# Continuous Multi-Views Tracking using Tensor Voting

Jinman Kang, Isaac Cohen and Gerard Medioni
*Institute for Robotics and Intelligent Systems*
*University of Southern California Los Angeles, CA 90089-0273.*
*{jinmanka|icohen|medioni}@iris.usc.edu*

## Abstract

*This paper presents a new approach for continuous tracking of moving objects observed by multiple fixed cameras. The continuous tracking of moving objects in each view is realized using a Tensor Voting based approach. We infer objects trajectories by performing a perceptual grouping in 2D+t using Tensor Voting. Also, a multi-scale approach bridging gaps in object trajectories is presented. The trajectories obtained from the multiple cameras are registered in space and time allowing a synchronization of the video streams and a continuous tracking of objects across multiple views. We demonstrate the performance of the system on several real video surveillance sequences.*

## 1. Introduction

Video surveillance of large facilities usually requires a set of cameras for ensuring a complete coverage of the scene. While visual tracking from a single camera has been extensively studied in computer vision and related fields very little attention was given to tracking objects across a heterogeneous network of cameras. Tracking objects across multiple heterogeneous cameras is a challenging task, as it requires a space and time registration of the trajectories recovered from each camera.

Several multi-camera tracking algorithms have been developed recently and most of them use "color distribution" as a cue for tracking across the views. Various approaches were proposed such as matching color histogram of the blobs across the views [4], switching to another cameras when the tracked blob is no longer visible from the current view [5], constructing blobs in 3D space using short-base line stereo matching with multiple stereo cameras [6], or using volume intersection [7]. Since the "color" information can be easily biased by several factors such as illumination, shadow, blobs segmentation, or different camera controls, color cue is not very reliable for tracking moving objects across large scenes such as halls or campuses. The use of such methods is limited to the case of synchronized cameras. In [8], author proposed an approach for space and time self-calibration of cameras, but the proposed approach is limited to small moving objects where the observed shape is similar across views.

In this paper, we describe a novel approach to continuously track moving regions across cameras using the graph-based description and the Tensor Voting methodology for enforcing the smoothness constraint and tracks completion. The camera views and the trajectories are registered using a perspective transformation allowing us to provide space and time registration of objects trajectories across views.

Our proposed method uses the graph-based tracking for obtaining initial tracking description from detection. The description includes the center positions of moving regions and relationship between two frames. Our method adopts Tensor Voting approach for dealing with these problems. It solves the false detection problem, and fragmentation and ambiguity problems of trajectories at the same time. We also propose a multi-scale approach for merging the fragmented trajectories efficiently and therefore achieving continuous tracking of moving objects. This efficient tracking approach is applied to the problem of tracking objects across a network of heterogeneous cameras. The integration of trajectories across multiple views requires a registration in the 2D+t space. We propose an approach based on the use of perspective registration of the multiple-views and the trajectories in space and time.

The paper is organized as follows. Section 2 introduces the continuous tracking using tensor-voting formalism. Section 3 presents the proposed approach for space-time registration of trajectories across a network of heterogeneous cameras. Section 4 presents some result obtained from real sequences. Finally, in section 5 we conclude our paper with future work.

## 2. Continuous Tracking using Tensor Voting

Tracking using the graph-based representation of moving objects was proposed in [1]. It preserves both spatial and temporal information. The obtained trajectories are often fragmented due to occlusions and false detection. Furthermore, the lack of a model for object trajectories prevents us from merging the fragmented tracks. The graph representation provides an exhaustive description of the

regions where a motion was detected and the way these regions evolve in time. However, more edges than necessary are created and the lack of trajectory model prevents us from merging sub-tracks of the same object into a single trajectory. In [2] a Tensor Voting based tracking approach was proposed, providing an efficient way for handling discontinuous trajectories but does not allow an online processing. Indeed the proposed method requires a complete detection of moving object in the image sequence prior to the extraction of the trajectories. Also the proposed method could not merge sub-trajectories too far apart.We propose to adapt the tensor voting-based tracking proposed in [2] for achieving an efficient continuous tracking method. We reformulate the tracking problem as a problem of extracting salient curves in 2D+t space.

The graph representation provides a cloud of points in 2D+t with an estimate of the motion associated to each moving object. We propose in this paper to perform a perceptual grouping in the 2D+t space using Tensor Voting for extracting object's trajectory as a salient structure in 2D+t.

## 2.1. Overview of tracking by Tensor Voting

The two main elements of the Tensor Voting theory are:

**(A) Tensor-based data representation:** We use tensors to encode simultaneously data and the corresponding uncertainties. In the case of a single point in 2D+t space, no specific orientation is known a priori and therefore we encode this uncertainty uniformly along all directions by considering a ***ball tensor***. If an orientation (or motion) is available, we use the following ***stick tensor*** aligned along the given motion vector $v = (v_x, v_y, 1)$:

$$v^T v = \begin{bmatrix} v_x^2 & v_x v_y & v_x \\ v_x v_y & v_v^2 & v_y \\ v_x & v_y & 1 \end{bmatrix} \quad (1)$$

This stick tensor allows the encoding of the orientation and its uncertainty. In Figure 1 we illustrate the tensor description and tensor field in 2D+t space as proposed by [2]. The shape of the ellipsoid encodes the uncertainty. The stick tensor describes the point location and velocity simultaneously.

**(B) Data propagation:** Based on the uncertainties of a given tensor, the information of each location is propagated to the neighbouring points using a voting scheme. Every point propagates its tensor representation to its neighbouring points. The tensor field depicted in Figure 1.b defines the shape of the neighbourhood. The scale defines its size. Every point collects votes from its neighbouring points by combining their tensors. This data propagation scheme allows collecting local properties and representing the local structure through the combination of the tensors. The resulting tensor is characterized by three positive eigenvalues $\lambda_{\max} \geq \lambda_{mid} \geq \lambda_{\min} \geq 0$. Its "shape" is then the representative of the collected votes. More precisely, the eigenvector associated with $\lambda_{\max}$ is the privileged orientation and $\lambda_{\max} - \lambda_{mid}$, is the associated saliency; it indicates the confidence in this direction.

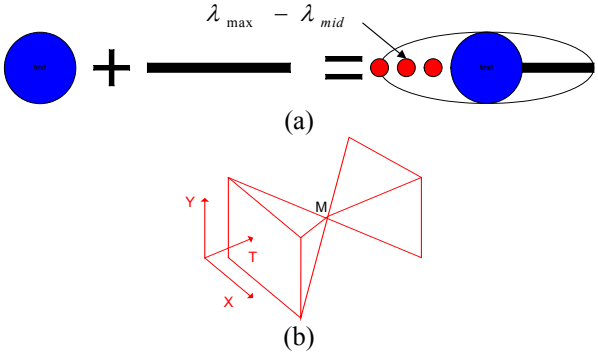



***Figure 1. 2D+t Tensorial description and Tensor field; (a) tensorial description, (b) tensor field***

The key point in the tensor voting formalism is the tensor field. Its role is to enforce the continuity properties such as curvature consistency. The 2D+t case is a neither 2D nor 3D inference system. Since the 2D+t space is not isotropic, the tensor fields are different from the generic ones used in the 3D case [3].

The classical steps of a tensor voting approach can be summarized as follow: We first estimate the tangent and saliency at each input token. Then, we densify the information creating a dense tensor distribution by voting everywhere. Finally, curves and junctions are extracted from the saliency map.

The tensor voting framework described in [2] allows us to disambiguate multiple hypotheses, infer direction and velocities of the objects, and bridge gaps in object trajectories using a smoothness constraint. We reformulate tracking problem as a problem of extracting salient curves in 2D+t. The output is a set of trajectories with corrected bounding boxes of moving objects. It provides an efficient way for handling discontinuous trajectories. However, the proposed method is only applicable to an off-line computation since the voting procedure can be performed only after processing all available frames. Also, the ability of the proposed method to merge sub-tracks into a single trajectory is dependent on a correct choice of the size of the voting field. The method fail to bridge gaps in object trajectories when this parameter is not chosen properly.

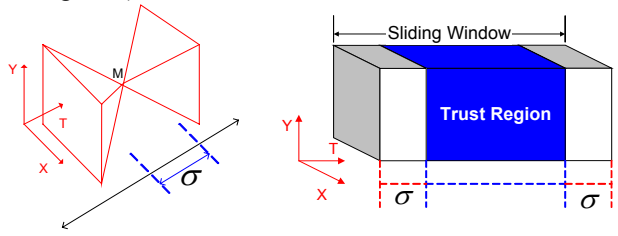
In the following we address these issues and provide an approach for continuous object tracking.

## 2.2. Merging tracks using a multi-scale approach

In order to perform a continuous tracking from a live input we use a sliding window method and multi-scale approach to achieve continuous tracking. The difference between using batch and sliding window mode is the correction and normalization of the bounding boxes. In batch

mode case, bounding boxes are adjusted by the size of the most salient feature in the batch. On the other hand in sliding window mode case, the search region is limited to only the sliding window, and therefore, the most salient feature in the sliding window adjusts the box size.

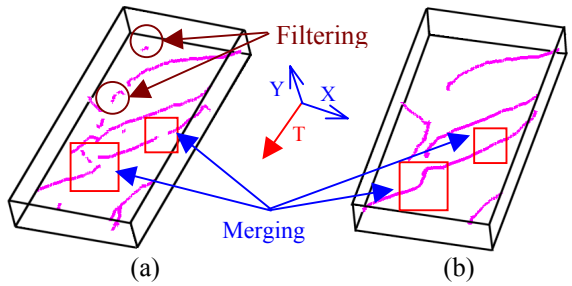
Tensor Voting is dependent on the size of the field used for propagating the local information. Given a domain size, neighbouring tensors are selected for voting. Therefore, the influence of the voting process is limited by the size of voting field ( $\sigma$ ). As a result, the centre region of a processing sliding window is selected as trust region since boundary regions do not collect all the votes in the domain (see Figure 2).




(a) (b)

***Figure 2. Domain influence and trust region; (a) Limitation of domain influence by σ (b) Trust region in one sliding window***

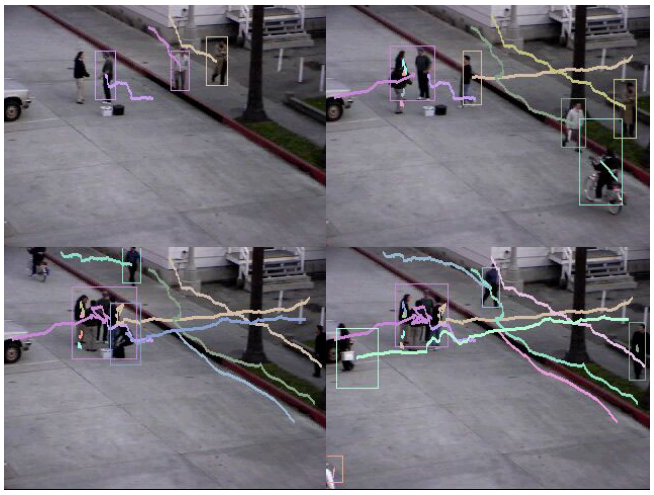
The ability to merge detected tracks depends on $\sigma$ since it characterizes the influence domain. However, a large $\sigma$ implies a small trust region for a fixed size sliding window and a higher computational complexity. We propose a multi-scale approach, which uses multi-scale voting fields for merging sub-tracks into the global trajectory. We search for an ending point of each trajectory segment and possible fragmented trajectories based on the Euclidean distance in 2D+t space. Given a set of tracks and isolated points, provided by the graph-based representation, we apply the tensor voting with a small $\sigma$ for extracting salient curves in the 2D+t space. This local tensor voting does not allow merging sub-trajectories: It generates smooth fragments of the tracks. Our approach consists in merging these sub-tracks by performing a perceptual grouping at various scales in the 2D+t space. The process focuses on searching for potential completion for each ending/starting points of the computed sub-tracks.



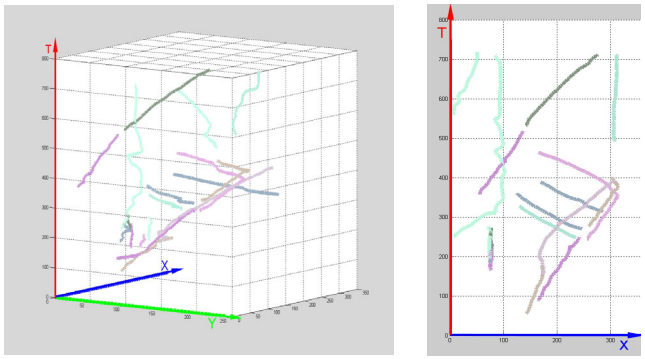

(a) (b)

***Figure 3. Using multi-scale σ (a) σ = 10, (b) σ = 20***

Given a set of trajectories in the 2D+t space, we characterize the potential locations for curves merging as being the ending/starting points within a specified neighbourhood (see Figure 3).



(a)




(b)

***Figure 4. Tracking example (800 frames) (a) Real Image with trajectory overlap. (b) 2D+t representation of the trajectories from 3 different view***

This approach allows performing trajectory completion in the 2D+t space and provides an efficient solution for merging fragmented tracks due to occlusions or bad detections. The variation of the scales and the range of $\sigma$ are depending on the size of the sliding window. The choice of $\sigma$ should be smaller than the half size of the given sliding window in order to ensure a large data propagation for the tensor voting formalism. However, the minimum value depends on the temporal distribution of the input data. In Figure 4 we show an example of objects tracking and trajectory completion using the presented approach. We have used two scales, $\sigma$ = 10 and $\sigma$ = 20 for a window size of 40 frames. The processing time is 5-7 frames/sec for 640 by 480 resolution image.

## 3. Continuous Tracking Across Cameras

Multiple cameras are usually required for monitoring large scenes such as halls or campuses. The set of cameras provide a good coverage of the scene but the integration of the information extracted from each camera still requires to address issues such as: tracking the same object from multiple cameras or tracking an object across different cameras. While geometric registration is performed offline and allows to map the relative position of the cameras with regard to the scene, object trajectories cannot usually be mapped across the views. Indeed, object's trajectory is characterized by the centroid's space-time evolution of the detected moving object. The locations of the centroid across multiple cameras do not correspond to the same physical 3D points and therefore cannot be registered using the same perspective transform used for registering the cameras locations. Furthermore, the cameras are not necessarily synchronized and may have different shutter speed, requiring a synchronization of the computed trajectories prior to their integration.

The integration of trajectories across multiple views requires a registration step in 2D+t space. We propose an approach based on the use of perspective registration of the multiple-views and the trajectories in 2D+t space.

### 3.1. Geometric Registration

The geometric registration of cameras viewpoint is performed using a perspective transform from a set of 4 matching points. The perspective parameters correspond to a null space of the matrix A (given in equation (2)) and are estimated using SVD decomposition of A. In Figure 5, we show the registration of two views.

$$AH=\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 & -y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 & -x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 & -y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 & -x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 & -y'_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

If we denote a homography from view 2 to 1, we can register multiple cameras by series of concatenated homography as like (3).

$$H_m^n = H_{n+1}^n H_{n+2}^{n+1} \cdots H_{m-1}^{m-2} H_m^{m-1} \quad (3)$$

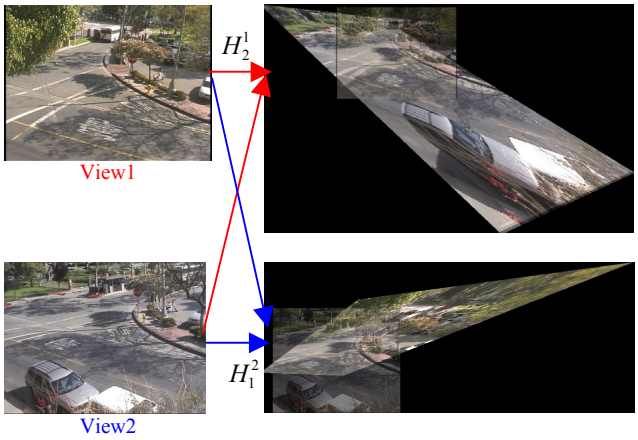


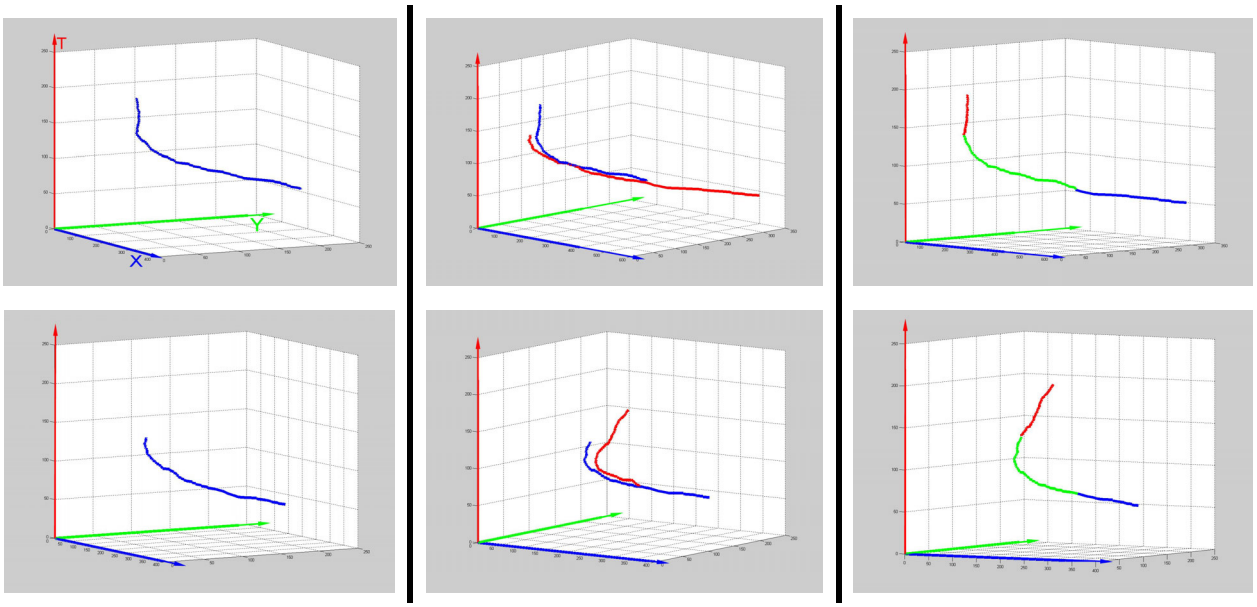

***Figure 5. Camera registration using two views***

### 3.2. Stream Synchronization

The integration of trajectories obtained from various sensors requires a synchronization of the acquired video streams. The estimation of the time offset is not sufficient for synchronizing the video streams since the cameras can have different shutter speeds therefore requiring a registration of the trajectories in space and time.

There are several ways for the registration of observed trajectories. If we assume that each trajectory corresponds the same 3D physical points through all views and the moving object moves above the ground plane with same height, the homography using the ground plane is sufficient for registering trajectories. In this case, the synchronization can be done using minimization of the distance between trajectories for each frame by shifting the trajectory along the time axis. If the centroids correspond to the same 3D physical point, but the 3D physical point is not the same across the views, the homography obtained from the ground plane is not sufficient for the registering the trajectories. In this case, we need calibration information for each cameras and the epipolar geometry between the considered views. Using the fundamental matrix, the synchronization can be done by finding the intersection of the trajectory point of one view and epipolar line calculated by the fundamental matrix from the other view since these points correspond to the same 3D points. Once the synchronization is done, we can correct the height by calculating 3D position of matched trajectory points using the calibration information. Then, the homography from the ground plane is sufficient for the registration since the trajectories are already synchronized.

In most cases, the centroids describing the object trajectories do not correspond to the same 3D point. In this paper we propose a simple approach to solve space and time registration by using the homography obtained from ground plane and trajectories. First, we register each view and trajectories using the homography obtain from ground



(a) (b) (c)

***Figure 6. Example of trajectory registration (a) Tracking result by Tensor Voting. (b) Trajectory registration using the ground plane homography. (c) Refinement by the homography obtaining from the trajectories correspondence.***

plane by (2). Since the trajectories do not correspond the same 3D point and they are not synchronized, the misalignment of the registered trajectories can be observed (see Figure 6.b). These misalignments are reduced using another homography obtained from the trajectories correspondences. Here we propose to use a MLE approach. We obtain a homography by arbitrary selecting 4 points of each trajectory, then calculate the errors between base trajectory and transferred trajectory. We select the homography that minimizes the trajectory registration errors (see Figure 6.c). The proposed two-step approach allows us to register trajectories acquired by heterogeneous cameras using the spatio-temporal registration.

## 4. Experimental Results

We present in this section some results obtained on real sequences for illustrating the multi-views registration of cameras and trajectories.

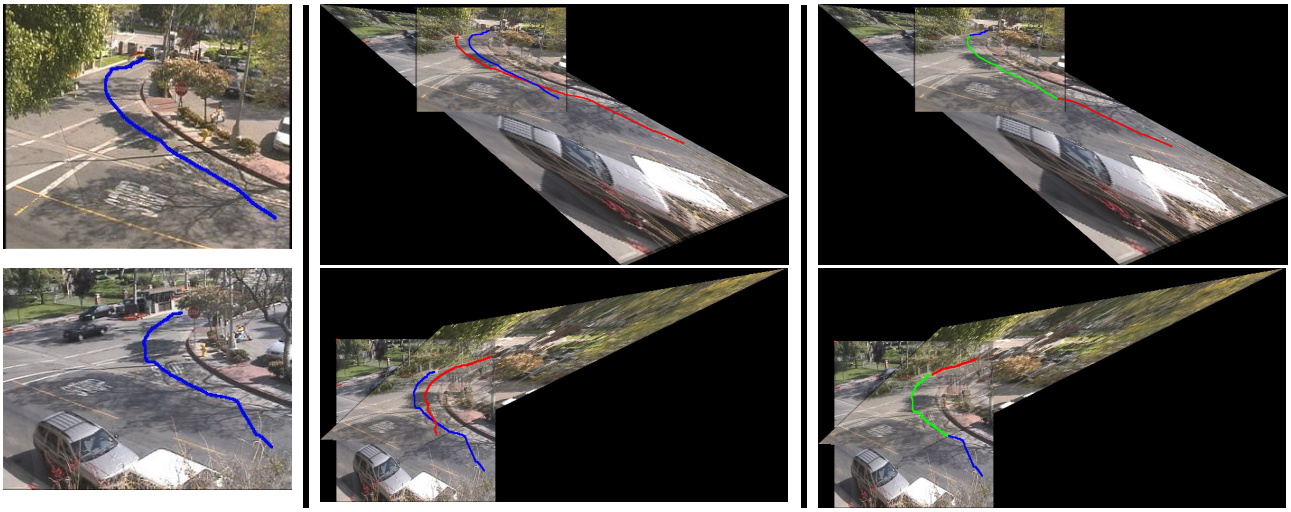
In Figure 7.a and 8.a, we show the trajectories computed on each video stream using the tensor-voting technique presented in Section 2. In these examples, one can observe that the trajectory of each moving object is very smooth and continuous. The registration of the trajectories using the ground plan is presented in Figure 7.b and 8.b. As expected, the trajectories are not properly aligned in space and time. Finally, in Figure 7.c and 8.c. we illustrate the proposed the spatio-temporal registration technique. A blue line represents an original trajectory belonging to the reference view. A red line represents a registered trajectory transferred from other view and a green line represents an overlapping trajectory between the two views.

## 5. Conclusion

We have presented a novel approach for continuous tracking of moving regions accross multiple views. As proposed in Section 3, a spatio-temporal homography allows to register multiple tracks and allows to track moving objects across multiple cameras. For each view, the combination of the sliding batch window approach and multi-scale approach allows us to track efficiently the moving objects continuously. The multi-scale approach with sliding window method provides an efficient way to solve the continuity problem in Tensor Voting methodology. This multi-scale approach can be used not only for merging trajectories, but also for other temporal related Tensor Voting problems such as feature inference in 3D from video streams. Several issues are still to be addressed such as cameras with different shutter speeds.

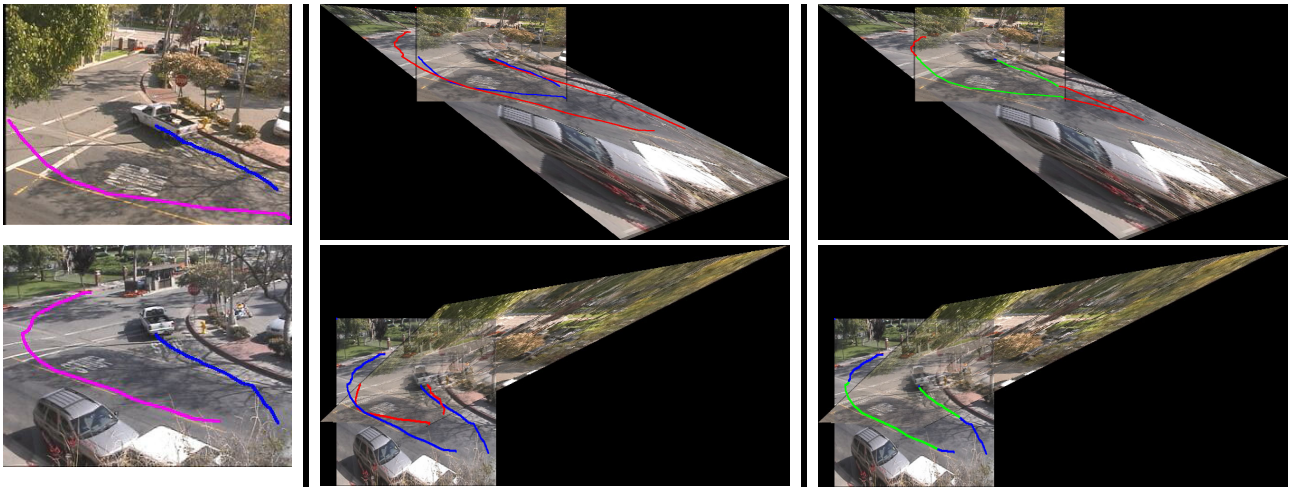
## Acknowledgements

This research was supported by the Advanced Research and Development Activity of the U.S. Government under contract No. MDA-908-00-C-0036.



(a) (b) (c)

***Figure 7. Multiple View Registration (a single moving object) (a) Tracking result by Tensor Voting. (b) Ground plane registration. (c) Trajectory registration.***



(a) (b) (c)

***Figure 8. Multiple View Registration (multiple moving object) (a) Tracking result by Tensor Voting. (b) Ground plane registration. (c) Trajectory registration.***

## References


[1] I. Cohen and G. Medioni, "Detecting and Tracking Moving Objects for Video-Surveillance", *IEEE CVPR*, Vol 2, pp. 319-325, 1999.

[2] P. Kornprobst and G. Medioni, "Tracking Segmented Objects using Tensor Voting", *IEEE CVPR*, Vol. 2, pp. 118-125, 2000.

[3] G. Medioni, M.S. Lee and C.K. Tang, *A Computational Framework for Segmentation and Grouping*, Elsevier, Dec. 1999.

[4] J. Orwell, P. Remagnino, and G.A. Jones, "Multi-Camera Color Tracking", *proc. of the 2nd IEEE Workshop on Visual Surveillance*, 1999.

[5] Q. Cai and J.K. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized video Streams", *ICCV*, 1998.

[6] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer, "Multi-camera Multi-person Tracking for EasyLiving", *3rd IEEE IWVS*, 2000.

[7] A. Mittal and L. Davis, "M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene Using Region-Based Stereo", *ECCV 2002*, 2002.

[8] G. Stein, "Tracking from Multiple View Points: Self-calibration of Space and Time", *IEEE CVPR*, pp. 521-527, 1999.