Multi-view video segmentation and tracking for video surveillance

Gelareh Mohammadi, Frederic Dufaux*, Thien Ha Minh, Touradj Ebrahimi

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

ABSTRACT

Tracking moving objects is a critical step for smart video surveillance systems. Despite the complexity increase, multiple camera systems exhibit the undoubted advantages of covering wide areas and handling the occurrence of occlusions by exploiting the different viewpoints. The technical problems in multiple camera systems are several: installation, calibration, objects matching, switching, data fusion, and occlusion handling. In this paper, we address the issue of tracking moving objects in an environment covered by multiple un-calibrated cameras with overlapping fields of view, typical of most surveillance setups. Our main objective is to create a framework that can be used to integrate object-tracking information from multiple video sources. Basically, the proposed technique consists of the following steps. We first perform a single-view tracking algorithm on each camera view, and then apply a consistent object labeling algorithm on all views. In the next step, we verify objects in each view separately for inconsistencies. Correspondent objects are extracted through a Homography transform from one view to the other and vice versa. Having found the correspondent objects of different views, we partition each object into homogeneous regions. In the last step, we apply the Homography transform to find the region map of first view in the second view and vice versa. For each region (in the main frame and mapped frame) a set of descriptors are extracted to find the best match between two views based on region descriptors similarity. This method is able to deal with multiple objects. Track management issues such as occlusion, appearance and disappearance of objects are resolved using information from all views. This method is capable of tracking rigid and deformable objects and this versatility lets it to be suitable for different application scenarios.

Keywords: Multi-view, Object Tracking, Video Surveillance, Homography Transform

1. INTRODUCTION

Tracking moving objects is a key problem in computer vision and image processing. It is important in a wide variety of applications, like three-dimension (3D) broadcasting, virtual reality, special effects, image composition, human computer interaction (HCI), video surveillance, human motion analysis and traffic monitoring. Automatically monitoring people in crowded environments such as metro stations, city markets, or public parks, has nowadays become feasible for many reasons. First, from the accuracy's point of view, human operators are likely to fail in monitoring crowded and cluttered environments through tens of cameras. Automatic techniques have reached a degree of maturity to be employed at least as a first automatic step to alert human operators, reducing their effort and the sources of distraction. Second, from the economical point of view, the cost of mounting cameras and developing automatic solutions has declined in comparison to the cost of hiring human operators to watch them. Despite of the complexity increase, multiple camera systems exhibit the undoubted advantages of covering wide areas and enhancing the management of occlusions by exploiting the different viewpoints. Single camera tracking is limited in scope of its applications. While suited for certain applications like local environments, even simple surveillance applications demand the use of multiple cameras for two reasons. Firstly, it is not possible for one camera to provide adequate coverage of the environment because of limited field of view (FOV). Secondly, it is desirable to have multiple cameras observing critical areas, to provide robustness against occlusion.

Multiple-cameras provide us with more complete history of an object's actions in an environment. To take advantage of additional cameras, it is necessary to establish correspondence between different views. Thus, we see a parallel between

[*] frederic.dufaux@epfl.ch

the traditional tracking problem in a single camera and that in multiple cameras: tracking in a single camera is essentially a correspondence problem from frame to frame over time. Tracking in multiple cameras, on the other hand, is a correspondence problem between tracks of objects seen from different viewpoints at the same time instant. However, the automatic merging of the knowledge extracted from single cameras is still a challenging task.

Multi view tracking has the obvious advantage over single view tracking because of its wide coverage range. When a scene is viewed from different viewpoints there are often regions which are occluded in some views but visible in other views. A visual tracking system must be able to track objects which are partially or even fully occluded.

The technical problems in multiple camera systems are several and they have been summarized in [1] into six classes: installation, calibration, object matching, switching, data fusion, and occlusion handling.

In this paper, we address the issue of tracking moving objects in an environment covered by multiple un-calibrated cameras with overlapping fields of view, typical of most surveillance setups. Our main objective is to create a framework that can be used to integrate object-tracking information from multiple video sources and resolve the all mentioned technical drawbacks.

1.1. Related work

There are numerous single-camera detection and tracking algorithms, all of which face the same difficulties of tracking 3D objects using only 2D information. These algorithms are challenged by occluding and partially-occluding objects, as well as appearance changes. Some researchers have developed multi-camera detection and tracking algorithms in order to overcome these limitations. Haritaoglu et. al. [2] have developed a system which employs a combination of shape analysis and tracking to locate people and their parts (head, hands, feet, torso etc.) and tracks them using appearance models. In [3], they incorporate stereo information into their system. Kettnaker and Zabih [4] have developed a system for counting the number of people in a multi-camera environment where the cameras have a non-overlapping field of view. By combining visual appearance matching with mutual content constraints between cameras, their system tries to identify which observations from different cameras show the same person. Cai and Aggarwal [5] extend a single-camera tracking system by switching between cameras, trying to always track any given person from the best possible camera - e.g. a camera in which the person is un-occluded. All these systems use background subtraction techniques in order to separate out the foreground and identify objects, and would fail for cluttered scenes with more densely located objects and significant occlusion.

Approaches to multi-camera tracking can be generally classified into three categories: geometry-based, color-based, and hybrid approaches. The first class can be further subdivided into calibrated and un-calibrated approaches. A particularly interesting paper of calibrated approach is reported in [11] in which homography is exploited to solve occlusions. Single camera processing is based on particle filter and on probabilistic tracking based on appearance to detect occlusions. A very relevant example of the un-calibrated approaches is the work of Khan and Shah [15]. Their approach is based on the computation of the so called “Edges of Field of View”, i.e. the lines delimiting the field of view of each camera and, thus, defining the overlapped regions. Through a learning procedure in which a single track moves from one view to another, an automatic procedure computes these edges that are then exploited to keep consistent labels on the objects when they pass from one camera to the adjacent one.

With color-based approaches, the matching algorithm essentially uses of the color of the tracks, In [12] a color space invariant to illumination changes and histogram based information at low (texture) and mid-level are exploited to solve occlusions and match tracks with a modified version of the mean shift algorithm.

Hybrid approaches mix information about the geometry and the calibration with those provided by the visual appearance. These methods use probabilistic information fusion or Bayesian Belief Networks (BBN) [13].

In this paper we propose a new method which can be classified as a hybrid approach. In this technique, first we recover the homography relation between camera views. Homography mapping allows us to find the corresponding regions between different views by use of region descriptors. It is then possible to track objects in 2D/3D simultaneously across multiple viewpoints. In this paper, we consider a surveillance network of two cameras. However the same approach can be generalized to a network of more than two cameras.

2. METHODOLOGY

2.1. General overview of the algorithm

We first present an overview of the algorithm. More precisely this approach consists of a number of blocks. The basic step is to run a single-view tracking algorithm on each of the camera views. It is followed by a consistent object labeling algorithm on all views. Next step is to verify objects of each view separately: objects that are not consistent over time will not be considered in multi-view tracking step.

Correspondent objects are extracted through a homography transform from one view to the other and vice versa. Having found the corresponding objects in different views, each object is partitioned into homogeneous regions. In the last step, a homography transform is applied to find the region map of first view in the second view and vice versa. For each region (in the main frame and mapped frame) a set of descriptors are extracted and the best match of regions between two views is found based on region descriptors similarity. The block diagram of whole system is shown in Fig.1.

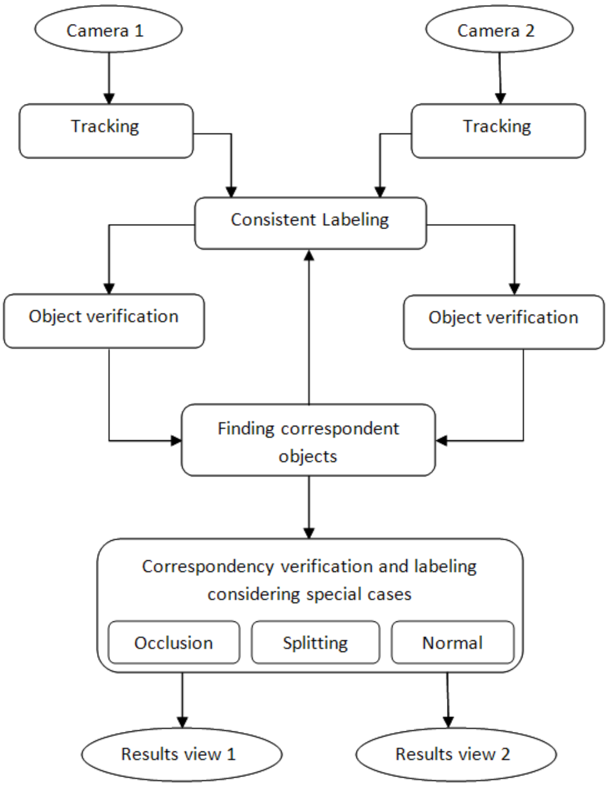


Fig. 1: Block Diagram of Multi-View Object Tracking

2.2. Single-view tracking

Many methods have been developed for object tracking in single view. In this paper, we used a multilevel object-region tracking algorithm [18]. A distinctive feature of the proposed algorithm is that this method operates on region descriptors instead of region themselves. This means that instead of projecting the entire region into the next frame, only region descriptors need to be processed. Therefore, there is no need for computationally expensive models. A brief description of this method is as follow:

1. Foreground object extraction: this decision is taken by thresholding the frame difference between the current frame and the frame representing the background.

2. Object Partition: each object is processed separately and is decomposed into a set of non-overlapping regions to produce the region partition. This step takes into account the spatio-temporal properties of the pixels in the computed object partition and extracts homogeneous regions.
3. Region descriptors: for each region a set of features are extracted as region descriptors. The feature space, used here, is composed of spatial and temporal features. Spatial features are color and a measure of local texture based on variance. The temporal features are the displacement vectors from optical flow computed via block matching. Then each region is represented by a region descriptor.
4. Region Tracking: the first step of tracking regions is the projection of the information of the current frame n into the next frame n+1. Regions of frame n and frame n+1 with most similarity considered as the correspondent objects and receive same labels. (Fig.2)
5. Object Tracking: after finding the corresponding regions between two successive frames, through a top-down and a bottom-up interaction with the region partition step, objects of current frame are validated and are given same labels as previous frame.

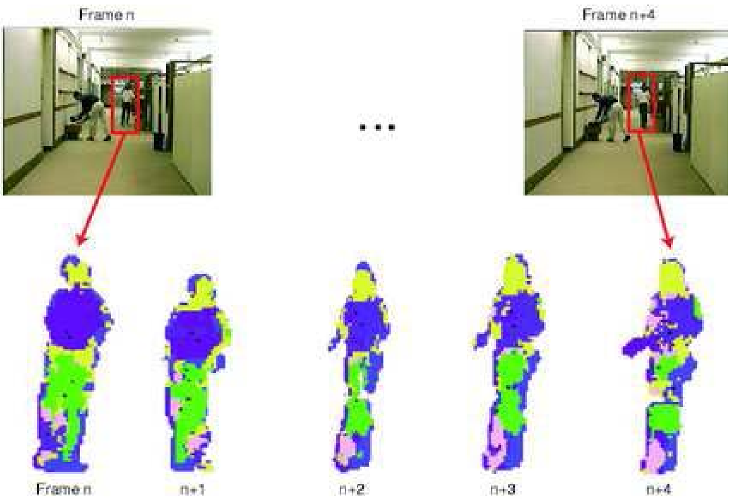


Fig. 2: Object-region extraction and tracking

This algorithm is capable of dealing with multiple simultaneous objects. Track management issues such as appearance and disappearance of objects, splitting and partial occlusion are resolved through interaction between regions and objects. Defining the tracking based on the parts of objects, identified by region segmentation, has led to a flexible technique that exploits the nature of the video object tracking. Fig.3 shows general block diagram of this method.

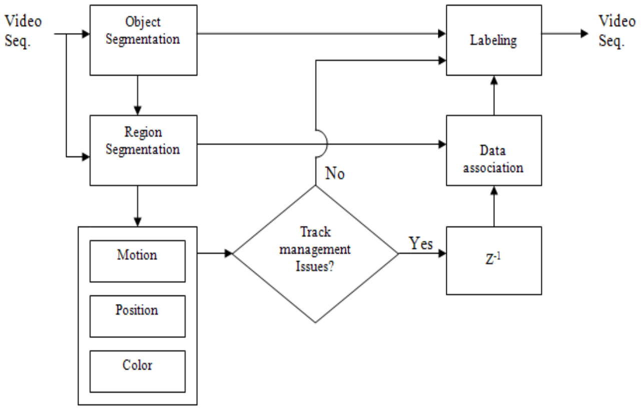


Fig. 3: General block diagram of single view tracking algorithm.

2.3. Consistent object labeling

In a multi-view object tracking scenario, it is essential to establish correspondence between different views of the same object, seen from different cameras, to recover complete information about the object. That is, all views of the same object should be given the same label, as illustrated in Fig. 4. Objects in common are given same labels and other objects receive distinctive labels so that there is no confusion between labels of all views.

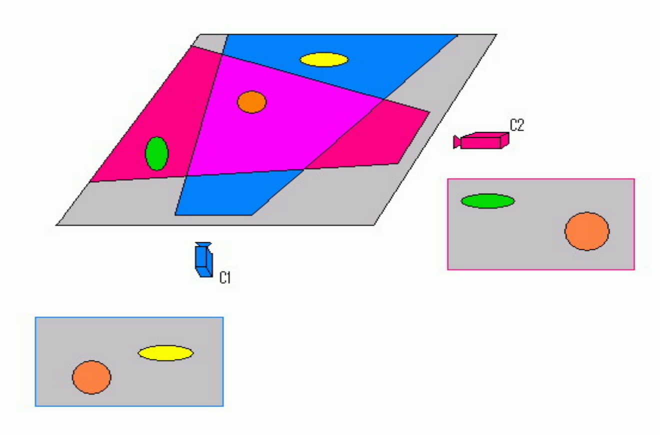


Fig. 4: Two cameras and their footprints are shown. The projections of boundaries of the footprint are also shown in the image that will be observed in two cameras.

2.4. Object consistency verification

Noise or segmentation methods deficiency may erroneously create some objects, called non-semantic objects, in some frames which interfere with the whole tracking system. In order to avoid the effect of such objects, the stability of each object is verified. Namely, this step considers the number of appearance in successive frames and tracking status from one frame to another. After initializing the single-view tracking procedure, a set of tracked objects $O_i^{C_p}(n)$ for each view in frame n are extracted where C_p denotes the camera p. Each object i is characterized by its regions $R_{i,j}(n)$ and the relation between objects and regions of each frame in two successive frames can be expressed as:

$$\forall O_i(n) \quad i=1,\dots,N_O^n \ \ \exists R_{i,j}(n) \quad j=1,\dots,N_{R_i}^n \ , \tag{1}$$

$$\forall O_i(n+1) \quad i=1,\dots,N_O^{n+1} \ \ \exists \tilde{R}_{i,j}(n+1) \quad j=1,\dots,N_{R_i}^{n+1} \ , \tag{2}$$

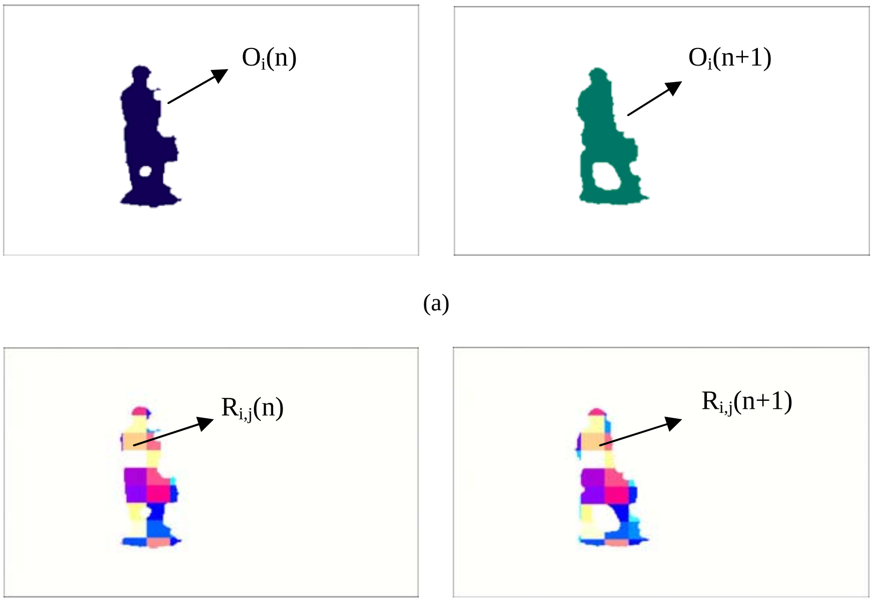
In which N_O^n is number of video objects in frame n, and $N_{R_i}^n$ is number of regions for object i (Fig. 5-a). Based on the employed single-view algorithm, after tracking process, corresponding regions are described as follows:

$$\forall O_i(n), O_k(n+1)\ i = 1,\ldots, N_O^n, k = 1,\ldots, N_O^{n+1} \exists\ CRD_i = \{\left(R_{i,j}(n), \tilde{R}_{k,h}(n+1)\right) |\ R_{i,j}(n) \leftrightarrow \tilde{R}_{k,h}(n+1)\}\ , \quad (3)$$

In which CRD_i is a set of corresponding regions in object i from frame n and object k from frame $n+1$ (Figure 5-b). The stability factor for each object is defined as:

$$SF_i = |CRD_i| / N_{R_i}^n\ . \quad (4)$$

If $SF_i < T_1$ then object i will not be considered in the next step of multi-view object correspondence. By applying this method, objects which appear in one frame and disappear in the next frame are not considered as stable.



(a)

(b)

Fig. 5: (a) left image is object i in frame n and right image is object i in frame $n+1$. (b) Different regions of object in two successive frames (regions with same color are correspondent).

2.5. Objects Correspondence

Before we can jointly track objects between each camera view, it is necessary to recover some camera calibration information. We assume that the camera views are widely separated and moving objects are constrained to move along a dominant ground plane.

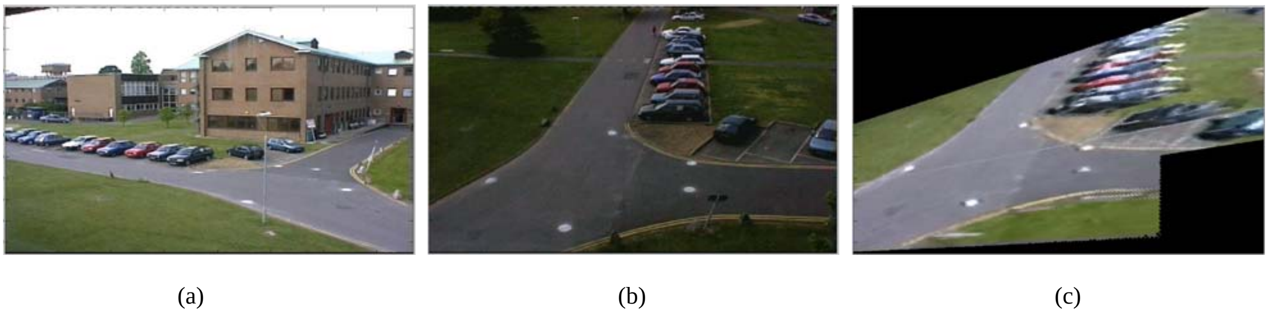
2.5.1. Homography alignment

A homography mapping defines a planar mapping between two overlapping camera views:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y} \qquad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y}\ , \quad (5)$$

where (x, y) and (x', y') are image coordinates for the first and second views, respectively. Hence, each image point correspondence between two camera viewpoints results in two equations in terms of the homography coefficients. Given at least four correspondence points, those coefficients can be estimated. For this purpose we employed Singular Value Decomposition (SVD) for computing the homography [17] using a set of known landmarks as illustrated in Fig. 6.

With the homography matrix, we can find the transform of objects from the first view to the second one as well as from the second view to the first one.



(a) (b) (c)

Fig. 6: (a) First view, (b) Second view, (c) Homography transform of the first view to the second.

2.5.2. Transfer error

The transfer error is the sum of the projection error in each camera view for a corresponding pair. It indicates the size of the error between correspondent objects and their expected projection according to the transfer function. The transfer error associated with a corresponding pair is defined as:

$$TE = (x' - Hx)^2 + (x - H^{-1}x')^2 \ , \tag{6}$$

where x and x' are projective image coordinates in the first and second camera views, respectively.

If $TE < T_2$ then x and x' are considered as a potential match. By considering all possible pairs of objects, we create a list of potential matches, which associates each label of the first view with its corresponding label or labels of second view.

2.6. Correspondence verification

In this step, the same objects receive the same labels in both views. Moreover occlusion or splitting of objects is handled. Two cases may occur. First if each object is just corresponding to one object of the other view straightforwardly it is assigned the same label and it will be tracked later with this new label. Conversely, if an object i in the first view matches with more than one object in the second view, it means that an occlusion has not been handled correctly with the single view information.

To address this problem, we use region level analysis. A set of features like gravity center, histogram and texture, are extracted for each region referred to as region descriptors. It can be expressed as follow:

$$\Phi_{i,j}(n) = (\phi_j^1(n), \phi_j^2(n), ..., \phi_j^n(n))^T \ , \tag{7}$$

where $\Phi_{i,j}(n)$ is the set of features of region j of object i in frame n. The homography transform of each region of object i from first view, $R_{i,j}^{C_1}(n)$ to the second view $\tilde{R}_{i,j}^{C_1}(n)$ and also the homography transform of each region of object k from second view $R_{k,h}^{C_2}(n)$ to the first view $\tilde{R}_{k,h}^{C_2}(n)$ are extracted as follow:

$$\tilde{R}_{i,j}^{C_1}(n)=H\,R_{i,j}^{C_1}(n) \tag{8}$$

$$\tilde{R}_{k,h}^{C_2}(n)=H^{-1}\,R_{k,h}^{C_2}(n)\quad , \tag{9}$$

in which H is the Homography matrix. The distance between regions of two views is calculated as follow:

$$D_{(i,j),(k,h)}^{C_1\to C_2}=\left|\tilde{R}_{i,j}^{C_1}-R_{h,k}^{C_2}\right| \tag{10}$$

$$D_{(h,k),(i,j)}^{C_2\to C_1}=\left|R_{i,j}^{C_1}-\tilde{R}_{h,k}^{C_2}\right| \tag{11}$$

So the total distance between each pair of regions is:

$$D_{(i,j),(k,h)}=D_{(i,j),(k,h)}^{C_1\to C_2}+D_{(k,h),(i,j)}^{C_2\to C_1} \tag{12}$$

Pairs of regions whose distances are above a given threshold, T_3, are discarded from the optimization process that follows. The best set of corresponding regions is obtained through applying the minimum mean square error (MMSE) method as follows:

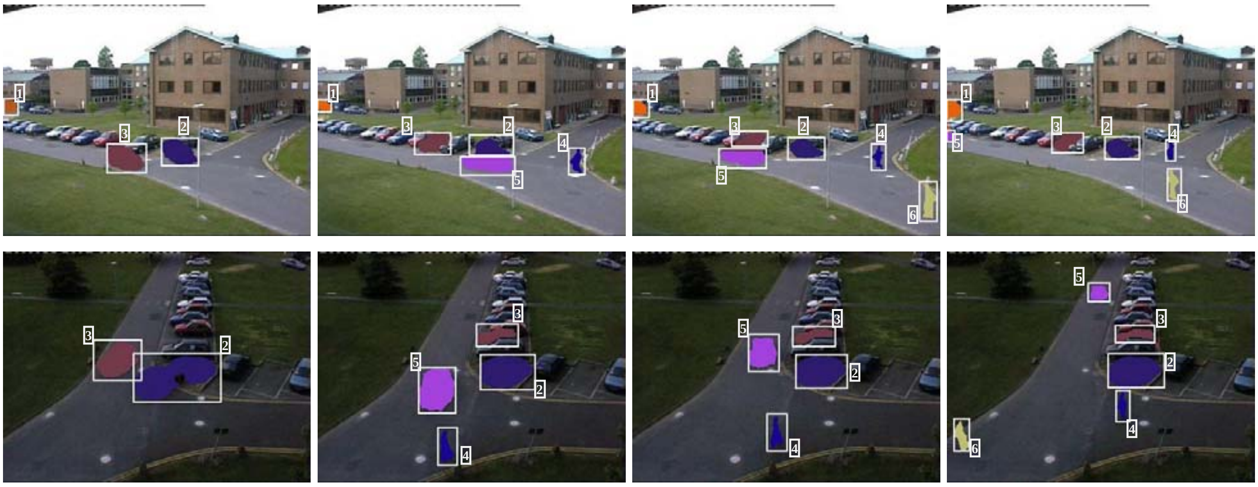
$$MR=\left\{(R_{i,j}^{C_1},R_{h,k}^{C_2})\right\}=\arg\min_{MR} E\{(R_{i,j}^{C_1}-R_{h,k}^{C_2})^2\} \tag{13}$$

in which MR is the set of corresponding regions. Furthermore, by utilizing top-down and bottom-up method, we assign object labels to each region. If some regions are not assigned to any object label during this procedure, they are given the label of the closest region.

3. RESULTS AND INTERPRETATION

The performance of the method described in this paper is evaluated using the first two PETS2001 data sets. We tried to modify the information of first camera based on second camera information. As explained, this helps us to handle the occlusion problem.

Here, the results of the algorithm for two sequences are shown. Fig. 7 displays a sequence which suffers from occlusions. In this Figure, top row represents frames of the first view and bottom row represents frames of the second view. Each column shows the same time instant from two views. In this sequence, from the first frame to frame #33 two objects are occluded in both views and these objects are presented by label #2 in fig.7-a, but as soon as these two objects are seen individually in the second view, the algorithm is able to correctly label them in the first view. In the Fig.7-b, (frame #41) it can be observed that occluded objects of first view are represented by two different labels base on the second view object splitting. A similar problem also occurs in frame #61 and the algorithm can follow the object in both views correctly.



(a) Frame#1 (b) Frame#41 (c) Frame#61 (d) Frame#101

Fig. 7: Top row is the first view and bottom row is the second view at the same time.

The second sequence is showed in Fig. 8. A pedestrian is crossing the junction and in the first view, it has been occluded by a car. Based on the second view information, it is possible to track both pedestrian and car in both views correctly.

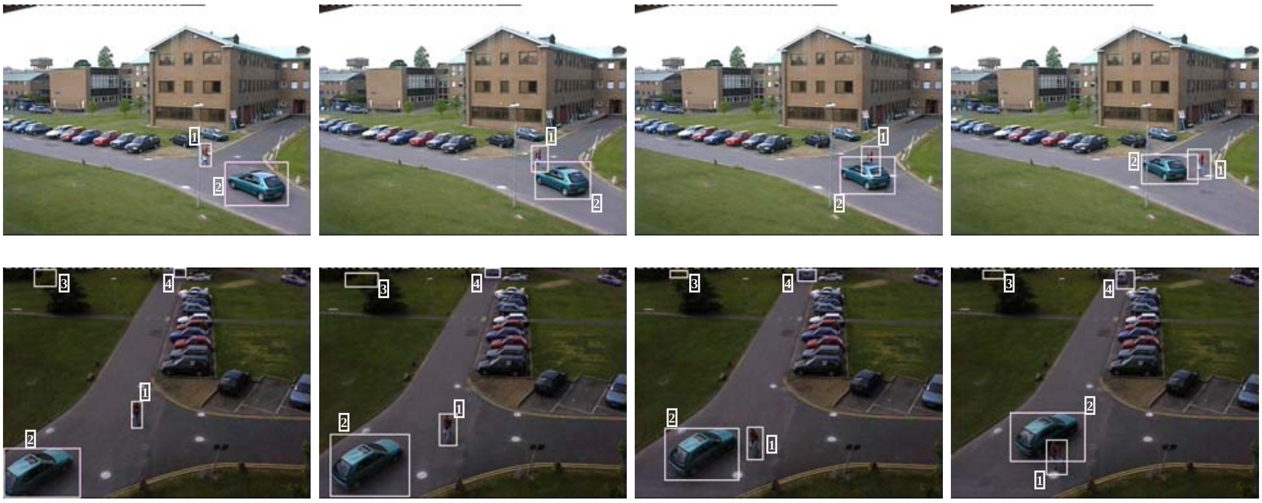


Fig. 8: Top row is the first view and bottom row is the second view at the same time.

To conclude this section, it is interesting to enumerate the advantages and disadvantages of the proposed algorithm. The proposed method based on Homography transform has the following advantages:

- It is fast enough to be used for online scenario on a standard PC (around 1 sec for each pair of 384×288 frames in MATLAB)
- It can handle the occlusion problem if objects are visible at least in one view
- It is possible to extend the method for more than two views
- Using the region descriptors to track multi-view objects is a novelty in multi-view object tracking

And the disadvantages are:

- If a group of objects enters the scene and then they get separated while they are occluded with another object/objects in the scene, the results will not be reliable
- Final results are rather dependent on single-view algorithm and if there is a tremendous error in single-tracking of both views, the algorithm might fail.

Region descriptors and homography transform provide an estimation of regions from one view to another, which helps to handle the occlusion and splitting issues and to refine the individual objects in a scenario with occlusion.

4. CONCLUSION AND FUTURE WORK

This paper presents an automatic multi-view object tracking algorithm based on interactions between object regions in one view and homography transform of object regions of other views. This method is able to deal with multiple simultaneous objects. Track management issues such as occlusion, splitting, appearance and disappearance of objects are resolved using information of other views.

This method is capable of tracking rigid and deformable objects and this versatility makes it suitable for different scenarios. All the component of this algorithm, included the single view tracking algorithm, can be run in real-time applications. We are currently investigating the method for more than two views to fuse the information of different views to get the most efficiency.

ACKNOWLEDGMENT

This work was partially supported by European Network of Excellence VISNET II (http://www.visnet-noe.org) IST Contract 1-038398, funded under the European Commission IST 6th Framework Program.

REFERENCES

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, “A Survey on Visual Surveillance of Object Motion and Behaviours”, IEEE Trans. on Systems, Man, and Cybernetics – Part C, 34(3), p. 334-352, 2004.

[2] I. Haritaoglu, D. Harwood, and L. Davis, “W4: Who, When, Where, What: A Real Time System for Detecting and Tracking People”, 3rd Int. Conf. on Automatic Face and Gesture, 1998.

[3] I. Haritaoglu, D. Harwood, and L. Davis, “W4: A Real-time System for Detecting and Tracking People in 2 1/2D”, 5th European Conf. on Computer Vision, 1998.

[4] V. Kettnaker, R. Zabin, “Counting People from Multiple Cameras”, Proc. of IEEE ICMCS, p.267-271, 1998.

[5] Q. Cai, and J.K. Aggareal, “Automatic Tracking of Human Motion in Indoor Scenes across Multiple Synchronized Video Streams”, Proc. Of 6th Int. Conf. on Computer Vision, p 356-362, 1998.

[6] J. Li, C.S. Chua, and Y.K. Ho, “Colour Based Multiple People Tracking”, Proc. Of IEEE Int. Conf. on Control, Automation, Robotics and Vision, 1, p.309-314, 2002.

[7] J. Black, T.J. Ellis, and P. Rosin, “Multi-View Image Surveillance and Tracking”, Proc. IEEE Workshop Motion and Video Computing 2002.

[8] N. Nguyen, H. Bui, S. Venkatesh, and G. West, “Multiple Camera Coordination in a surveillance system”, ACTA Automatic Sinica, Vol 23(3), p408-422, 2003.

[9] J. Black, T. Ellis, ‘Multi Camera Image Tracking”, Image and Vision Computing (24), No. 11, p 1256-1267, 2006.

[10] S. Iwase and H. Saito, “Tracking Soccer Players based on Homography among Multiple Views”, Proc. SPIE2003, v. 5150, p283-292, 2003.

[11] Z. Yue, S. Zhou, and R. Chellappa, “Robust two-camera Tracking using Homography”, Proc. IEEE Intl Conf. on Acoustics, Speech, and Signal Processing, vol. 3, p 1-4, 2004.

[12] R. Cucchiara, A. Prati, R. Vezzani, “Posture Classification in a Multi-camera Indoor Environment”, Proc. IEEE Intl Conf. on Image Processing(ICIP), vol. 1, p. 725-728, 2005.

[13] S. Calderara, R. Vezzani, A. Prati, and R. Cucchiara, “Entry Edge of Field of View for Multi-camera Tracking in Distributed Video Surveillance”, Proc. IEEE Intl Conf. on Advanced Video and Signal-based Surveillance (AVSS’05), p. 93-98, 2005.

[14] I. Paek, C. Park, M. Ki, K. Park, and J. Paik, “Multiple_view object tracking using Metadata”, Proc. Int. Conf. ICWAPR, vol. 1, no. 1, p 12-17,2007.

[15] S. Khan, and M. Shan, “Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping fields of View”, IEEE Trans. On PAMI, 25(10), p. 1355-1360, 2003.

[16] A. Mittal, and L. Davis, “Unified Multi-camera detection and Tracking using Region_Matching”, Proc. Of IEEE Worckshop on Multi-Object Tracking, p. 3-10, 2004.

[17] R. Hartley, and A. Zisserman, “Multiple View Geometry in Computer Vision”, Cambridge University Press, 1998.

[18] A. Cavallaro, O. Steiger, and T. Ebrahimi, “Tracking Video Objects in Cluttereds Background”, IEEE Trans. on Circuit and Systems for Video Technology, 2005.

[19] A. Yilmaz, and M. Shan, “Object Tracking: A Survey”, ACM Computing Surveys (CSUR), vol. 38, 2006