Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2014, Article ID 645145, 11 pages
http://dx.doi.org/10.1155/2014/645145

Research Article

Activity-Based Scene Decomposition for Topology Inference of Video Surveillance Network

Hongguang Zhang,[1,2] Jianzhu Cui,[1] Peng Wang,[1] and Shibao Zheng[2,3]

[1] *Shanghai Advanced Research Institute, Chinese Academy of Sciences, China*
[2] *Shanghai Key Laboratory of Digital Media Processing and Transmission, China*
[3] *Shanghai Jiao Tong University, China*

Correspondence should be addressed to Hongguang Zhang; hongguang.zhang@gmail.com

Received 14 October 2013; Accepted 12 January 2014; Published 26 February 2014

Academic Editor: Mohamad Sawan

The topology inference is the study of spatial and temporal relationships among cameras within a video surveillance network. We propose a novel approach to understand activities based on the visual coverage of a video surveillance network. In our approach, an optimal camera placement scheme is firstly presented by using a binary integer programming algorithm in order to maximize the surveillance coverage. Then, each camera view is decomposed into regions based on the Histograms of Color Optical Flow (HCOF), according to the spatial-temporal distribution of activity patterns observed in a training set of video sequences. We conduct experiments by using hours of video sequences captured at an office building with seven camera views, all of which are sparse scenes with complex activities. The results of real scene experiment show that the features of histograms of color optic flow offer important contextual information for spatial and temporal topology inference of a camera network.

1. Introduction

Video surveillance networks are being widely deployed in public security field to monitor wide areas and detect events. With the development of intelligent video and data fusion, collaborative information processing among multiple smart cameras is becoming more and more essential to identify, reconstruct, and track targets automatically. To facilitate more efficient multicamera surveillance, growing research efforts have been undertaken on automated activity understanding in camera networks, focusing on camera topology inference [1], camera placement, scenes decomposition, and activity analysis [2]. The camera placement is a typical optimization problem, where some constraints are given by the characteristics of the camera (field of view and focal length) and the quality (resolution), as well as the environment (obstacle and occlusions). Semantic scenes decomposition is the basis for camera topology inference. The aim of topology inference is to infer spatial and temporal relationships among cameras. An example is shown in Figure 1 in our experiment of camera topology inference, which describes the relationships of 7 cameras within a building in Shanghai. As for scenes decomposition and activity analysis, one wishes to understand activities captured by multiple cameras holistically by building activity models. These problems are nontrivial, especially given multiple disjoint cameras with nonoverlapping views, in which activities can only be observed partially with different views being separated by unknown time gaps. The unknown and large separation of cameras in space and over time increases the uncertainties in activity understanding due to drastic feature variations and temporal discontinuity in visual observations.

Visual coverage is the most important constraint for traditional camera placement. The AGP (Art Gallery Problem) [3] is a classical theory on how to place guards in an arbitrarily shaped polygon. However, AGP cannot be applied to camera placement directly. Because AGP theory assumes the sensor has unlimited visibility, while every visual sensor or camera has its constraints on sensing range referred to as FoV (Field of View). During the last two decades, the camera placement attracts lots of research interests. Hörster and Lienhart [4]

Figure 1: Camera topology inference.

developed a binary integer programming model by discretizing the region and determined the minimum number of cameras needed to cover the space completely at a given sampling frequency. Zhao and Cheung [5] proposed a general framework for camera placement, where the goal is to identify distinctive visual features of objects in two or more camera views. Gonzalez-Barbosa et al. [6] specifically describes the surveillance model of directional cameras and omnidirectional cameras in 2D grid graphs and simulates the minimal number of cameras under the situation of simultaneously using two kinds of cameras. Yabuta and Kitazawa [7] have introduced the concept of "Essential Region." The essential areas are fully covered, while ordinary ones are weighted to be covered; the number of cameras have been further decreased. Gupta et al. [8] proposed a framework for optimal camera placement, giving simultaneous consideration to different qualitative aspects using multiobjective genetic algorithm.

In [9], Loy et al. proposed a semantic scene segmentation method based on static and moving foreground, which got good performance for crowded pedestrian. But in the building, the number of pedestrians is sparse; Loy' method is not very well in this situation. A solution to semantic scene segmentation and activity understanding seems to be tracking objects within and across camera views. Indeed, most previous methods rely on tracking to detect entry and exit events [10, 11]. These methods generally assume reliable object detection. However, these assumptions are invalid in real-world surveillance settings.

2. System Overview

As shown in Figure 2, we propose a novel approach to understanding activities from their observations monitored through multiple nonoverlapping cameras.

Firstly, a novel 3D surveillance model is proposed to simulate the FoV of a camera. An optimal camera placement scheme is then presented by using a binary integer programming algorithm, in order to maximize the visual coverage of video surveillance network.

Secondly, in our approach each camera view is decomposed automatically into regions based on the correlation of object dynamics across different spatial locations in all camera views.

A new method of histograms of color optical flow (HCOF) is then formulated to decompose each camera view. Then the correlations of regional activities observed within and across multiple camera views are discovered and quantified in a single common reference space. We automatically decompose a complex scene into N regions, according to the spatial-temporal distribution of activity patterns observed in a training set of video sequences.

In particular, the image space is first divided into equal-sized blocks with $10 * 10$ pixels each. histogram of color optical flow was computed using Horn-Schunck (HS) model over the whole image space.

Correlation distances are computed among local block activity patterns to construct an affinity matrix, which is then used as an input to a spectral clustering algorithm for semantic scene decomposition. Given the scene decomposition, regional activity patterns of a camera view are formed based on the local block.

We demonstrate the effectiveness of the proposed approach using several hours of videos captured at 0.5 fps from a building with seven camera views, all of which feature sparse scene and complex activities. We show that the features of histograms of color optical flow offer important contextual information for spatial and temporal topology inference of a camera network.

The rest of this paper is organized as follows. In Section 3, we describe the blind-area issue caused by the 2D model and introduce our 3D surveillance model. In Section 4, the coverage requirement and the camera placement scheme is discussed. In Section 5, we describe the histograms of

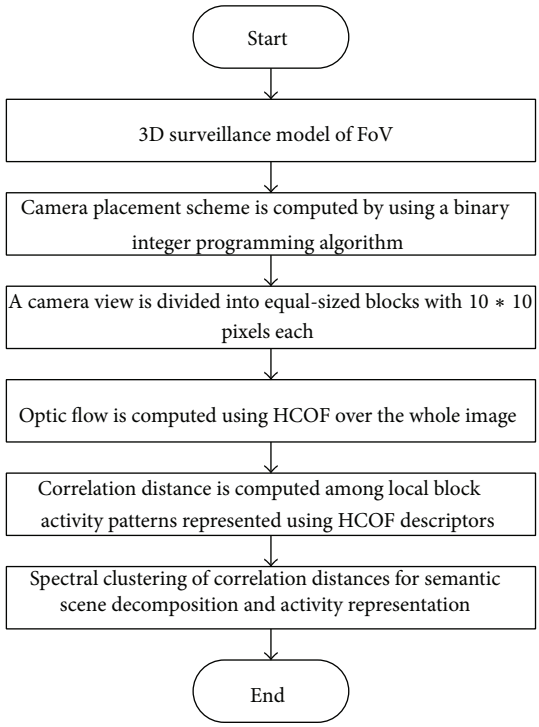


Figure 2: Activity-based scene decomposition.

oriented color optical flow. In Section 6, we describe the local block activity pattern representation. In Section 7, activity-based scene decomposition is presented. In Section 8, we present scene decomposition results of the proposed method. Finally, we summarize the paper.

3. Blind-Area Issue and 3D Surveillance Model

In the actual surveillance scene, the sensing area of a camera and the target to be monitored are both in 3D. However, the 3D scene is usually simplified to 2D for the convenience of computation. Since a directional camera has perspective restrictions, there are always blind areas existing under the kind of camera. Figure 3(a) illustrates the blind-area issue, where the target is out of the sensing area. But the 2D surveillance model would consider that the target can be covered by the upper camera. When applying the 2D model, the blind-area issue is not able to be predicted, which will reduce the coverage performance of the video surveillance network.

This paper proposes to extend the classical 2D surveillance model to 3D as in Figure 3(b) in order to avoid the blind-area issue. The sensing area of the camera is the shadowed cone approximately, which is characterized by parameters of a and d. Our camera placement task is to find out the camera configuration parameters in the 3D surveillance model, including the minimum number of cameras that has to be deployed, the coordinate value (x_0, y_0, z_0) to denote the location, and the rotation angle α and tilt angle β to denote the camera posture.

The XYZ coordinate system (we call it as scene coordinate) is established from the real surveillance scene. We establish a new coordinate system $X'Y'Z'$ (we call it the camera coordinate) whose origin is the location of the camera and x-axis passes through the midline of the sensing cone. On the basis of homogeneous coordinate transformation, we establish the formula as defined by

$$\mathbf{X}' = \mathbf{X}\mathbf{T}\mathbf{R}_Z\mathbf{R}_Y, \tag{1}$$

where $\mathbf{X}' = [x', y', z']$ and $\mathbf{X} = [x, y, z]$ are the coordinate values of the point in the two coordinate systems, respectively;

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix} \tag{2}$$

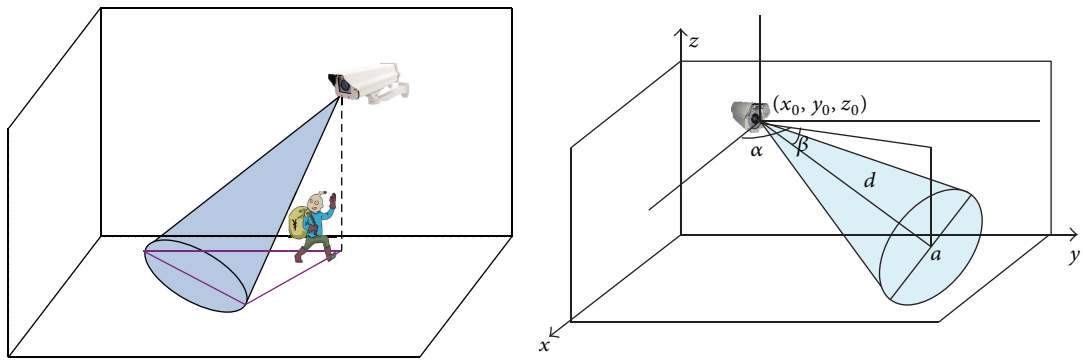
and $[x_0, y_0, z_0]$ are the camera position in scene coordinate;

$$\mathbf{R}_Z = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_Y = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

are transformation matrices.

According to Geometrical Visibility Analysis, the point in camera coordinate can be monitored as long as the following conditions are satisfied:

$$0 \le x' \le d, \qquad \left|y'\right| \le \frac{a}{2 * d}x', \qquad \left|z'\right| \le \frac{a}{2 * d}x'. \tag{4}$$



(a) The blind-area issue (b) The 3D surveillance model

Figure 3: An instance of blind-area issue caused by 2D model and the 3D surveillance model of camera.

If obstacles exist in the surveillance area, we need to guarantee that the obstacles would not impact the sensing area of a camera, that is, no obstacles across the lineation of camera and test point. We consider the obstacles in surveillance area mainly are bottom-to-top like uprights, walls,and so on; so we analyze it in 2D plane. From Figure 4(a), we know when the point is inside the obstacle, there is

$$S_{\Delta PAB} + S_{\Delta PBC} + S_{\Delta PCD} + S_{\Delta PDA} = S_{ABCD}, \quad (5)$$

where S represents the area of triangle or rectangle.

Similarly, when the point is outside the obstacle, see Figure 4(b), there is

$$S_{\Delta PAB} + S_{\Delta PBC} + S_{\Delta PCD} + S_{\Delta PDA} > S_{ABCD}. \quad (6)$$

So besides condition (4), we have to ensure that all the points in the lineation of camera and test point satisfy condition (6).

4. Visual Coverage Constraint

First, we discretize the surveillance scene for convenient processing. The process involves region division, defining essential regions, and defining test points of the region, which is illustrated in Figure 5.

We take a real exhibition center with the size of 90(m) × 64(m) × 20(m) as the surveillance scene. The whole area is divided into small regions by the function blocks (booth, gate, service facility, and so on). In order to reduce the number of cameras and costs, several regions are categorized as essential regions according to the particular requirements, for instance, the entrance of the hall. Canceling the limit that there is only one test point in a region, we propose an adaptive method of test point selection. The algorithm will search the region and set a test point at the center of every 4(m) × 4(m) plane. The number and the position of test points are adaptive to the size and the shape of the regions. This method can enhance the surveillance quality. The result of the surveillance scene discretization is shown in Figure 6. Surveillance regions in the real scene are a set of cubes, and we use floor plans just for convenience. We totally get 64 regions and 163 test points, in which 10 regions and 29 points are essential.

We assume that the camera can only be deployed at the center of the discrete regions. We need to find out the optimal locations and angles of the cameras with constraints of surveillance coverage and deployment cost.

We define a 0-1 decision variable $x_{i\alpha\beta}$ to represent whether there is a camera with rotation angle α and tilt angle β in region i, and it is equal to 1 if the camera exists; otherwise, it is equal to 0. As for the coverage of test point j, we define the variable $p_{i\alpha\beta,j}$. It is equal to 1 if the camera in region i, with orientation α and β, can cover point j; otherwise, it is equal to 0.

If there is only one test point in region j, it can be monitored by the surveillance network only if $\sum_i \sum_\beta \sum_\alpha p_{i\alpha\beta,j} x_{i\alpha\beta} \geq n$; that is, the test point in this region can be monitored by at least n cameras. If the test points in a region are more than one, we can similarly define $p_{i\alpha\beta,j_m}$ and p_{j_m} to represent whether the test point m in region $j$can be monitored, and

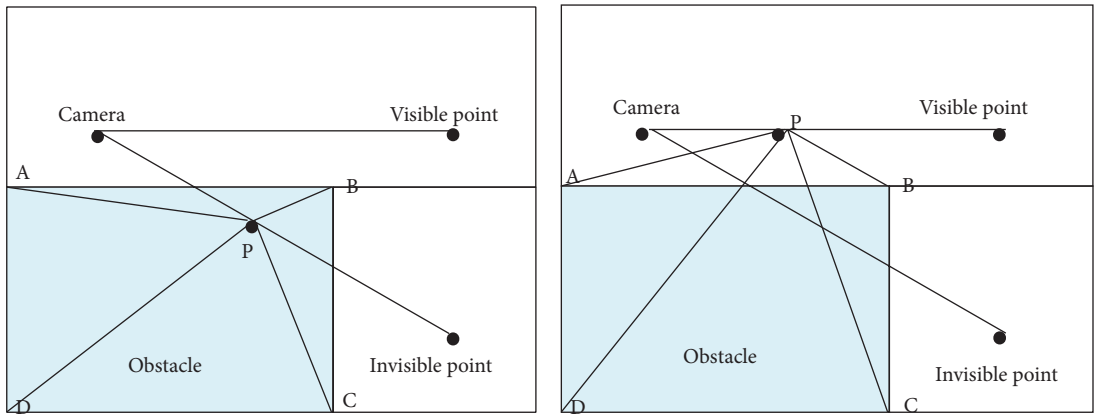
$$p_{j_m} = \begin{cases} 1 & \text{if } \sum_i \sum_\beta \sum_\alpha p_{i\alpha\beta,j_m} x_{i\alpha\beta} \geq n \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

We can set the constraint flexibly according to the surveillance importance of the region when considering the whole region coverage. For example, we require all of test points to be observed by the cameras for the essential region, while part of the test points being observed is enough for the ordinary region.

We define s_j to define whether region j is essential. For all the essential regions ($s_j = 1$), they must be totally covered by surveillance network for any cost, while for ordinary ones ($s_j = 0$), defining y_j as (7) for their coverage:

$$y_j = \begin{cases} 1 & \text{if } \sum_i \sum_\beta \sum_\alpha p_{i\alpha\beta,j} x_{i\alpha\beta} \geq n \text{ or} \\ & \sum_m p_{j_m} \geq N \quad \text{when } s_j = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

For the purpose of getting a tradeoff between coverage level and deployment cost, we apply the balance scheme in [7]. The video surveillance network should cover all the



(a) Point inside the obstacle (b) Point outside the obstacle

Figure 4: The impact of obstacle: if the line segment crosses the obstacle, then the point cannot be monitored.

essential regions and as many ordinary regions as possible. We define M as expected gains. If one camera can bring about M or more regions to be monitored, then the camera is deployed or it will not be deployed. In practice, we can set proper expected gain M according to specific circumstance to achieve the balance between coverage and cost.

Based on the above analysis, the camera placement algorithm can be abstracted as the following optimal problem with constraints:

$$\min \sum_i \sum_\beta \sum_\alpha x_{i\alpha\beta} - \frac{1}{M} \sum_j \left(1 - s_j\right) y_j, \tag{9}$$

subject to

$$\begin{aligned} &\sum_i \sum_\beta \sum_\alpha p_{i\alpha\beta,j} x_{i\alpha\beta} \geq n \quad \text{or} \quad \sum_m p_{j_m} \geq N' \quad \text{for } s_j = 1, \\ &-y_j + \sum_i \sum_\beta \sum_\alpha p_{i\alpha\beta,j} x_{i\alpha\beta} \geq 0 \quad \text{for } s_j = 0. \end{aligned} \tag{10}$$

5. Features Selection: Histograms of the Oriented of Color Optical Flow (HCOF)

In [12], Wang and Snoussi proposed histograms of the orientation of optical flow (HOF) for abnormal events detection. In this paper, we extend HOF to use HCOF for scenes decomposition. HCOF can offer more useful features for semantic scenes decomposition than HOF.

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image. It can give important information about the spatial arrangement of the objects and the change rate of this arrangement [13]. Abnormal action can be exhibited by the direction and the amplitude of the movement; optical flow is chosen for scene description. Horn and Schunck [13] proposed the algorithm introducing a global constraint of smoothness to computer optical flow. The basic Horn-Schunck (HS) optical method is used in our work. The HS method combines a data term that assumes constancy of some image property with a spatial term that models how the flow is expected to vary across the image [14]. For two-dimensional image sequences, the optical flow is formulated as a global energy functional.

We propose, in this paper, to compute the histograms of the orientation of optical flow (HOFs). HOFs are similar to histograms of oriented gradients (HOGs) [15], but they are computed quite differently. HOGs are computed in dense grids of the gradients of the image at a single scale without dominant orientation alignment. HOFs are computed in dense grids of the optical flow. HOFs are also different from the descriptor proposed in [16], where the differential optical flow is considered. Here, HOFs descriptors are computed over dense and overlapping grids of spatial blocks, with optical flow orientation features extracted at fixed resolution and gathered into a high dimensional feature vector [17].

We propose, in this paper, to compute the histograms of the orientation of optical flow (HCOF). HOFs are similar to histograms of oriented gradients (HOGs) [9], but they are computed quite differently. HOGs are computed in dense grids of the gradients of the image at a single scale without dominant orientation alignment. HCOFs are computed in dense grids of the optical flow. HCOFs are also different from the descriptor proposed in [10], where the differential optical flow is considered. Here, HCOFs descriptors are computed over dense and overlapping grids of spatial blocks, with optical flow orientation features extracted at fixed resolution and gathered into a high dimensional feature vector [11]. HCOFs are more useful than HOGs for motion detection.

The image is divided into small spatial regions ("cells"), for each cell accumulating a local 1D histogram of the orientation optical flow over the pixels of the cell. Several cells combine into a block.

Figure 7 shows a 2×2 cells HOFs descriptor. The HOFs feature vectors are on individual frames. Horizontal and vertical optical flows voting into orientation bins are in 0°–360°. In our examination, a block contains $2 * 2$ cells, while a cell contains $10 * 10$ cells pixels. HOFs are computed with an overlapping proportion of two neighboring blocks.

The combined histogram entries form the representation. For better invariance to illumination, shadowing, and so forth, it is also useful to contrast-normalize the local

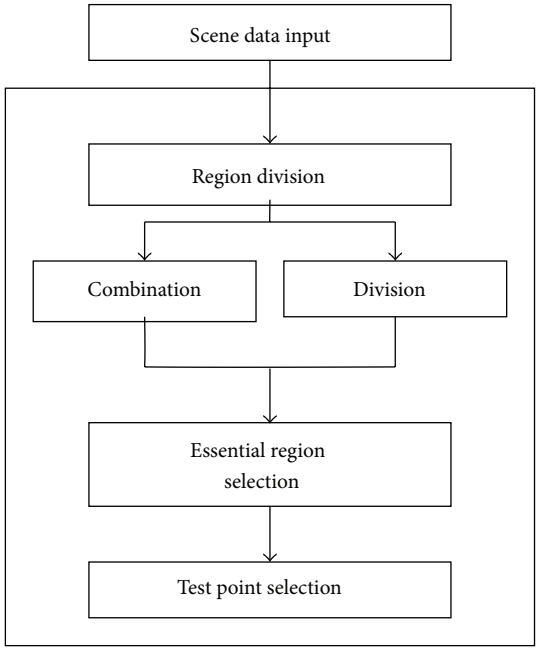


Figure 5: The process of surveillance scene discretization.

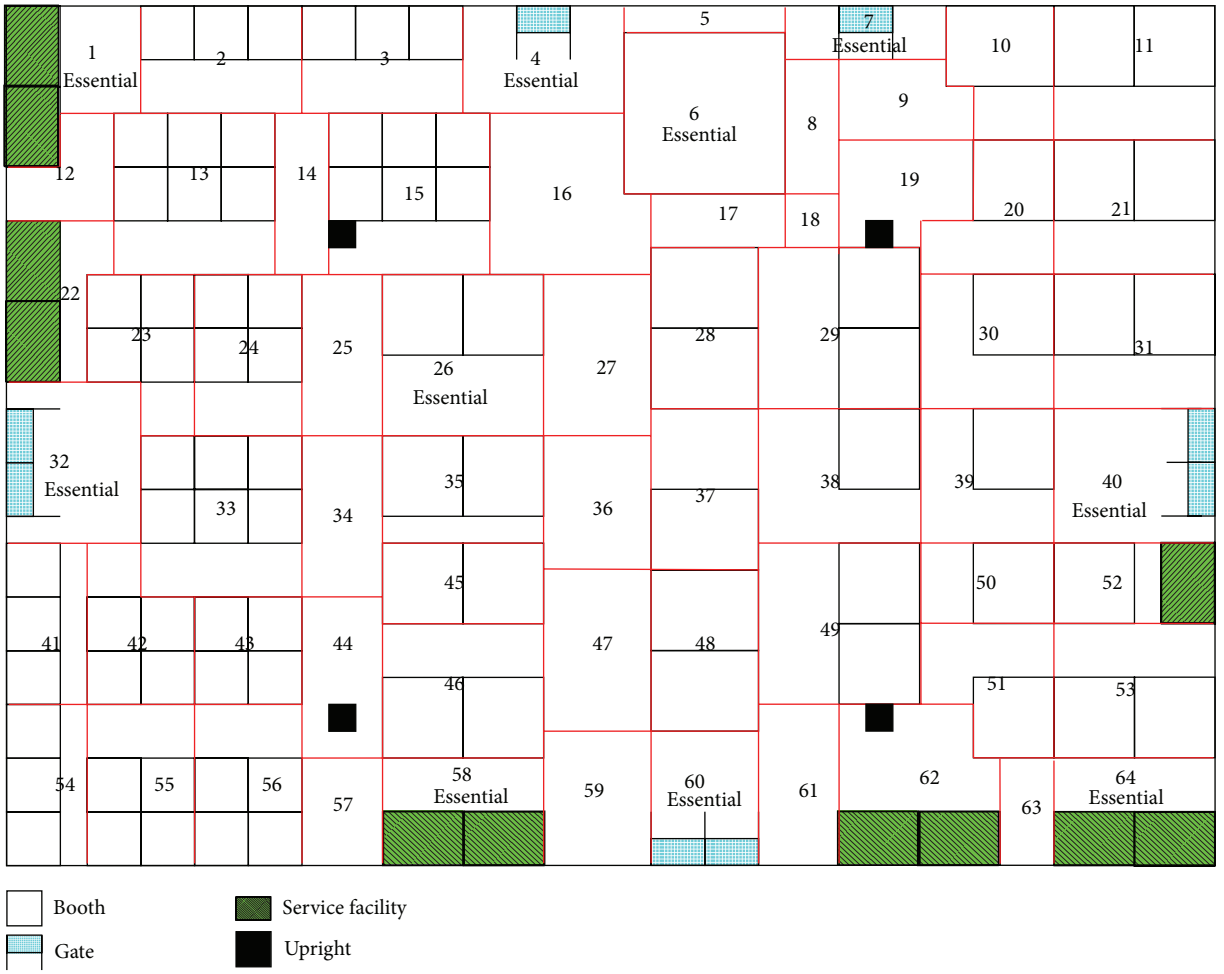


Figure 6: The result of surveillance scene discretization.

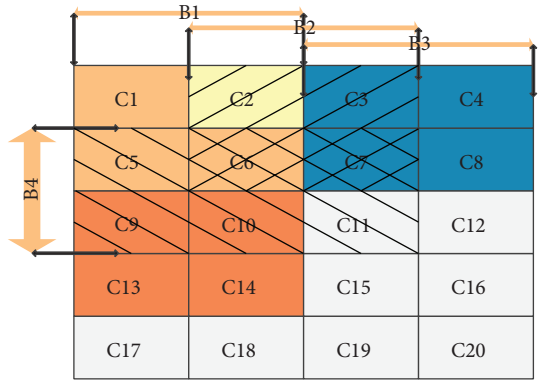


Figure 7: Blocks allocation.

responses before using them. This can be done by accumulating a measure of local histogram "energy" over spatial regions ("blocks") and using the results to normalize all of the cells in the block. We will refer to the normalized descriptor blocks as histogram of oriented optical flow (HOF) descriptors.

Different block normalization schemes are chose;, the normalization methods are shown in (5). Let V_i be the unnormalized vector of descriptor feature, let V_i^* be its 2-norm, and let ε be a small constant. The number of the characteristics of each block is $4 \times 9 = 36$ dimension; $K = 36$.

Consider

$$V_i^* = \frac{V_i}{\sqrt{\sum_{i=1}^{K} V_i^2 + \varepsilon}}. \tag{11}$$

The YUV model decomposes the color as brightness (Y) and a color coordinate system (U, V). The difference between the two is the description of the color plane. H and S describe a vector in polar form, representing the angular and magnitudinal components, respectively. Y, U and V, however, form an orthogonal Euclidean space.

Conversion formula of YUV and RGB is as follows:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B, \\ U &= -0.147R - 0.289G + 0.436B, \\ V &= 0.615R - 0.515G - 0.100B. \end{aligned} \tag{12}$$

The number of the HCOF characteristics of each block is $36 \times 3 = 108$ dimension.

6. Local Block Activity Pattern Representation

Consider

$$\begin{aligned} U_1 &= \left(V_{1,1}^*, \ldots V_{i,j}^* \ldots V_{1,T}^*\right), \\ &\vdots \\ U_i &= \left(V_{i,1}^*, \ldots V_{i,j}^* \ldots V_{i,T}^*\right), \\ &\vdots \\ U_{108} &= \left(V_{108,1}^*, \ldots V_{i,j}^* \ldots V_{108,T}^*\right). \end{aligned} \tag{13}$$

First, we divide the image space of a camera view into equal-sized blocks with $(2 * 10) \times (2 * 10)$ pixels each (Figure 7). Activity patterns of a block are then represented as a 108 dimension time series.

$V_{i,j}^*$ represents ith feature of jth frame. U_i represents ith activity pattern in each block in the image space. T is the total number of frames. Note that T needs to be sufficiently large to cover enough repetitions of activity patterns, depending on the complexity of a scene.

7. Activity-Based Scene Decomposition

After feature extraction, we group blocks into regions according to the similarity of local spatiotemporal activity patterns represented as histograms of oriented color optical flow (HCOF). Specifically, two blocks are considered similar and grouped together if they are closed to each other spatially and exhibit high correlations in HCOF activities over time. The grouping process begins with computing correlation distances among local activity patterns of each pair of blocks.

A correlation distance is defined as a dissimilarity metric derived from Pearson's correlation coefficient [18] given as

$$\rho_{U_i U_j} = \frac{Cov\left(U_i, U_j\right)}{\sqrt{D\left(U_i\right)}\sqrt{D\left(U_j\right)}}. \tag{14}$$

Correlation distant is given as

$$D_{U_i U_j} = 1 - \rho_{U_i U_j}. \tag{15}$$

Upon obtaining the normalised affinity matrix, we employed spectral clustering method proposed by Zelnik-Manor and Perona [19] to decompose each camera view into regions with the optimal number of regions being determined automatically.

8. Experimental Results

8.1. Dataset. The dataset employed in our experiments contain synchronized views, captured at a frame rate of 0.5 fps from uncalibrated and disjoint cameras installed in a building of Shanghai Advanced Research Institute, China Academy of

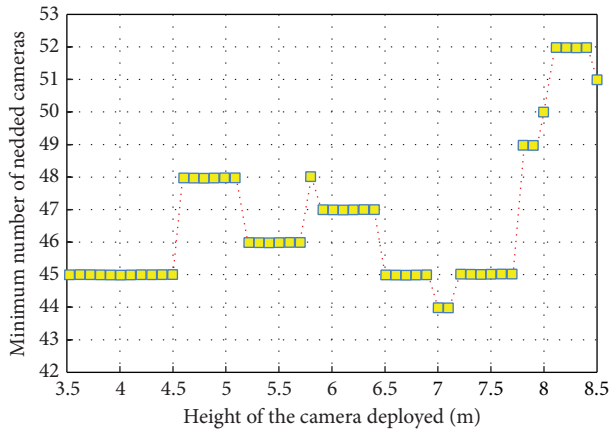


Figure 8: The minimum number of cameras versus deployed height.

Figure 9: The experiment result of optical flow.

(a) (b)

Figure 10: Experiment of activity-based scene decomposition, (a) experiment of activity-based scene decomposition based on HCOF, and (b) experiment of activity-based scene decomposition based on static and moving foreground.

Science. Each image frame has a size of 320 × 230 pixels. A snapshot of each of the 7 camera views and the camera topology of this building are depicted in Figure 1.

8.2. Visual Coverage. We conduct the simulation experiments on the surveillance scene of Figure 6. We set the directional camera with focal length 20(m) and field of view 30 degrees.

Our proposed 3D surveillance model adds the camera heights of cameras and test points as optimization parameters and applies adaptive selection scheme of test points. Figure 8 shows the minimum number of camera versus the height of cameras. The minimum number is showing a general tendency to increase as the height rises; that is, the probability of the test points appearing in the blind area of camera increases as the camera height rises and more cameras are needed to cover these points. When the cameras are placed at the height of 7(m) or so, the minimum number of cameras is needed to satisfy the coverage requirement. In that case, the configurations of camera match the scene suitably, which means it is the optimal height. We can place the cameras at this height when deploying video surveillance network for this scene. It is to be noted that some singular points exist in Figure 8 (e.g., the point with height of 5.8(m)), because the location and rotation angle of cameras are not continuous.

8.3. The Experiment of Optical Flow. The experiment of optical flow is shown in Figure 9.

8.4. The Experiment of Scene Decomposition. The experiment of activity-based scene decomposition is shown in Figure 10.

We used 3000 frames (≈1-hour in length) from each camera view for activity-based scene decomposition. In particular, the seven camera views from dataset were automatically decomposed into several regions (Figure 5). As can be seen from Figure 5, the camera views were decomposed automatically into semantically meaningful regions.

It is difficult to provide quantitative result on semantic scene decomposition as the correct region segmentation is subjective, especially when the segmentation is not based on visual information but activity patterns observed over time. We performed only qualitative comparisons between scene decomposition methods introduced by Loy et al [9] and our method (Figure 5). The two methods differ mainly in their feature representations, that is, time-series representation in Loy's method and HCOF in our method. We found that our method yielded more meaningful region boundaries. Experiment of activity-based scene decomposition based on HCOF is better than experiment based on static and moving foreground. Some results are shown in Figure 5.

9. Discussion and Future Work

In this work we have presented a novel approach to multicamera activity understanding by modeling the correlations within a video surveillance network. In particular, a camera placement algorithm is presented for collaborative information processing between correlative cameras. Then we introduced HCOF to detect and quantify correlation and temporal relationships between partial observations across local regions. Experimental results have shown that the activity correlations are useful for activity interpretation and video temporal segmentation. Consequently, as demonstrated through our experiments, it can be applied to the topology of a camera network and most challenging surveillance videos for future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the Opening Project of Shanghai Key Laboratory of Digital Media Processing and Transmission (Grant no. 2011KF02).

References

[1] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, pp. II205–II210, July 2004.

[2] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 758–767, 2000.

[3] J. Urrutia, "Art gallery and illumination problems," in *Handbook of Computational Geometry*, pp. 973–1027, North-Holland, 2000.

[4] E. Hörster and R. Lienhart, "Approximating optimal visual sensor placement," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 1257–1260, July 2006.

[5] J. Zhao and S.-C. S. Cheung, "Multi-camera surveillance with visual tagging and generic camera placement," in *Proceedings of the 1st ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '07)*, pp. 259–266, September 2007.

[6] J.-J. Gonzalez-Barbosa, T. García-Ramírez, J. Salas, J.-B. Hurtado-Ramos, and J.-D. Rico-Jimenez, "Optimal camera placement for total coverage," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 844–848, May 2009.

[7] K. Yabuta and H. Kitazawa, "Optimum camera placement considering camera specification for security monitoring," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '08)*, pp. 2114–2117, May 2008.

[8] A. Gupta, K. A. Pati, and V. K. Subramanian, "A NSGA-II based approach for camera placement problem in large scale surveillance application," in *Proceedings of the IEEE International Conference on Intelligent and Advanced Systems (ICIAS '12)*, pp. 347–352, 2012.

[9] C. C. Loy, T. Xiang, and S. Gong, "Multi-camera activity correlation analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1988–1995, 2009.

[10] E. E. Zelniker, S. Gong, and T. Xiang, "Global abnormal behaviour detection using a network of CCTV cameras," in *Proceedings of the IEEE International Workshop on Visual Surveillance*, 2008.

[11] X. Wang, K. Tieu, and E. L. Grimson, "Correspondence-free activity analysis and scene modeling in multiple camera views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 56–57, 2010.

[12] T. Wang and H. Snoussi, "Histograms of optical flow orientation for visual abnormal events detection," in *Proceedings of the IEEE 9th International on Advanced Video and Signal-Based Surveillance (AVSS '12)*, 2012.

[13] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.

[14] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2432–2439, June 2010.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 886–893, June 2005.

[16] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," *European Conference on Computer Vision*, vol. 3952, pp. 428–441, 2006.

[17] N. Dalal, *Finding people in images and videos [Ph.D. thesis]*, Insttut National Polytechnique de Grenoble, 2006.

[18] T. Warren Liao, "Clustering of time series data—a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[19] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2004.