Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2009, Article ID 496934, 12 pages
doi:10.1155/2009/496934

Research Article

Slow Motion and Zoom in HD Digital Videos Using Fractals

Maurizio Murroni, Cristian Perra, and Daniele D. Giusto

DIEE, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy

Correspondence should be addressed to Cristian Perra, cperra@diee.unica.it

Received 1 March 2009; Accepted 19 October 2009

Recommended by Sandro Scalise

Slow motion replay and spatial zooming are special effects used in digital video rendering. At present, most techniques to perform digital spatial zoom and slow motion are based on interpolation for both enlarging the size of the original pictures and generating additional intermediate frames. Mainly, interpolation is done either by linear or cubic spline functions or by motion estimation/compensation which both can be applied pixel by pixel, or by partitioning frames into blocks. Purpose of this paper is to present an alternative technique combining fractals theory and wavelet decomposition to achieve spatial zoom and slow motion replay of HD digital color video sequences. Fast scene change detection, active scene detection, wavelet subband analysis, and color fractal coding based on Earth Mover's Distance (EMD) measure are used to reduce computational load and to improve visual quality. Experiments show that the proposed scheme achieves better results in terms of overall visual quality compared to the state-of-the-art techniques.

1. Introduction

Today's TV broadcasting industry is rapidly facing new challenges to chase the technological progress if compared to the previous fifty years of its existence. The migration from analog to digital systems which has begun in the early days of the last decade with the satellite broadcasting is almost completed also for its terrestrial counterpart. Furthermore, the DVB family of standards has recently extended its arena with the release, alongside traditional DVB-S, DVB-T, and DVB-C, of the new DVB-H and DVB-SH to cope with mobile applications for handheld terminals, while several new means to service delivering are arising beside traditional terrestrial and satellite systems, such as TV video streaming over IP (IPTV) based either on XDSL or in the forthcoming near future on WiMAX access. A common characteristic of this new access technologies is the ability to provide broadband services allowing High Digital TV (HDTV) to become now a reality. Furthermore, new generation set-top boxes provided with the multiple access feature are able to decode heterogeneous TV input signals (e.g., DVB-T, DVB-S, IPTV). Within this framework, regardless the broadband access technology deployed, more and more new common features and services are been developed to enlarge the quality of the video at user side. Video rendering refers to all the techniques able to add flexibility to the end user by modifying somehow the view of a video sequence. With the birth of the new LCD or Plasma screen fully supporting the Full HD technology special effectS like image zoom and resize as well as slow motion are likely to be integrated in the new generation Full HDTV set-top boxes.

Slow motion replay is another special effect used in video rendering. It consists of a presentation of video scenes at rates lower than originals. Already consolidated as a commercial feature for analog video players, today slow motion is ready to be extended to the digital formats. Within an analog framework, given a video sequence at a certain frame rate, the classical slow motion effect is obtained, at the display, by reducing the frame rate to a certain amount, so that a frame is frozen and remains visible for a time proportional to the slow motion factor. On the other hand, analog slow motion is realized in preproduction by means of fast-shuttered cameras able to capture the scene at a frame rate higher than the standard rate (i.e., 25 frame/sec for PAL/SECAM, 30 frame/sec for NTSC). Slow motion is achieved by filming at a speed faster than the standard rate and then projecting

the film at the standard speed. In this case, the slow motion factor achievable is limited to shutter speed and is fixed at the preproduction stage.

In a digital environment, these limits for fast-shuttered cameras can be overcome through processing techniques.

At present, commercial digital video players allow users to browse a video sequence frame by frame, or by chapter selection with prefixed indexes. Slow motion replay is achieved by reducing the frame rate display or keeping it constant [1, 2] and inserting within the sequence additional intermediate frames generated by interpolation. Interpolation can be applied at either at pixel or grouping pixels into blocks. Data replication, linear or cubic spline can be used at first sight. A major drawback of these approaches is that yield to a marked degradation of the video quality which can be noticed in terms of "fading" effect (spline) and "jerky" distortion (data replication), both resulting in low motion quality for the human visual system.

Similar issues arise in image plane if interpolation is used to perform spatial zoom. Block distortion and/or blurring effect can be experienced in the enlarged frames.

In the recent years, motion compensated frame interpolation (MCFI) techniques were proposed to improve performance in case of slow motion. Although these techniques were formerly used to convert frame rate between PAL, NTSC, and HDTV, MCFI methods are also used in video streaming and conferencing applications [3–5]. MCFI idea is to implement motion estimation on the previous and current frame, and then generate the corresponding interpolated frame by averaging pixels in the previous and current frame that are pointed by the half of motion vectors. Motion estimation can be achieved by block-based or pixelwise methods. In general, pixelwise motion estimation can attain more accurate motion fields, but needs a huge amount of computations. Thus, it is often used in off-line MCFI rather than real-time processing. In contrast, block matching algorithms (BMAs) can be efficiently implemented and provide good performance (most MCFI methods are based on BMA). A comparison between pixelwise and block-based motion estimation for MCFI is discussed in [6].

In [7], the joint use of fractal coding and wavelet subband analysis to obtain spatial zoom and slow motion replay of luminance video sequences is proposed to avoid the above mentioned distortion effects.

In digital pictures, fractals are mainly used to achieve data compression by exploiting self-similarity of natural images [8, 9], but their potentials are not limited to compression. The properties of fractal coding allow expanding a multidimensional signal (e.g., image and video) along any of its dimensions. A major weakness of fractal representation is the high computational complexity in searching similarities among blocks using affine transformations; therefore, a "best match" algorithm is very time-consuming for multidimensional data sets.

Several methods have been proposed to speed up fractal coding [10]. A class of proposed solutions is based on wavelet subband analysis [11]. Due to their orthogonal and localization properties, wavelets are well suited (and extensively adopted) for subband data analysis and processing.

Our algorithm exploits these features by performing the fractal coding of each subband with particular attention to the frequency distribution of the coefficients. To further reduce the high computational cost of fractal encoding, we use active scene detection so as to perform fractal coding in high information (moving) areas only. Furthermore to improve overall visual quality, overlapped block coding and postfiltering are used, as suggested in [12] but extended to the three-dimensional case. Experimental results presented in the following show that our approach achieves higher subjective and objective quality, with respect to state-of-the-art techniques.

Conventional fractal coding schemes can easily be extended to color image (video) as represented in multichannels such as Red, Green, and Blue (RGB) components. Thus each channel in color image can be compressed as a grey-level image. Hurtgen, Mols, and Simon proposed a fractal transform coding of color images in [13]. To exploit the spectral redundancy in RGB components, the root mean square (RMS) error measure in gray-scale space is extended to 3-dimensional color space for fractal-based color image coding [14]. Experimental results show that a 1.5 compression ratio improvement can be obtained using vector distortion measure in fractal coding with fixed image partition as compared to separate fractal coding in RGB images. However, RGB space is not perceptually uniform. A system is not uniform if a little perturbation of a value is perceived linearly along the possible variation of that value. This means that a color space is perceptually uniform if a distance from a color a and another color $b = a + \Delta c$ will be perceived as constant independently from a or b. Using a nonperceptually uniform space as RGB has the drawback that the Human Vision System (HVS) will be affected by computer measures for digital video processing, since the distance from RGB value will not be uniform in respect of the HVS. Starting from these considerations, the Commission Internationale d'Eclairage (CIE) defined a uniform color model, called L*a*b* that represents all the color humans being able to resolve. Danciu and Hart [15] presented a comparative study of fractal color image compression in the L*a*b* color space with that of Jacquin's iterated transform technique for 3-dimensional color. It has been shown that the use of uniform color space has yielded compressed images less noticeable color distortion than other methods. In this paper we will propose a novel approach for coding color images based on the joint use of the L*a*b* color space and Earth Mover's Distance (EMD) measure [16]. EMD has been suitably deployed for color image retrieval applications [17]. It is a vector metric that combines spatial and color information to resolve similarities among color images. In this work we implement a fractal coding approach that relies on EMD for finding self-similarities within color images represented in the L*a*b* color space. The proposed approach has achieved better results in terms of objective quality assessment compared a classic codec based on RMS measure.

The present paper is the natural extension of [7] to the case of HD color video sequences and it copes with the further issue distinctive of the fractal coding

of color video scenes. The EMD measure is used during the coding. This measure has proven to be suitable for detecting similarities between color multimedia contents [18]. To reduce the high computational cost of fractal coding, an active scene detector is used, so as to perform full three-dimensional coding only in high information areas (moving areas), whereas static zones are coded using a two-dimensional coder. To further speed up the coding process a wavelet subband analysis is performed whereas postprocessing techniques are used to improve visual quality. In addition, a fast scene change detection algorithm [17] is exploited for determining the optimal temporal window for maximizing the video quality of the zoom and slow motion processing.

The paper is organized as follows. In Section 2, a description of fractal theory applied to color video processing is given. Section 3 details the proposed method. Experimental results are provided in Section 4. Conclusions are finally drawn in Section 5.

2. Fractal Theory Applied to Color Video Processing

The fractal representation of natural signals is possible irrespective of the signal dimensions and can be used in applications concerning voice/sound, images, video sequences, and so on. Fractal representation/coding is based on the Iterated Function System (IFS). The basic idea of IFS approach is to exploit the redundancy given by the self-similarity always contained in natural signals. For instance, a "fractal image" can be seen as a collage composed by copies of parts of an original image that have been transformed through opportune geometric and "massive" transformations (i.e., luminance or contrast shift).

The mathematical foundation of this technique is the General Theory of Contractive Iterated Transformations [8, 9]. Basically, fractal coding of an image consists in building a code τ (i.e., a particular transformation) such that, if μ_{orig} is the original image, then $\mu_{\text{orig}} \approx \tau(\mu_{\text{orig}})$, that is, μ_{orig} is approximately self-transforming under τ. If τ is a contractive transformation, μ_{orig} is approximately the attractor of τ, that is, $\mu_{\text{orig}} \approx \lim_{k\to\infty}\tau^k(\mu_0)$ for the some initial image μ_0. The code τ is built on a partition of the original image. Each block R_i of this partition is called Range Block and is coded independently of the others by a matching (local code τ_i) with another block D_i in the image, called a Domain Block. If R and D are the range and domain block's sizes (in case of squared blocks), respectively, then $D = p \cdot R$ with $p > 1$ scaled factor used for the local self-similarity search.

Classical τ_i transforms are isometries (i.e., rotations, flip, etc.) and massive transform (i.e., contrast scaling and grey shifting). If L is the number of range blocks, the fractal code of the initial image is then $\tau(\mu_{\text{origin}}) = \bigcup_{i=1}^{L} \tau_i$ where $\tau_i : D_i \to R_i$ and $\tau_i = M_i \circ I_i \circ r_{i,p}$ with $M_i(x) = a_i \cdot x + b_i$ an affine operator with a scale a_i and a shift b_i on the luminance pixel, I_i a transformation selected from eight discrete isometries, and $r_{i,p}$ a reduction by a factor p using an averaging. In other words, the fractal encoder has to find, for each range block, a larger domain block that constitutes, after an appropriate transformation, a good approximation of the present range block.

The fractal code for the original image is a collection of so extracted local codes. This approach, implemented by Jacquin [9], gives a representation of an image as composed by copies of parts of itself. The classical fractal decoding stage consists in an iterated process starting from an arbitrary initial image μ_0. In fact, if τ is a contractive transformation, the τ's attractor $\tau^{\infty}(\mu_0)$ gives an approximation of the original image μ_{orig} independently from the initial image. Essentially, the fractal code τ is a collection of linear affine transforms τ_i, and it has no intrinsic size. Hence, we can assume that self-similarities, that is, matching between areas with different sizes in the original image are scale independent. As to this, the decoding process results "resolution independent," that is, at decoding stage the fractal code enables zoom [19]. Practically, this operation consists in increasing, during the decoding stage, the range block's size R, and therefore the domain block's size D (being $D = p \cdot R$). For a zoom by a factor z, the new sizes will be $R' = z \cdot R$ and $D' = z \cdot D$, but all the local codes τ_i and consequently the fractal code τ will be unchanged.

In this work we propose a novel solution to the self-similarity search for IFS fractal coding of HD color video sequences. At first we transform the image mapping of the colors in the uniform L*a*b* space. Then, by means of a clustering process, for each range and domain block, we extract a block *signature* which is a summary of the spatial and color information of the image blocks. To obtain the fractal code as previously described, we compare the range and domain blocks signatures by means of the EMD measure. Here, is the novelty of our approach. In fact, we perform the comparison between summaries of the space and colors information contained within image blocks, in opposition to classic IFS schemes that compare them at pixel level by means of RMS measures. A sketch of the algorithm is shown in Figure 1. In the following we give the details of the proposed scheme.

2.1. Image Blocks Signature Extraction. An image block signature is a set of features extracted by means of a clustering process. Clustering is a technique aiming at partitioning the image block in a set of subregions, (i.e., clusters) formed by pixels gathered according to some distance rule. To each cluster is associated a feature representative of the cluster. Formally, given an image block C of size n, its signature $S(C) = \{(c_j, w_j)\}_{j=1}^{T}$, with T number of clusters, w_j weight, and c_j centroid (i.e., the representative element) of the cluster j. For the clustering process we use the classic k-mean algorithm [20]. We measure the distance among pixels both in the spatial and color domains. As to the spatial domain, for every pixel $\{y_i\}_{i=1}^{n}$ we limit the search area to a circle centered in y_i with radius r. The length of r is computed considering the medium spatial distance between y_i and the initial distribution of centroids. The color distance is also upper bounded by the resolution of the HVS in the uniform

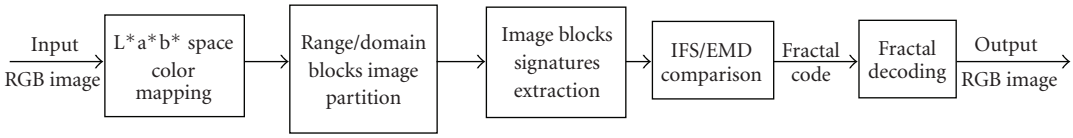

Figure 1: Fractal coding of colour images.

color L*a*b* space ($\mathrm{HVS}_{\mathrm{res}}$), that is, the minimum distance in the L*a*b* color space that allows the HVS discriminating two different colors. Formally, we define the distance between the generic pixel y_i and a centroid c_j as

$$\mathbf{d}\left(y_i, c_j\right) = \sqrt{\mathbf{dis}_s^2\left(y_i, c_j\right) + \mathbf{dis}_c^2\left(y_i, c_j\right)},$$

$$\mathbf{dis}_s\left(y_i, c_j\right) = \frac{\left\|y_i - c_j\right\|_{\mathrm{spatial}}}{r} < 1, \quad r = \frac{1}{T}\sum_{j=1}^{T}\left\|y_i - c_j\right\|_{\mathrm{spatial}},$$

$$\mathbf{dis}_c\left(y_i, c_j\right) = \frac{\left\|y_i - c_j\right\|_{\mathrm{L^*a^*b^*}}}{\mathrm{HVS}_{\mathrm{res}}} < 1, \tag{1}$$

where $\mathbf{dis}_s(y_i, c_j)$ and $\mathbf{dis}_c(y_i, c_j)$ are their normalized Euclidean distances in the spatial domain and in the L*a*b* color space, respectively. It is worth noticing that $\mathbf{d}(\cdot,\cdot)$ is nonnegative, symmetric and satisfies the triangle inequality—thus we really work with a metric space. The clustering process associates y_i to c_j according to

$$\mathbf{d}\left(y_i, c_j\right) = \min_j \mathbf{d}\left(y_i, c_j\right). \tag{2}$$

The initial position of the centroids is chosen to be invariant to the possible affine transformation τ performed by the fractal coding. This assures that, given a block signature $S(C)$ and a transform τ, $\tau[S(C)] = S[\tau(Q)]$ [16].

The number of centroids T is chosen as to satisfy two constraints: maximum uniformity in the distance among centroids and invariance to the geometrical affine transformations (i.e., isometries).

The initial set of 12 centroids for an 8×8 size image block is shown in Figure 2. This positioning is spatially homogeneous while the distance between centroids as well as the distance between pixel and surrounding centroids is essentially constant. This displacement is invariant as to the 8 possible isometries. At the end of the clustering process a signature is assigned to each range and domain block.

2.2. Earth Mover's Distance for IFS. The self-similarity search within the color image is performed by IFS comparing the signatures of the domain and range blocks as defined in Section 2.1. The matching process relies on the Earth mover's distance (EMD). EMD is a useful and extendible metric distance, developed by the Stanford Vision Laboratory (SLV), based on the minimal cost that must be paid to transform one signature into another. The EMD is based

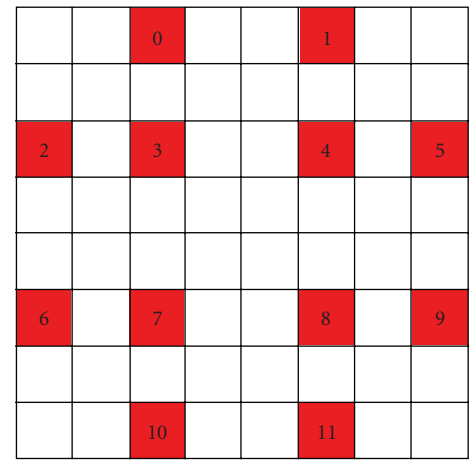

Figure 2: Initial displacement of centroids.

on the *transportation problem* from linear optimization, also known as the Monge-Kantorovich problem [21]. Suppose that several *suppliers*, each with a given amount of goods, are required to supply several *consumers*, each with a given limited capacity. For each supplier-consumer pair, the cost of transporting a single unit of goods is given. The transportation problem is then to find a least expensive flow of goods from the suppliers to the consumers that satisfies the consumers demand. Signature matching can be naturally cast as a transportation problem by defining one signature as the supplier and the other as the consumer, and by setting the cost for a supplier-consumer pair to equal the *ground distance* between an element in the first signature and an element in the second. The ground distance is defined as the distance between the basic features that are aggregated into the signatures. Intuitively, the solution is then the minimum amount of "work" required to transform one signature into the other. Formally the EMD is defined as a linear programming problem. Let P, Q be two image blocks and $S(P) = \{(p_h, w_h)\}_{h=1}^{N}$, $S(Q) = \{(q_k, w_k)\}_{k=1}^{M}$ their signatures with N and M clusters, respectively; let d_{hk} be the ground distance between two centroids p_h and q_k, and let f_{hk} be the flow between p_h and q_k, defined as the amount of weight of p_h matched to q_k, we want to find a flow that minimizes the overall cost:

$$\mathrm{WORK}[S(P), S(Q), f_{hk}] = \sum_{h=1}^{N}\sum_{k=1}^{M} d_{hk} \cdot f_{hk} \tag{3}$$

with the following constraints:

$$\begin{aligned} & f_{hk} \geq 0, \quad 1 \leq h \leq N,\ 1 \leq k \leq M, \\ & \sum_{k=1}^{M} f_{hk} \leq w_h, \quad 1 \leq h \leq N, \\ & \sum_{h=1}^{N} f_{hk} \leq w_k, \quad 1 \leq k \leq M, \\ & \sum_{h=1}^{N}\sum_{k=1}^{M} f_{hk} = \min\left(\sum_{h=1}^{N} w_h, \sum_{k=1}^{M} w_k\right). \end{aligned} \tag{4}$$

The first constraint assures for unidirectional supplies transportation from $S(P)$ to $S(Q)$. With the second we limit the amount of supplies that can be sent by the clusters in $S(P)$ to their weights. The third constraint allows the clusters in $S(Q)$ to receive no more supplies than their weights, while the last constraint forces to move as much supplies as possible. We call this amount the total flow. Once the transportation problem is solved, and we have found the optimal flow f_{hk}, the EMD is defined as the work normalized by the total flow:

$$\mathrm{EMD}[S(P), S(Q)] = \frac{\sum_{h=1}^{N}\sum_{k=1}^{M} d_{hk} \cdot f_{hk}}{\sum_{h=1}^{N}\sum_{k=1}^{M} f_{hk}}. \tag{5}$$

The normalization is needed when the two signatures have different total weight, to avoid giving more importance to smaller signatures. In general, the ground distance d_{hk} can be any distance and will be chosen according to the problem at hand. We need to define a ground distance that match our purposes. For the extraction of range and domain blocks signatures we deploy a clustering process based on a metric distance as defined in (1). Such a distance was an Euclidean-based metric able to compare pixels and centroids both in the spatial color domains. The comparison is restricted in the spatial domain by r, that is, the medium spatial distance between pixels and the initial distribution of centroids. In the color space the search is limited to the centroids that differ less than the resolution of the human visual system (i.e., the $\mathrm{HVS}_{\mathrm{res}}$). To define the ground distance d_{hk}, we use a similar, but slight different approach. Although we still keep the boundary for the color component, and we use the $\mathrm{HVS}_{\mathrm{res}}$ value to normalize the Euclidean metric, we do not have elements to limit the search area in the spatial domain. Therefore, in the spatial domain, we do not set any constraint, we just normalize the distance component to the maximum measured Euclidean distance between centroids. Moreover, in a matching process based on signatures as above defined, to the success of the search, the importance of the spatial component is not the same as the relevance the color component. In fact, in two image blocks having a similar color distribution, the color position can be very different and this can lead to a weak best match algorithm. As to the above considerations, we propose the following measure for the ground distance:

$$\begin{aligned} d_{hk} &= \sqrt{\lambda \mathbf{dis}_s^2(p_h, q_k) + (1-\lambda)\mathbf{dis}_c^2(p_h, q_k)}, \\ & \lambda \in \mathbb{R},\ 0 < \lambda < 1, \end{aligned} \tag{6}$$

where $\mathbf{dis}_s(\cdot,\cdot)$ and $\mathbf{dis}_c(\cdot,\cdot)$ are the same as in (1), but for the former, here, $r = \max_{h,k}\|p_h - q_k\|_{\mathrm{spatial}}$. In fact, the parameter λ in (6) weights the importance given to the color distance respect to the spatial distance and it is chosen as to maximize the quality in the reconstructed image. It is worth remarking that also d_{hk}, as well as $\mathbf{d}(\cdot,\cdot)$ of (5), is non-negative, symmetric and satisfies the triangle inequality, hence it is a true metric. To extract the fractal code, IFS look for similarities between range and domain blocks by comparing their signatures. IFS works with contractive transformations reducing the size of the domain blocks to the one of the range blocks. Therefore, the matching process compares signatures of same total weight. In this case, since the ground distance d_{hk} is a true metric, also the EMD as to (5) defines a metric space. Moreover, it can be shown that in this particular case

$$\begin{aligned} & \mathrm{EMD}[S(P), S(Q)] < d_{pq}, \\ & p = \frac{1}{w}\sum_{h=1}^{N} w_h p_h, \\ & q = \frac{1}{w}\sum_{k=1}^{N} w_k q_k, \\ & w = w_p = w_k, \end{aligned} \tag{7}$$

where w is the total weight of the two signatures and p, q their average centroids. In other words, the ground distance between the average centroids of two signatures of same total weight is a lower bound for the EMD between the two signatures [16]. This property is used by the IFS process to reduce the complexity of the similarities search algorithm. Using the EMD for the IFS best matching search has several advantages. In fact, comparing summary information of image blocks extracted by a clustering process leads to an increased robustness of the search process to the offset errors. This is not true for the pixel-based RMS approach. Moreover, it is less sensitive to quantization errors due to the intrinsic "averaging" nature of the clustering process.

The extension of the theory to video signals is straightforward. In fractal video coding [22] range and domain blocks become three-dimensional objects and thus, the number of isometries and massive transforms to be computed is higher. This fact dramatically raises the computational cost of the matching algorithm. Therefore, the application of fractal coding to video signals turns out to be possible only by following an accurate policy of data reduction and problem simplification. Three-dimensional zooming is achieved using the fractal code extracted from the sequence treated as a three-dimensional object.

3. Proposed Architecture

Within a framework of interactive HDTV applications, the user should select a scene of interest (i.e., a subsequence corresponding to the desired time interval) to be spatially zoomed and replayed in slow motion. The scene of interest is

then passed to the proposed architecture shown in Figure 3 and explained in this section.

The scene of interest chosen by the user is at first decomposed in homogeneous shots in order to avoid that scene changes are going to be part of the fractal zoom process.

A set of frames containing the scene of interest is processed by video scene decomposition. In the proposed architecture a scene change algorithm is exploited for determining the optimal temporal window for maximizing the video quality of the zoom and slow motion processing. In fact, the scene of interest chosen by the user for slow motion needs to be preprocessed in order to be partitioned into homogeneous video shots in order to avoid that scene changes participate to the fractal zoom process.

A clustering-based segmentation approach is used for scene change detection. Basically, scene changes are identified on the basis of histogram variations and color variations where this variation is significant in one or both of them. Features function of YUV histogram variation and features function of subsampled YUV frame difference are discriminated choosing appropriate threshold values. The choice of such threshold is automatically performed by means of the Otsu method [23]. The scene change detection algorithm is presented in [23].

Let the homogeneous subsequence identified by the scene change algorithm be composed by M frames.

At first, being the computational complexity of the fractal encoder strictly proportional to the amount of data to be processed, frames are grouped into packets (GOPs) with length N. N is chosen according to the temporal activity of the sequence, so that higher values can be selected for slowly changing scenes without a significant time processing increase. Each GOP is treated as a single unit to be coded. The GOP size is chosen according to the temporal activity within the sequence, so that bigger sizes can be selected for slowly changing scenes without a significant time processing increase.

Packets are selected considering the temporal variance of the sequence, estimated by means of the Minimum Square Error (MSE) metric between frames:

$$\mathrm{MSE}(h,k)=\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\left(F_{i,j}^{h}-F_{i,j}^{h+k}\right)^{2}}{M\cdot N},\quad h,k\in[1,2,\ldots,n], \tag{8}$$

where $F_{i,j}^{p}$ is the pixel (i,j) of the frame and, p is the frame position within the sequence, $M \cdot N$ the frame size, and n the number of frames of the sequence. Among the totality of frames composing the sequence, a certain number of key-frames are selected. A packet is defined as composed by a set of adjacent frames temporally located between two consecutives key-frames, as shown in Figure 4.

At the beginning of the division process the first frame of the sequence to be expanded is chosen as initial key-frame. More in general, once a frame h has been identified as the first key-frame for a packet, a successive frame k is marked as the ending key-frame for the packet if

$$\mathrm{MSE}(h,k) > Th, \tag{9}$$

where Th is a threshold selected so that

$$\mathrm{Th}=\frac{\mathrm{MSE}(1,n)}{2}. \tag{10}$$

In other words, for each packet the temporal variance must be lower than the 50% of the temporal variance of the whole sequence. Equation (10) assures at least a two-packet subdivision of the sequence to be expanded. According to (9) and (10), each packet can be composed by a variable number of frames. At the end of the packetization process, each packet is considered for coding as a single unit: in this manner the computational load and, thus, the time consumed for the coding are significantly reduced.

The drawback of this packetization process is that it introduces a discontinuity along the temporal axis. To limit this effect, time overlapping is used: each GOP is coded having as a boundary condition the motion information of the previous one. Owing to this, the presence of a buffer is necessary to assure the process being causal. A more general constraint is that the GOP size must be a multiple of R, size of the range block, and not smaller than D, size of the domain block. This guarantees the packet being partitioned into range and domain blocks, and not into portions of them.

Within each GOP an active scene detector is used to find the "active object" so that a group of three-dimensional blocks is extracted. Each frame is divided into tiles of M × M size. The EMD among corresponding tiles belonging to different frames is computed. If the EMD is higher than a prefixed threshold, tiles are grouped to form a three-dimensional block. The threshold is adaptively evaluated by averaging the EMD over all tiles composing the GOP. The set of the so extracted blocks defines the active object, the remaining blocks constituting the "background."

The active object is suited to be coded with a full three-dimensional fractal coder whereas the static background is processed with a two-dimensional one. Fractal coding is performed according to the IFS theory [9]: at first, data are partitioned into range and domain blocks; then, a domain pool is created by means of domain blocks and their contractive affine transforms. Each range block is then compared to the elements composing the domain pool by means of the EMD and a set of correspondences among range blocks, domain blocks, and affine transforms (i.e., the fractal code) is created.

Using fractal zoom during the decoding step leads to blockness distortion (Figure 5) along both the time and spatial dimensions. This problem derives from partitioning the video into nonoverlapping range blocks during the encoding process, and overall visual quality decreases when high zoom (i.e., above 4× factor) is performed. To contrast this effect, the coding Overlapped Range Blocks (ORBs) technique [12] is used. ORB coding is extended to the three-dimensional case for the active object coding. Background is encoded with a two-dimensional fractal code, since data does not change, on the temporal axis, for background blocks. Extending [12], eight different partitions of the active object and four partitions for the static background are computed. Eight different fractal codes for the active object are extracted and coded independently (Figure 6).

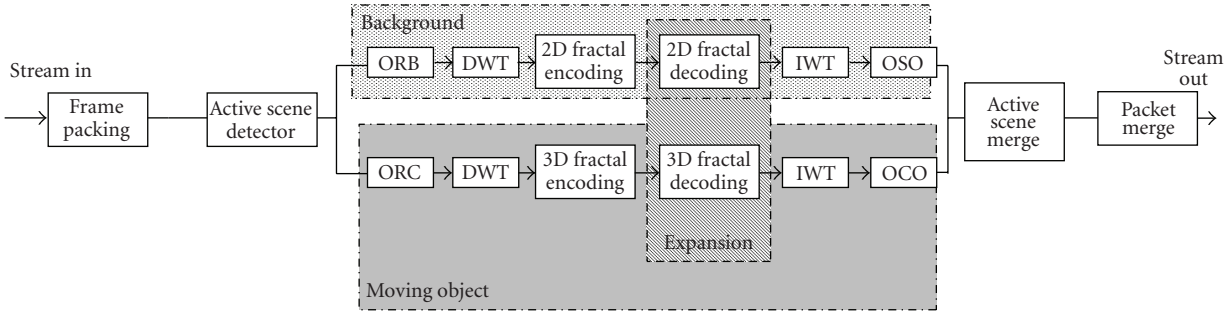

Figure 3: Flowchart of the proposed method.

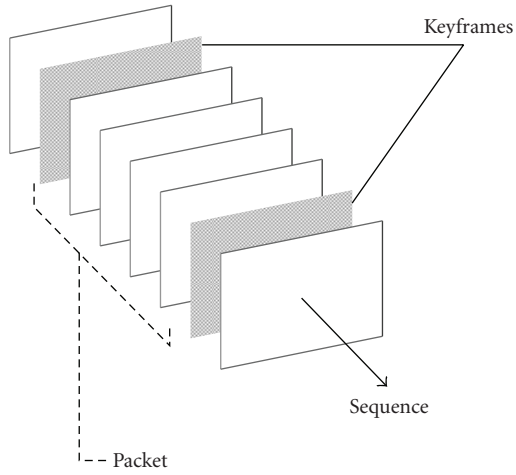

Figure 4: GOP extraction.

Figure 5: Example of block distortion on fractal interpolated frame.

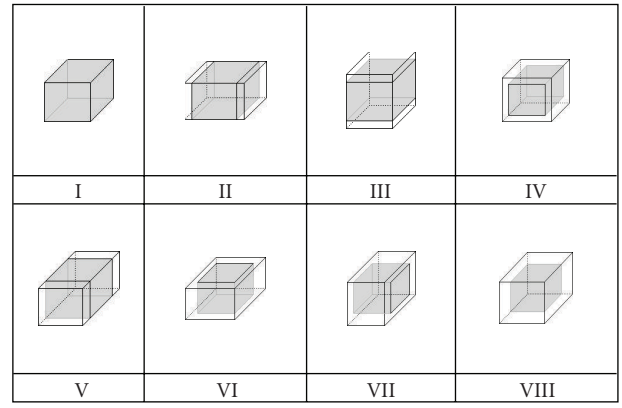

Figure 6: The set of partitions considered for the ORB/OCO process.

At decoding time, the inverse process is applied, and the fractal zoom is performed. An Ordered Cube Overlapping (OCO) postprocess, defined as an extension of Ordered Square Overlapping (OSO) [12], merges the parts created by the overlapped partition of three-dimensional fractal code. The OSO presented in [12] is a windowed median filter that computes the median value from each partition generated by ORB. The technique is applied here in the three-dimensional case and the OCO computes the median among the eight ORB partitions. A drawback of using ORB and OCO is the growth of the computational cost of the fractal coding process.

To cope with high computational burden, a wavelet-based approach is used [7]. For the active object a three-dimensional wavelet subband analysis is computed. For the entire low pass component a fractal code is then extracted using ORB partitioning. For the high-pass components, the following coefficients classification procedure is performed [24]. Let S_m be the mth subband; we denote by $\{x_i^m\}$ the wavelet coefficients of S_m and by $p^m(x)$ the histogram of $\{x_i^m\}$. In $p^m(x)$, starting from the maximum $x_{\max}$ and moving to the tails of the distribution (see Figure 7), two thresholds are identified, that is, $t_1^m, t_2^m : \int_{t_1}^{t_2} p^m(x)dx = K$, $K \in (0, 1]$.

These thresholds identify the wavelet coefficients constituting the active zone for S_m, that is, $S_m^{\text{az}} = \{\forall x \in \{x_i^m\}, x \notin [t_1^m, t_2^m]\}$. In other words, an active zone is composed by those coefficients located on the distribution's tails identified by the above thresholds. After the classification process, a binary-value mask, indicating the position of active zone coefficients within the subband, is extracted. Those coefficients that do not belong to an active zone are discarded, while the S_m^{az} coefficients are ORB partitioned and then fractal encoded. The K parameter is unique for

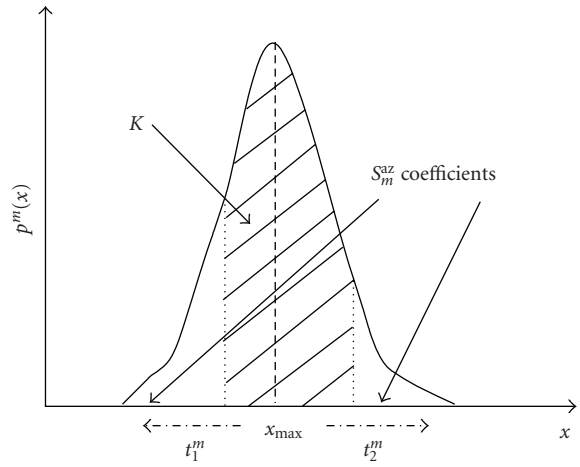

Figure 7: Wavelet coefficient classification process.

all the subbands and controls the speed up, and, on the other hand, the accuracy of the fractal coding process; higher values of K correspond to higher speed up factors, but also to lower final visual quality achieved. At decoding stage OSO/OCO filtering is applied independently to each subband. An additional advantage in terms of time saving of wavelet analysis is the "parallelization" of the entire process that increases the speed in a multithreaded environment.

At decoding time, the inverse process is applied, and the fractal zoom is performed. Since the extracted fractal code is resolution independent, during the decoding process an expansion can be performed independently along each dimension [19].

A three-dimensional (i.e., spatial and temporal) expansion of the active object and two-dimensional spatial zoom (i.e., frames of bigger size) of the background are performed. After the inverse wavelet transformation, an OSO/OCO filtering is performed on the background/active object, respectively. Combined ORB code and OSO/OCO filtering enhance visual quality performance of fractal, by coding reducing blocking artifacts generated by the block based nature of the IFS approach. Finally, an active scene merging and a packets merging processes are applied to release the desired output video sequence.

4. Experimental Results

We tested the effectiveness of the proposed method by comparing the result achieved to those obtained, under the same constraint (i.e., applying the same slow motion factors) by frame replica and classical interpolation techniques. Five HDTV test sequences in 10 seconds shots with 1280 horizontal pixels and 720 vertical pixels (lines), progressively scanned at 50 frames per seconds (namely, 720p/50), were used for experimental tests. Such sequences are freely available at [25]. The sequences are named *CrowdRun, ParkJoy, DucksTakeOff, IntoTree*, and *OldTownCross*. A snapshot from some of these sequences is shown in Figure 8. The first three sequences are classified as "difficult" for what concerns coding complexity while IntoTree and OldTownCross are classified as "easy."

In a framework of broadcasting HDTV, to measure the quality achieved we refer to the video quality assessment described on [26] and formalized in [27]. As to this, the perception of continuous motion by human vision faculties is a manifestation of complex functions, representative of the characteristics of the eye and brain. When presented with a sequence of images at a suitably frequent update rate, the brain interpolates intermediate images, and the observer subjectively appears to see continuous motion that in reality does not exist. In a video display, *jerkiness* is defined as the perception, by human vision faculties, of originally continuous motion as a sequence of distinct "snapshots" [27]. Usually, jerkiness occurs when the position of a moving object within the video scene is not updated rapidly enough. This can be a primary index of a poor performance for a slow motion algorithm. More in general, the total error generated by an incorrect coding of a moving object on a video sequence is representative of spatial distortion and incorrect positioning of the object. In [27] a class of full reference quality metrics to measure end-to-end video performance features and parameters was presented. In particular, [27] defines a framework for measuring such parameters that are sensitive to distortions introduced by the coder, the digital channel, or the decoder. Reference [27] is based on a special model, called the *Gradient Model*. Main concept of the model is the quantification of distortions using spatial and temporal gradients, or slopes, of the input and output video sequences. These gradients represent instantaneous changes in the pixel value over time and space. We can classify gradients into three different types that have proven to be useful for video quality measurement.

(i) The spatial information in the horizontal direction SI_h.

(ii) The spatial information in the vertical direction SI_v.

(iii) The temporal information TI.

Features, or specific characteristics associated with individual video frames, are extracted in quantity from the spatial and temporal information. The extracted features quantify fundamental perceptual attributes of the video signals such as spatial and temporal details. A scalar feature is a single quantity of information, evaluated per video frame. The ITU recommendation [27] divides the scalar features into two main groups: based on statistics of spatial gradients in the vicinity of image pixels and based on the statistics of temporal changes to the image pixels. The former features are indicators of the amount and type of spatial information, or edges, in the video scene, whereas the latter are indicators of the amount and type of temporal information, or motion, in the video scene from one frame to the next.

Spatial and temporal gradients are useful because they produce measures of the amount of perceptual information, or change in the video scene. Surprisingly parameters based on scalar features (i.e., a single quantity of information per video frame) have produced significant good correlation to subjective quality measurement (producing coefficients of

CrowdRun (a) IntoTree (b)

Figure 8: Snapshot of test sequences.

correlation to subjective mean opinion score from 0.85 to 0.95) [26]. This demonstrates that the amount of reference information that is required from the video input to perform meaningful quality measurements is much less than the entire video frame. A complete description of all the features and parameters in [27] is beyond the scope of this paper. In the following a brief summary of the above feature will be given, a mathematical determination of the above features is provided in [27].

(i) *Blurring*. A global distortion over the entire image, characterized by reduced sharpness of edges and spatial detail. Reference [20] defines a Lost Edge Energy Parameter for measuring the blurring-effect, which causes a loss of edge sharpness and a loss of fine details in the output image. This loss is easily perceptible by comparing the Spatial Information (SI) of the output image with the SI of the input image. The lost edge energy parameter compares the edge energy of the input image with the edge energy of the output image to quantify how much edge energy has been lost.

(ii) *Tiling*. Distortion of the image characterised by the appearance of an underlying block encoding structure. Reference [27] defines an HV to non-HV edge energy difference parameter for quantifying the tiling impairment. In contrast to blurring which results in lost edge energy, tiling creates false horizontal and vertical edges. By examining the spatial information (SI) as a function of angle, the tiling effects can be separated from the blurring effects.

(iii) *Error Block*. A form of block distortion where one or more blocks in the image bear no resemblance to the current or previous scene and often contrast greatly with adjacent blocks. Reference [27] defines an Added Motion Energy Parameter for detecting and quantifying the perceptual effects of error blocks. The sudden occurrence of error blocks produces a relatively large amount of added temporal information. So the Added Motion Energy Parameter compares the temporal information (TI) of successive input frames to the TI of the corresponding output frames.

(iv) *Jerkiness*. Motion that was originally smooth and continuous is perceived as a series of distinct snapshots. Reference [27] defines a Lost Motion Energy and Percent Repeated Frames Parameter for measuring the jerkiness impairment. The percent repeated frames parameter counts the percentage of TI samples that are repeated; whereas the average lost motion energy parameter integrates the fraction of lost motion (i.e., sums the vertical distances from the input samples to the corresponding repeated output samples, where these distances are normalised by the input before summing).

To extract the performance metrics we deployed the Video Quality Metric (VQM) software developed by the ITS-Video Quality Research project [28] and compliant with [27]. All tests performed on the different test sequences produced similar outcomes that have been proven to be dependent to the natural temporal activity of the sequences. For the sake of concision, in the following are reported only the results obtained for *CrowdRun* and *IntoTree* video sequences. *CrowdRun* is a sequence presenting a lot of activity since half of the frame is composed by a fast running crowd of people. *IntoTree* is a sequence presenting less activity than *CrowdRun*. Results are reported for a window of 4 seconds corresponding to a cut of 200 frames of the complete video sequence.

Figures 9, 10, 11, 12 show experimental results for the *CrowdRun* sequence. The features mentioned above were computed for the proposed method (Fractal expansion) and for the frame replica and spline cubic interpolation method. The slow motion factors taken into exam are 2×, 4×, and 8×. A general overview of the outcomes shows the advantage in using the proposed technique for higher slow motion factors. Figure 9 shows that blurring distortion for CrowdRun sequence at 2× slow motion is almost the same for all the methods. As slow motion ratio increases the difference between the three techniques becomes more evident. The blurring distortion for the proposed method is lower than the other, and this result is more evident at 4× slow motion. Figure 10 compares the tiling feature. As expected frame replica presents the highest tiling while frame interpolation is superior to the proposed method for 2×

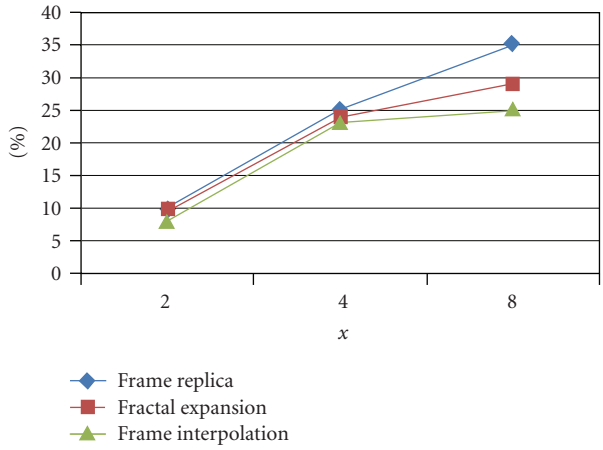

Figure 9: Measured blurring for *CrowdRun* sequence.

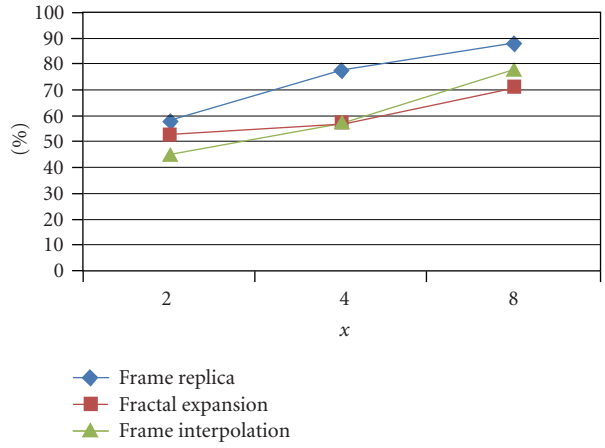

Figure 10: Measured tiling for *CrowdRun* sequence.

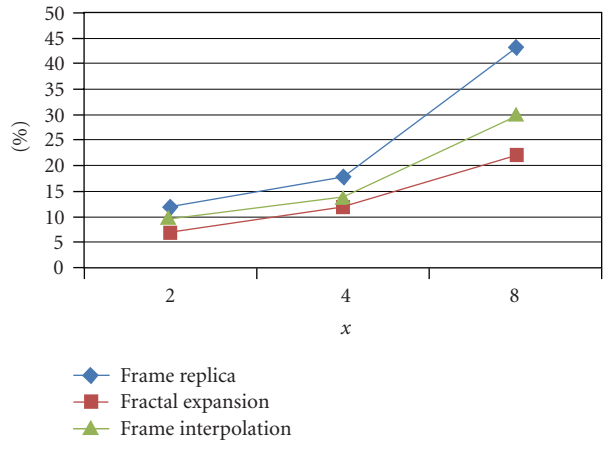

Figure 11: Measured error blocks for *CrowdRun* sequence.

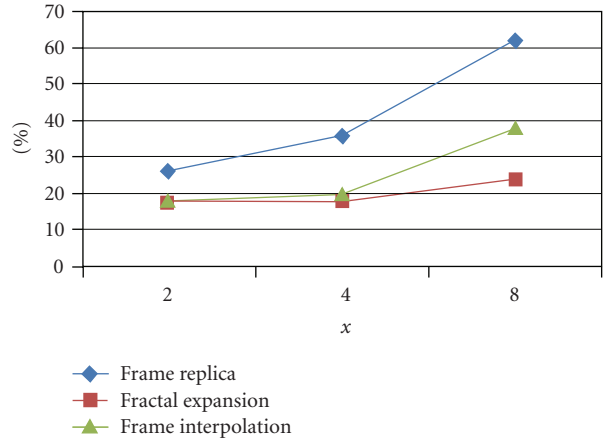

Figure 12: Measured jerkiness for *CrowdRun* sequence.

slow motion factor and vice versa for 8× slow motion factor. Figure 11 compares the error block features. Frame replica results in more error blocks than fractal interpolation and frame interpolation results in more error blocks than fractal expansion. Figure 12 shows the results for jerkiness feature. At 2× and 4× frame interpolation and fractal expansion are comparable while at 8× fractal expansion has a lower jerkiness than frame interpolation.

The use of a combined motion and subband analysis during the fractal coding and again the smoothing properties of the OSO filtering allow the method achieving high performance in terms of fluent motion flow during the presentation in slow motion of video sequences. However, for all the compared methods it is noticeable a degradation of the absolute performance in the presence of high and fast motion in the scene (e.g., for *CrowdRun*). While this is well known for classical frame replica and cubic spline interpolation, the weakness of the proposed algorithm in this case can be justified by the fairly simple method used to estimate the moving object within the sequences. In fact, more sophisticated schemes which assure superior accuracy on motion estimation can be applied and this will be a future task to be pursued during future research.

Figure 13 shows a region of a slow-motion frame from the ParkJoy video sequence. The image of the left is coded by fractal interpolation while the image on the rate is coded by cubic spline frame interpolation. It is evident that both images have a quality noncomparable with the original reference frames. Nevertheless the distortion generated by frame interpolation is subjectively worse than the fractal interpolated frame.

A further experiment session has been set up to compare the performance of the benchmark methods by Peak Signal to Noise Ratio (PSNR). Although it has been proven not to have strong correlation to subjective quality perception in case of video quality assessment [19, 20], PSNR is widely accepted as full reference objective metric for image and video coding.

PSNR experimental tests have been conducted as follows. The test sequences have been subsampled in time by discarding frames. Missing frames were then generated by means of frame replica, cubic spline, and fractal interpolation. PSNRs

(a) (b)

Figure 13: A region of *ParkJoy*, frame no. 117, 4× slow motion, coded by Fractal (a) and Cubic Spline (b) interpolation.

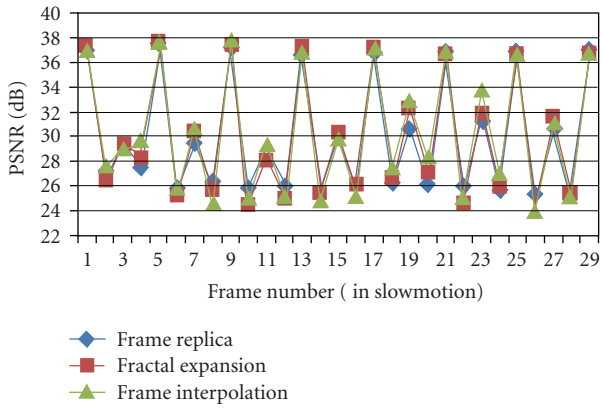

Figure 14: Measured PSNR (dB) for *IntoTree* (4× slow motion).

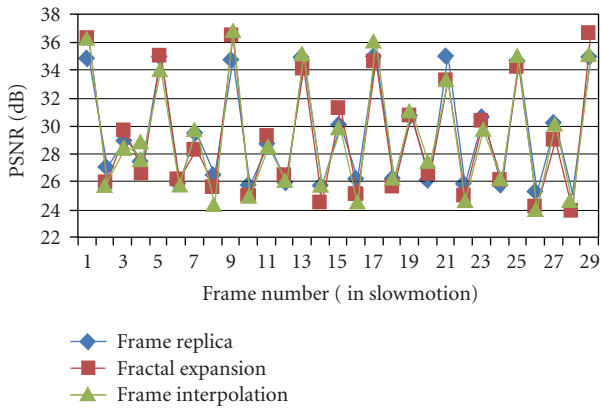

Figure 15: Measured PSNR (dB) for *CrowdRun* (4× slow motion).

on generated frames have been calculated. Figures 14 and 15 show the results for *IntoTree* and *CrowdRun* expanded with 4× slow motion factor. For all the compared methods, the average PSNR achieved with *IntoTree* is higher than the average PSNR achieved with *CrowdRun* due to the relative lower motion presence in *IntoTree*. A more in deep analysis shows, in most cases, also a prevalence of the proposed method over the others so as to confirm the results of both the subjective and objective ITU-R Recommendation BT.1683-based previous analysis.

5. Conclusions

In this work we have presented an alternative technique combining fractals theory and wavelet decomposition to achieve spatial zoom and slow motion replay of HD digital color video sequences, which can be integrated on the decoder set-top boxes irrespective of the specific broadcasting technology. Slow motion replay and spatial zooming are special effects used in digital video rendering. At present, most techniques to perform digital spatial zoom and slow motion are based on interpolation for both enlarging the size of the original pictures and generating additional intermediate frames. In our method fast scene change detection, active scene detection, wavelet subband analysis, and color fractal coding based on Earth Mover's Distance (EMD) measure are used to reduce computational load and to improve visual quality. Experiments show that the proposed scheme achieves better results in terms of overall visual quality compared to the state-of-the-art techniques. The proposed approach to video rendering is compliant with the new trend of convergence of digital TV systems and services.

References

[1] 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.

[2] H. Pan, P. van Beek, and M. I. Sezan, "Detection of slow-motion replay segments in sports video for highlights generation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '01)*, vol. 3, pp. 1649–1652, Salt Lake, Utah, USA, May 2001.

[3] T. Chen, "Adaptive temporal interpolation using bidirectional motion estimation and compensation," in *Proceedings of IEEE International Conference on Image Processing (ICIP '02)*, vol. 2, pp. 313–316, Rochester, NY, USA, September 2002.

[4] K. Hilman, H. W. Park, and Y. Kim, "Using motion-compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 869–877, 2000.

[5] M. E. Al-Mualla, "Motion field interpolation for frame rate conversion," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 652–655, Bangkok, Thailand, May 2003.

[6] C. W. Tang and O. C. Au, "Comparison between block-based and pixel-based temporal interpolation for video coding," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '98)*, vol. 4, pp. 122–125, Monterey, Calif, USA, May 1998.

[7] D. D. Giusto, M. Murroni, and G. Soro, "Slow motion replay of video sequences using fractal zooming," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 103–111, 2005.

[8] M. F. Barnsley and S. Demko, "Iterated function systems and the global construction of fractals," *Proceedings of the Royal Society of London, Series A*, vol. 399, no. 1817, pp. 243–275, 1985.

[9] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 18–30, 1992.

[10] M. Polvere and M. Nappi, "Speed-up in fractal image coding: comparison of methods," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1002–1009, 2000.

[11] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[12] E. Reusens, "Overlapped adaptive partitioning for image coding based on theory of iterated function systems," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '94)*, vol. 5, pp. 569–572, Adelaide, Australia, April 1994.

[13] B. Hurtgen, P. Mols, and S. F. Simon, "Fractal transform coding of color images," in *Visual Communications and Image Processing*, vol. 2308 of *Proceedings of SPIE*, pp. 1683–1691, Chicago, Ill, USA, September 1994.

[14] Y. Zhang and L.-M. Po, "Fractal color image compression using vector distortion measure," in *Proceedings of IEEE International Conference on Image Processing (ICIP '95)*, vol. 3, pp. 276–279, Washington, DC, USA, October 1995.

[15] I. M. Danciu and J. C. Hart, "Fractal color compression in the $L^*a^*b^*$ uniform color space," in *Proceedings of the Data Compression Conference (DCC '98)*, p. 540, Snowbird, Utah, USA, March-April 1998.

[16] S. Cohen and L. Guibas, "The Earth Mover's Distance under transformation sets," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1076–1083, Kerkyra, Greece, September 1999.

[17] L. Atzori, D. D. Giusto, and C. Perra, "Automatic scene change detection in uncompressed video sequences," in *Proceedings of the Tyrrhenian Workshop on Digital Communications (IWDC '02)*, Capri, Italy, September 2002.

[18] Y. Rubner, L. J. Guibas, and C. Tomasi, "The Earth Mover's Distance as a metric for image retrivial," Tech. Rep. STAN-CS-TN-98-86, Stanford Computer Science Department, Stanford, Calif, USA, 1998.

[19] E. Polidori and J.-L. Dugelay, "Zooming using Iterated Function systems," in *Proceedings of the NATO ASI Conference on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[20] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.

[21] S. T. Rachev, "The Monge-Kantorovich mass transference problem and its stochastic applications," *Theory of Probability and Its Applications*, vol. 29, no. 4, pp. 647–676, 1987.

[22] K. U. Barthel and T. Voye, "Three-dimensional fractal video coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '95)*, vol. 3, pp. 260–263, Washington, DC, USA, October 1995.

[23] O. Nobuyuki, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[24] M. Ancis and D. D. Giusto, "Image data compression by adaptive vector quantization of classified wavelet coefficients," in *Proceedings of the 6th IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing*, vol. 1, pp. 330–333, Victoria, Canada, August 1997.

[25] ftp://vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat.

[26] S. Wolf, "Measuring the end-to-end performance of digital video systems," *IEEE Transactions on Broadcasting*, vol. 43, no. 3, pp. 320–328, 1997.

[27] ITU-R Recommendation, "Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference," Tech. Rep. BT.1683, International Telecommunication Union, Geneva, Switzerland, 2004.

[28] Video Quality Research project, http://www.its.bldrdoc.gov/n3/video/.