker text extraction from the English and Bangla Channels tested with master keyword list. Each of the news stories is identified by a unique id in column [1]. Column [2] presents the number of distinct ticker text frames detected in the story. Column [3] indicates the total instances of keywords built from the master keyword list actually present in the ticker text accompanying the story. Columns [4]–[6] show the number of keywords correctly detected when the full-frame, the localized text region and the super-resolution image (of localized text region) are subjected to OCR. Column [7] depicts the number of keywords correctly identified after dictionary-based correction is applied over the OCR result from the super-resolution image of localized text region. We note that the overall accuracy of keyword detection progressively increases from 38.2% to 72.6% through these stages of processing. In Table 3, retrieval performance refers to the recall value. We have observed very few false positives

Table 5: Indexing performance for audio, visual and combined channels.

	Audio			Visual			Combined		
Story id	No. of distinct keywords	Keywords correctly identified	Indexing Performance IP_a (%)	No. of distinct keywords	Keywords correctly identified	Indexing Performance IP_v (%)	No. of distinct keywords	Keywords correctly identified	Indexing Performance IP_o (%)
[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
	$\lvert K_a\rvert$	$\lvert K_a\rvert$	[3]/[2] $*$ 100	$\lvert K_v\rvert$	$\lvert k_v\rvert$	[6]/[5] $*$ 100	$\lvert K_o\rvert$	$\lvert K_o\rvert$	[9]/[8] $*$ 100
English channels									
E001	8	5	62.50	13	9	69.23	13	11	84.62
E002	10	7	70.00	9	7	77.78	14	12	85.71
E003	7	5	71.43	10	8	80.00	11	9	81.82
E004	12	9	75.00	13	10	76.92	17	15	88.24
E005	21	12	57.14	11	8	72.73	21	18	85.71
E006	13	9	69.23	15	11	73.33	16	14	87.50
E007	5	2	40.00	10	9	90.00	14	13	92.86
E008	5	2	40.00	1	1	100.00	5	3	60.00
E009	12	9	75.00	9	9	100.00	15	14	93.33
Overall (English)	*93*	*60*	*64.52*	*91*	*72*	*79.12*	*126*	*109*	*86.51*
Bangla channels									
B001	3	2	66.67	4	2	50.00	5	5	100.00
B002	5	4	80.00	4	2	50.00	9	7	77.78
B003	7	4	57.14	6	5	83.33	9	8	88.89
B004	6	3	50.00	3	3	100.00	7	5	71.43
B005	9	6	66.67	5	4	80.00	10	8	80.00
Overall (Bangla)	*30*	*19*	*63.33*	*22*	*16*	*72.73*	*40*	*33*	*82.50*
Overall	**123**	**79**	**64.23**	**113**	**88**	**77.88**	**166**	**143**	**86.14**

(<1%), that is, a keyword mistakenly identified though it is actually not there in the text, and hence we do not present precision in the table. We also observe that the average accuracy for detecting Bangla text with OCR is significantly poor compared to that of the English text, which can be attributed to the OCR performance and quality of visuals, but there is significant improvement after dictionary-based correction.

Similar to audio keyword spotting we performed the same experiments with the constrained set of keywords. Table 4 presents the results in details. We found that by using constrained keywords list the results at every stage have improved, though not as significantly as in the case of speech.

5.3. Improving Indexing Performance by Exploiting Redundancy. While, we have presented the retrieval performance for audio and visual keyword recognition task in the previous sections, the goal of the system is to index the news-stories with appropriate keywords. We define the indexing performance of the system as

$$\text{IP} = \frac{|k|}{|K|} \times 100, \tag{1}$$

where k is the set of distinct keywords correctly identified (and used for indexing the story) and K is the set of distinct keywords present in the story.

The indexing performance is improved by exploiting redundancy in occurrence of keywords in audio-visual forms. In particular, we exploit two forms of redundancy.

(a) The same keywords are uttered several times in a story or appear several times on ticker text. A keyword missed out in one instance is often detected in another instance providing better indexing performance

(b) The same keyword may appear in both audio and visual forms. A keyword often missed in the speech is often detected in visuals and vice-versa. This adds to indexing performance too.

Let K_a and K_v denote the set of distinct keywords actually occurring in the speech and the visuals, respectively, in a news story. Then, $K_o = K_a \cup K_v$ represents the set of keywords appearing in the news-story. Similarly, let k_a and k_v represent the set of distinct keywords detected in the speech and visuals respectively. Then, $k_o = k_a \cup k_v$ represents the set of keywords detected in the news-story. The audio, visual, and overall indexing performance (IP_a, IP_v, and IP_o, resp.)

Stage	Image	OCR output	Keywords spotted
English news story (E004)			
Full frame (binarized)		The comment that star ED the controver Y …galnxn l:nursh¤•1o:Sa•:Mn Tendulkar has man bowled J al uc ua breaking Snlumn Khurmumd: Smzmn 'fuvwdulknr "('|[sAE*g5 Mw; Ima bcwlnd Thuckumy on thm Hamm N; gw :§:::,.·:, ?::~r gx: eé §§§ §¤§¥1s§ ¥	Tendulkar(1)
After text localization		Salman Knumhaco: Sachin Tendulkar has man bowled Bal Tbuckuav	Salman Sachin Tendulkar Bal (4)
After image cleaning and super resolution		Salman Knurshaadz Sachin Tendulkar nas clean howled Bal Thackeray	Salman Sachin Tendulkar Bal Thackeray (5)
After dictionary based correction	----	----	Salman Khursheed Sachin TendulkarBal Thackeray (6)
Bangla news story (B004)			
Fullframe (binarized)		ৈড় য্ \\\ ডুটুষ্ছু ডু-:ডু! জিঞো ঢুঘা ডুডু-- র ।.!ঢুরূ ৈড় ঢু হু- ছু */ ড়ী ৈড়খীয়ঁড়ী 6ঁ ।ো৾ড়-ৈড়. খ ।ঢা।ো।ো।ো।ো।।।ো, ঢ়ানূ -ঁ ৈড়। ঁ ।ঢুঢুড়ি ।ে। ।মুড়ী ।মুঢু৾ড় 44 সিuড়িয় 2 নs র বূ৾ত তৃ।ো৾মূ-ল-সিপিe সঘর্শ ঞী! 'শৃfk ।ঢু খ্।ডু ক্কেলেলেল-ে৾গঞা ঢুছু	None (0)
After text localization		সিu ড়িয় 2 নসুর বূ৾ত তৃ।ো৾মূ-ল-সিপিe ম সংঘর্ন	সিপিe ম (1)
After image cleaning and super resolution		সিu ড়ির 2 নs র- বূ৾ক ডুণসুল-সিপিe ম সংঘর্ন	সিপিe ম ডুণসুল (2)
After dictionary based correction	----	----	সিপিe ম ডুণসুল সংঘর্ষ (3)

Figure 8: OCR outputs at different stages of English and Bangla ticker text processing.

can be measured as

$$\mathrm{IP}_a = \frac{|k_a|}{|k_a|} \times 100, \qquad \mathrm{IP}_v = \frac{|k_v|}{|k_v|} \times 100,$$
$$\mathrm{IP}_o = \frac{|k_o|}{|k_o|} \times 100 \equiv \frac{|k_a \cup k_v|}{|k_a \cup k_v|} \times 100. \tag{2}$$

Table 5 depicts the indexing performance of the audio, the visual and the overall system, with the constrained keyword list. Note that the indexing performances of audio and visual channels, both English and Bangla, are significantly higher than the respective recall values. This is because of the redundancy of occurrence of keywords

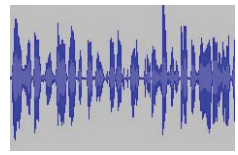
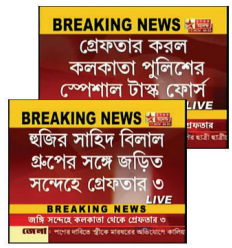
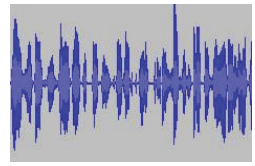
	Keyword spotted in ticker text	Keywords spotted in speech	Combined keyword list
[1]	[2]	[3]	[4]
English news story (E002)			
	Thackeray Sachin Sena Salman sports politics Milkha (7)	Kiran More* Sachin Tendulkar india politics Singh sports (9)	Thackeray Sachin Sena Salman sports politics Milkha Singh Kiran More Tendulkar india (12)
Bangala news story (E005)			
	গিরফ্তার, পুলিস, কলকাতা, হুজি (4)	লs র, গিরফ্তার, কলকাতা, লs র-e-তা বা, বাংলাদেশ, বিতর্ক (6)	গিরফ্তার পুলিস কলকাতা হুজি লs র লs র-e-তা বা বাংলাদেশ বিতর্ক (8)

Figure 9: Combining audio and visual keywords for indexing. *Kiran More: More (pronounced Moré) is a proper noun and not the English word.

in those individual channels. Finally, the overall indexing performance for the stories is greater than the indexing performances of individual audio/visual channels. This is because of the redundancy of keywords across audio and visual channels.

5.4. Illustrative Examples. This section provides some illustrative examples that explain the results in the previous sections. Figure 8 shows the OCR outputs at different stages of processing for examples of English and Bangla ticker text, taken from the stories E004 and B004, respectively. It illustrates the gradual improvement in results through the different stages of image processing and dictionary-based correction.

Figure 9 illustrates improvement in indexing performance by combining audio-visual cues, with an English and a Bangla example. Columns [2] and [3] in the figure show the correctly identified keywords from the ticker text and from speech, respectively. Column [4] depicts the combined keyword list that is used for indexing the story. The combined keyword list is derived as a union of keywords spotted in ticker text and in speech. In these examples, we observe that keywords not detected in speech are often detected in visuals and *vice-versa*. Thus, combining keywords detected in audio and visual forms leads to better indexing performance.

5.5. Comparison. While comparing the system performance, we keep in view the unreliability of the language tools for processing Indian transmission. For example, we have observed the average recall and precision values for keyword spotting in speech to be approximately 15% and 47%, respectively for English (see Table 1), as against typical values of 73% and 85%, respectively in [36]. We also observe that use of a constrained keyword list improves the average recall and precision values to 26% and 72%, respectively (see Table 2), which is still significantly below the reported figures. For keyword detection in ticker text, we have achieved an average recall of 59% (see Table 3) without dictionary-based correction; as compared to 70% reported in [50]. With dictionary-based correction, our recall improves to 67% (see Table 4), which is a reasonable achievement considering complexity of Indian Language alphabets.

An experiment to combine text from speech and visual has been reported in [51]. The authors report recall values for speech recognition and Video OCR as 13% and 6%, respectively. While speech recognition accuracy is comparable to ours, we find the poor OCR results surprising. The authors report a recall of 21% after combining audio and video and dictionary based postprocessing. We have achieved an indexing efficiency of 86%. Though the figures do not directly compare, our system seems to have achieved a much higher performance.

6. Conclusion

We have proposed an architectural framework for automated monitoring of multilingual news video in this paper. The basic idea behind our framework is to combine audio and visual modes to discover the keywords that characterize a particular news-story. Our primary contribution in this paper has been reliable indexing of Indian news telecasts with significant keywords despite inaccuracies of the language tools in processing noisy video channels and deficiencies of language technologies for many Indian Languages. The main contributing factor towards the reliable indexing has been selection of a few domain-specific keywords, in contrast to a complete transcription. Use of several preprocessing and postprocessing stages with the basic language tools has also added to the reliability of results. Moreover, use of RSS feeds to derive the keywords automatically results in contemporariness of the system, which could otherwise be a major operational issue. The conversion of English keywords, which are either proper or common nouns, to their Indian Language equivalents helps indexing non-English transmission with English (or any Indian Language) keywords. The complete end to end solution is made possible by integrating or enhancing available techniques in addition to proposing several techniques that make multilingual, multichannel news broadcast monitoring feasible. The experimental results establish the correctness of the system.

While we have so far experimented with English and one of the Indian languages, namely Bangla, we need to extend the solution to other Indian Languages by integrating appropriate language tools, which are being researched elsewhere in the country. Moreover, India is a large country with twenty-two officially recognized languages and many more "unofficial" languages and dialects. Language tools do not exist and are unlikely to be available in foreseeable future for many of these languages. We propose to direct our future work towards classification of news stories telecast in such languages based on their audio-visual similarity with stories in some reference channels (e.g., some channels in English), which can be indexed using the language technologies.

References

[1] G. S. Lehal, "Optical character recognition of Gurumukhi script using multiple classifiers," in *Proceedings of the International Workshop on Multilingual (OCR '09)*, Barcelona, Spain, July 2009.

[2] C. V. Jawahar, M. N. S. S. K. P. Kumar, and S. S. R. Kiran, "A bilingual OCR for Hindi-Telugu documents and its applications," in *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR '03)*, vol. 1, p. 408, 2003.

[3] E. Hassan, S. Chaudhury, and M. Gopal, "Shape descriptor based document image indexing and symbol recognition," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2009.

[4] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 444–457, 2009.

[5] S. K. Parui, K. Guin, U. Bhattacharya, and B. B. Chaudhuri, "Online handwritten Bangla character recognition using HMM," in *Proceedings of the International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, 2008.

[6] S. Eickeler and S. Mueller, "Content-based video indexing of TV broadcast news using hidden Markov models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '99)*, vol. 6, pp. 2997–3000, March 1999.

[7] J. R. Smith, M. Campbell, M. Naphade, A. Natsev, and J. Tesic, "Learning and classification of semantic concepts in broadcast video," in *Proceedings of the International Conference of Intelligence Analysis*, 2005.

[8] J.-L. Gauvain, L. Lamel, and G. Adda, "Transcribing broadcast news for audio and video indexing," *Communications of the ACM*, vol. 43, no. 2, pp. 64–70, 2000.

[9] H. Meinedo and J. Neto, "Detection of acoustic patterns in broadcast news using neural networks," *Acustica*, 2004.

[10] C.-M. Kuo, C.-P. Chao, W.-H. Chang, and J.-L. Shen, "Broadcast video logo detection and removing," in *Proceedings of the 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '08)*, pp. 837–840, Harbin, China, August 2008.

[11] D. A. Sadlier, S. Marlow, N. Connor, and N. Murphy, "Automatic TV advertisement detection from MPEG bit stream," *Pattern Recognition*, vol. 35, no. 12, pp. 2719–2726, 2002.

[12] T.-Y. Liu, T. Qin, and H.-J. Zhang, "Time-constraint boost for TV commercials detection," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 3, pp. 1617–1620, October 2004.

[13] X.-S. Hua, L. Lu, and H.-J. Zhang, "Robust learning-based TV commercial detection," in *Proceedings of the ACM International Conference on Multimedia and Expo (ICME '05)*, pp. 149–152, Amsterdam, The Netherlands, July 2005.

[14] K. Ng and V. W. Zue, "Phonetic recognition for spoken document retrieval," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, vol. 1, pp. 325–328, 1998.

[15] M. Laxminarayana and S. Kopparapu, "Semi-automatic generation of pronunciation dictionary for proper names: an optimization approach," in *Proceedings of the 6th International Conference on Natural Language Processing (ICON '08)*, pp. 118–126, CDAC, Pune, India, December 2008.

[16] J. Makhoul, F. Kubala, T. Leek, et al., "Speech and language technologies for audio indexing and retrieval," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1338–1352, 2000.

[17] S. Renals, D. Abberley, D. Kirby, and T. Robinson, "Indexing and retrieval of broadcast news," *Speech Communication*, vol. 32, no. 1, pp. 5–20, 2000.

[18] T. Chua, S. Y. Neo, K. Li, et al., "TRECVID 2004 search and feature extraction tasks by NUS PRIS," in *NIST TRECVID-2004*, 2004.

[19] T. Chua, S.-F. Chang, L. Chaisorn, and W. Hsu, "Story boundary detection in large broadcast news video archives: techniques, experience and trends," in *Proceedings of the 12th ACM International Conference on Multimedia (MM '04)*, pp. 656–659, 2004.

[20] A. Rosenberg and J. Hirschberg, "Story segmentation of broadcast news in English, Mandarin and Arabic," in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, June 2006.

[21] M. Franz and J.-M. Xu, "Story segmentation of broadcast news in Arabic, Chinese and English using multi-window features," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07)*, pp. 703–704, 2007.

[22] M. A. Hearst, "TextTiling: segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[23] X. Gao and X. Tang, "Unsupervised video-shot segmentation and model-free anchor-person detection for news video parsing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 9, pp. 765–776, 2002.

[24] S.-F. Chang, R. Manmatha, and T.-S. Chua, "Combining text and audio-visual features in video indexing," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '05)*, vol. 5, pp. 1005–1008, 2005.

[25] L. Chaisorn, T.-S. Chua, and C.-H. Lee, "A multi-modal approach to story segmentation for news video," *World Wide Web*, vol. 6, no. 2, pp. 187–208, 2003.

[26] L. Besacier, G. Quénot, S. Ayache, and D. Moraru, "Video story segmentation with multi-modal features: experiments on TRECvid 2003," in *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR '04)*, pp. 221–226, October 2004.

[27] Anonymous, "F1 Score," *Wikipedia—The Free Encyclopedia*, February 2010, http://en.wikipedia.org/wiki/F1_score.

[28] F. Colace, P. Foggia, and G. Percannella, "A probabilistic framework for TV-news stories detection and classification," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 1350–1353, July 2005.

[29] G. Harit, S. Chaudhury, and H. Ghosh, "Using multimedia ontology for generating conceptual annotations and hyperlinks in video collections," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '06)*, pp. 211–217, Hong Kong, December 2006.

[30] Anonymous, "News Ticker," Wikipedia—The Free Encyclopedia, February 2010, http://en.wikipedia.org/wiki/News_ticker.

[31] D. Winer, "RSS 2.0 Specification," Wikipedia—The free Encyclopedia, February 2010, http://cyber.law.harvard.edu/rss/rss.html.

[32] S. Kopparapu, A. Srivastava, and P. V. S. Rao, "Minimal parsing key concept based question answering system," *Human Computer Interaction*, vol. 3, 2007.

[33] P. Gelin and C. J. Wellekens, "Keyword spotting for video soundtrack indexing," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 299–302, May 1996.

[34] Y. Oh, J.-S. Park, and K.-M. Park, "Keyword spotting in broadcast news," in *Global-Network-Oriented Information Electronics (IGNOIE-COE06)*, pp. 208–213, Sendai, Japan, January 2007.

[35] G. Quenot, T. P. Tan, L. V. Bac, S. Ayache, L. Besacier, and P. Mulhem, "Content-based search in multi-lingual audiovisual documents using the international phonetic alphabet," in *Proceedings of the 7th International Workshop on Content-Based Multimedia Indexing (CBMI '09)*, Chania, Greece, June 2009.

[36] D. Dimitriadis, A. Metallinou, I. Konstantinou, G. Goumas, P. Maragos, and N. Koziris, "GRIDNEWS1a distributed automatic Greek broadcast transcription system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09)*, 2009.

[37] J. Yi, Y. Peng, and J. Xiao, "Color-based clustering for text detection and extraction in image," in *Proceedings of the ACM International Multimedia Conference and Exhibition (MM '07)*, pp. 847–850, Augsburg, Germany, Sebtember 2007.

[38] J. Sun, Z. Wang, H. Yu, F. Nishino, Y. Katsuyama, and S. Naoi, "Effective text extraction and recognition for WWW images," in *Proceedings of the ACM Symposium on Document Engineering (DocEng '03)*, pp. 115–117, Grenoble, France, November 2003.

[39] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and robust text detection in images and video frames," *Image and Vision Computing*, vol. 23, no. 6, pp. 565–576, 2005.

[40] J. Gllavata, R. Ewerth, and B. Freisleben, "Tracking text in MPEG videos," *ACM*, 2004.

[41] A. D. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo, "Trademark matching and retrieval in sports video databases," in *Proceedings of the International Workshop on Multimedia Information Retrieval (MIR '07)*, pp. 79–86, Augsburg, Germany, Sebtember 2007.

[42] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[43] R. Y. Tsai and T. S. Huang, "Multiple frame image restoration and registration," in *Advances in Computer Vision and Image Processing*, pp. 317–339, JAI Press, Greenwich, Conn, USA, 1984.

[44] V. H. Patil, D. S. Bormane, and H. K. Patil, "Color super resolution image reconstruction," in *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '07)*, vol. 3, pp. 366–370, 2007.

[45] P. Vandewalle, S. Süsstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to super-resolution," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–14, 2006.

[46] M. A. Hasnat, M. R. Chowdhury, and M. Khan, "Integrating Bangla script recognition support in Tesseract OCR," in *Proceedings of the Conference on Language and Technology*, 2009.

[47] S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The fourth annual test of OCR accuracy," Tech. Rep. 95-04, Information Science Research Institute, University of Nevada, Las Vegas, Nev, USA, April 1995.

[48] R. Smith, "An overview of the Tesseract OCR engine," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, vol. 2, pp. 629–633, September 2007.

[49] M. Gilleland, "Levenshtein Distance, in Three Flavors," February 2010, http://www.merriampark.com/ld.htm.

[50] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 256–268, 2002.

[51] A. G. Hauptmann, R. Jin, and T. D. Ng, "Multi-modal information retrieval from broadcast video using OCR and speech recognition," in *Proceedings of the 2nd ACM International Conference on Digital Libraries*, pp. 160–161, Portland, Ore, USA, 2002.