

Detecting and Tracking Moving Objects for Video Surveillance

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

Abstract

We address the problem of detection and tracking of moving objects in a video stream obtained from a moving airborne platform. The proposed method relies on a graph representation of moving objects which allows to derive and maintain a dynamic template of each moving object by enforcing their temporal coherence. This inferred template along with the graph representation used in our approach allows us to characterize objects trajectories as an optimal path in a graph. The proposed tracker allows to deal with partial occlusions, stop and go motion in very challenging situations. We demonstrate results on a number of different real sequences. We then define an evaluation methodology to quantify our results and show how tracking overcome detection errors.

1 Introduction

The increasing use of video sensors, with Pan-Tilt and Zoom capabilities or mounted on moving platforms in surveillance applications, have increased researchers attention on processing arbitrary video streams. The processing of a video stream for characterizing events of interest relies on the detection, in each frame, of the objects involved, and the temporal integration of this frame based information to model simple and complex behaviors. This high level description of a video stream relies on accurate detection and tracking of the moving objects, and on the relationship of their trajectories to the scene.

In this paper, we address the problem of detecting and tracking moving objects in the context of video surveillance. Most of the techniques used for this problem deal with a stationary camera [4, 3] or closed world representations [8, 6] which rely on a fixed background or a specific knowledge on the type of actions taking place. We deal with a more

challenging type of video streams: the one obtained from a moving airborne platform. This more general case allows us to evaluate the proposed approach for processing video streams acquired in real world video surveillance situations.

We propose an approach which relies on a *graph representation* of detected moving regions for deriving a robust tracker. The detection phase performed after the compensation of the image flow induced by the motion of observation platform produces a large number of regions. Indeed, the use residual flow field and its normal component, *i.e. normal flow*, to locate moving regions also detects the registration errors due to local changes not correctly handled by the stabilization as well as 3D structures *i.e. parallax*. Defining an attributed graph where each node is a detected region and each edge is a possible match between two regions detected at two different frames, provides an exhaustive representation of all detected moving objects. This graph representation allows us to maintain a *dynamic template* of all moving objects which is used for their tracking. Moreover, the graph is used to characterize objects trajectories through an *optimal search path* along each graph's connected component.

The paper is organized as follows; we first describe in section 2 the detection technique used. The graph representation and the dynamic template inference are described respectively in sections 3 and 4. Section 5 presents the method used for deriving objects trajectories from the associated graph. Finally, in section 6 we describe the evaluation technique used for quantifying the results obtained on the set of processed videos.

2 Detection of Moving Objects

Most available techniques for detecting moving objects have been designed for scenes acquired by a stationary camera. These methods allow to segment each image into a set

of regions representing the moving objects by using a background differencing algorithm [6, 4]. More recently, [3] have proposed a local modeling of the background using a mixture of K-Gaussian allowing to process video streams with time varying background. These methods give satisfactory results and can be implemented for real time processing without dedicated hardware.

The availability of video sensors, at low cost, with Pan-Tilt and Zoom capabilities or video streams acquired by moving platforms, have focused the attention of researchers on the detection of moving objects in a video streams acquired by a moving platform. In this case, the background differencing techniques cannot be used. They have to rely on a stabilization algorithm in order to cancel the camera motion. Such a two-step technique, *i.e.* stabilization and detection, does not perform perfectly since the detection techniques based on background differencing assume a perfect stabilization. Indeed, stabilization algorithms use an affine or perspective model for motion compensation and the quality of the compensation depends on the observed scene and on the type of acquisition (*i.e.* Pan-Tilt-Zoom, arbitrary motion...). Therefore, the motion compensation is not error free and induces false detection. However, one can use the temporal coherence of the detected regions in order to increase the accuracy of the moving object detection [10].

Instead of using this two-step approach, we propose to integrate the detection into the stabilization algorithm by locating regions of image where a residual motion occurs. These regions are detected using the normal component of the optical flow field.

Normal flow is derived from image spatio-temporal gradients of the stabilized image sequence. Each frame of this image sequence is obtained by mapping the original frame to the selected reference frame. Indeed, let \mathcal{T}_{ij} denote the warping of the image i to the reference frame j . The mapping function is defined by the following equation:

$$\mathcal{T}_{ij} = \prod_{k=i, \dots, j+1} \mathcal{T}_{k, k-1} \quad (1)$$

and the stabilized image sequence is defined by $\mathcal{I}_i = I_i(\mathcal{T}_{ij})$. The estimation of the mapping function amounts to estimate the egomotion, based on the camera model which relates 3D points to their projection in the image plane. The approach we use, models the image induced flow instead of the 3D parameters of the general perspective transform [7]. The parameters of the model are estimated by tracking a small set of feature points (x_i, y_i) in the sequence. Given a reference image I_0 and a target image I_1 , image stabilization consists of registering the two images and computing the geometric

transformation \mathcal{T} that warps the image I_1 such that it aligns with the reference image I_0 . The parameter estimation of the geometric transform \mathcal{T} is done by minimizing the least square criterion:

$$E = \sum_i \{I_0(x_i, y_i) - I_1(\mathcal{T}(x_i, y_i))\}^2 \quad (2)$$

where outliers are detected and removed through an iterative process. We choose an affine model, which approximates well the general perspective projection, while having a low numerical complexity. Furthermore, a spatial hierarchy, in the form of a pyramid, is used to track selected feature points. The pyramid consists of at least three levels and an iterative affine parameter estimation produces accurate results.

The reference frame and the warped one do not, in general, have the same metric since, in most cases, the mapping function \mathcal{T}_{ij} is not a translation but a true affine transform, and therefore influences the computation of image gradients for moving object detection. This change in metric can be incorporated into the optical flow equation associated to the image sequence \mathcal{I}_i in order to detect more accurately the moving objects. Indeed, the optical flow associated to the image sequence \mathcal{I} is:

$$\nabla \mathcal{I}_i \nabla^T \mathcal{I}_i w = -\nabla \mathcal{I}_i \frac{d\mathcal{I}_i}{dt} \quad (3)$$

where $w = (u, v)^T$ is the optical flow. Expanding the previous equation we obtain:

$$\begin{aligned} \nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij}) \nabla^T I_i(\mathcal{T}_{ij}) \nabla^T \mathcal{T}_{ij} w = \\ -\nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij}) (I_{i+1}(\mathcal{T}_{i+1, j}) - I_i(\mathcal{T}_{i, j})) \end{aligned} \quad (4)$$

and therefore, the normal flow w_{\perp} is characterized by:

$$w_{\perp} = -\frac{(I_{i+1}(\mathcal{T}_{i+1, j}) - I_i(\mathcal{T}_{i, j}))}{\|\nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})\|} \cdot \frac{\nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})}{\|\nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})\|} \quad (5)$$

Although w_{\perp} does not always characterize image motion, due to the aperture problem, it allows to accurately detect moving points. The amplitude of w_{\perp} is large near moving regions, and becomes null near stationary regions. Figure 1 illustrates the detection of moving vehicles in a video stream taken from an airborne platform. We encourage the reader to view the movie files available at http://iris.usc.edu/home/iris/icohen/public_html/tracking.htm which illustrate the detection on the raw video sequence and on the projected mosaic.

3 Graph Representation of Moving Objects

The detection of moving objects in the image sequence gives us a set of regions which represent the locations where

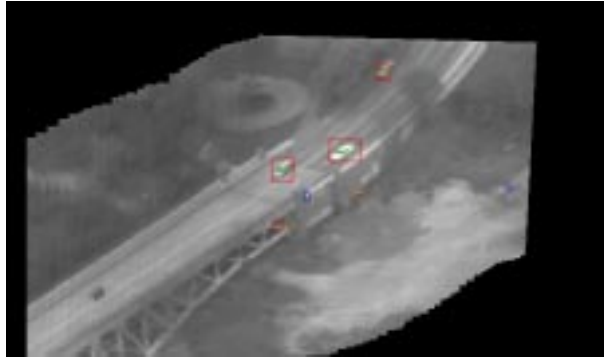


Figure 1: *Detection of several vehicles in a video stream acquired by an airborne platform.*

a motion was detected. The normal component given by equation (5) allows, given a pair of frames, to detect points of the image where a motion occur. These points are then aggregated into regions by considering a thresholded value of the normal component of the optical flow, and then labeled using a 4-connectivity scheme. Each of these connected components represents a region of the image where a motion was detected.

The purpose of detecting moving objects in video stream is to be able to track these objects over time and derive a set of properties from their trajectory such as their behaviors. Commonly used approaches for tracking are token-based, when a geometric description of the object is available [2], or intensity-based (optical flow, correlation...). These techniques are not appropriate for blob tracking since a reliable geometric description of the blobs cannot be inferred. On the other hand, intensity-based techniques ignore the geometric description of the blob. Our approach combines both techniques by incorporating in the representation of the moving objects both spatial and temporal information. Such a representation is provided by a *graph* structure where nodes represent the detected moving regions and edges represent the relationship between two moving regions detected in two separate frames. Each newly processed frame generates a set of regions corresponding to the detected moving objects. We search for possible similarities between the newly detected objects and the previously ones. Establishing such connections can be done through different approaches such as template matching [5] or correlation [11]. However, in video surveillance, little information about the moving object is available, since the observed objects are of various types. Also, objects of small size (humans in airborne imagery) or large changes of objects size are frequent and therefore unsuitable for template matching approaches.

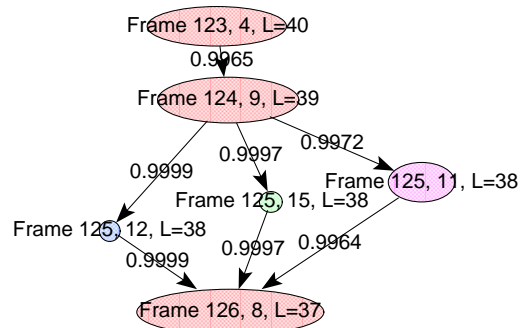
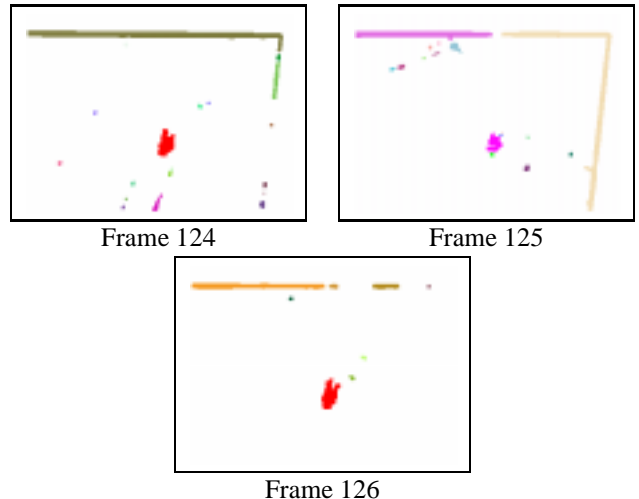


Figure 2: *Detected regions and associated graph.*

Each pair of frames gives us a set of regions where residual motion was detected (see Figure 2). These regions can be related to the previously detected one by measuring the gray level similarity between a region at time t and a set of regions at time $t + 1$ located in its neighborhood. A region may have multiple matches, and the size of this neighborhood is obtained from the objects motion amplitude. In Figure 2 we show the graph representation associated to the detected red blob. Each node is a region represented by an ellipsoid derived from the principal directions of the blob and the associated eigenvalues. Also, a set of attributes is associated to each node as illustrated in Figure 3. We assign to each edge a cost which is the likelihood that the regions correspond to the same object. In our case, the likelihood function is the image gray level correlation between a pair of regions.

4 Dynamic Template Inference

The graph representation gives an exhaustive description of the regions where a motion was detected, and the way these regions relate one to another. This description is appropriate for handling situations where a single moving object is detected as a set of small regions. Such a situation

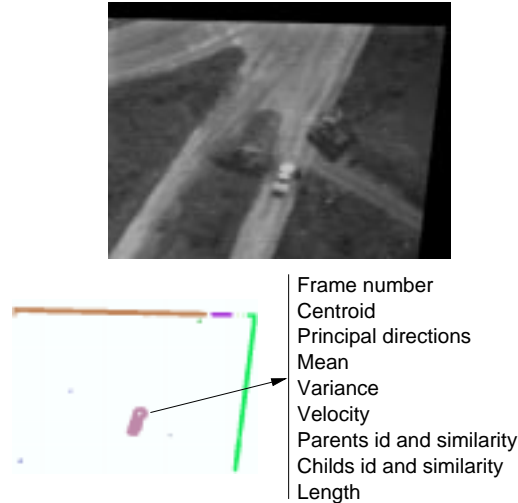


Figure 3: *Description of the attributes associated to each node of the graph. Each color represents a moving region.*

happens when, locally, the normal component of the optical flow is null (aperture problem) and consequently, instead of detecting one region, we have a set of small regions. Usually, clustering techniques are applied for merging the detected blobs in order to recover the region corresponding to the moving object. These image-based techniques [6, 9] rely on the proximity of the blobs in the image and frequently merge regions that belong to separate objects.

Among the detected regions some small regions should be merged into a larger region, or have a trajectory of their own. In both cases, based on the graph representation, these regions belong to a connected component of the graph. In our approach, we cluster the detected regions in the graph rather than in a single image as used in previous works [6, 9]. Indeed, clustering through the graph prevents us from merging regions belonging to objects having a distinct trajectory, since the clustering based on image proximity, is done within a connected component of the graph.

The robustness of the clustering technique is also improved by maintaining a dynamic template of the moving objects for each connected component and therefore for each moving object in the scene. Several techniques were proposed for automatically updating a template description of the moving objects; weighted shape description [9] or cumulative motion images [1] were proposed. The main drawback of these approaches is that error in shape description (i.e. boundaries) are propagated and therefore these techniques are not suitable for moving camera. We propose an approach based on *median shape template* which is more stable and produces a robust description of templates. The templates are computed by applying a median filter (after

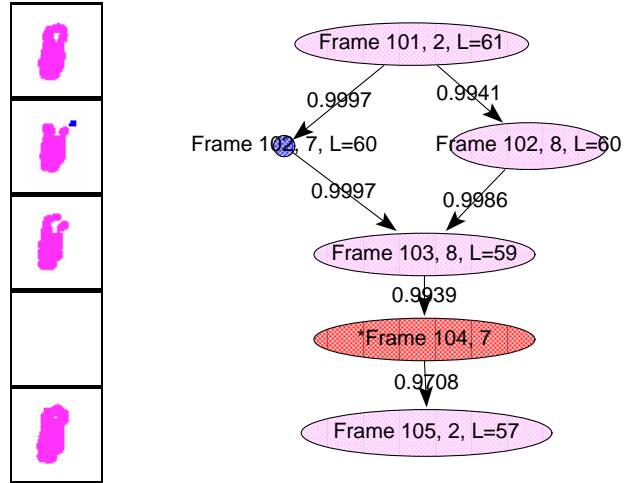


Figure 4: *Propagation of the nodes in order to recover the description of undetected objects. On the left we show the detected region at each frame and, on the right, the associated graph where the red node represents a node inferred from the median shape of the template.*

aligning the centroid and the orientation of each blob) over the last five detected frames of the region.

The dynamic template allows completing the graph description. In video surveillance applications, objects often stop, then resume their motion, such an object will be described through several connected components in the graph. These connected components are merged by using the dynamic template, of the object being tracked: we propagate each node without a successor, into a given number of frames and search for the matching regions in these areas. This defines a set of possible matches, which are incorporated in the graph structure by defining new edges connecting the matched regions. This step is illustrated in figure 4, where the object, not detected in frame 104, is represented by the red node in the graph.

5 Extraction of Objects Trajectories

As new frames are acquired and processed, we incrementally construct the graph representation of moving objects. Deriving the trajectories of the objects from the graph and from the newly detected regions amounts to extract a path along each graph's connected component. We propose an approach for automatically extracting the trajectories of all moving objects through the search of an optimal path representing object's trajectory. Furthermore, the starting node (source) as well as the destination node (goal) are not known in advance. We therefore, consider each graph node without a parent as a potential source node, and each node without a

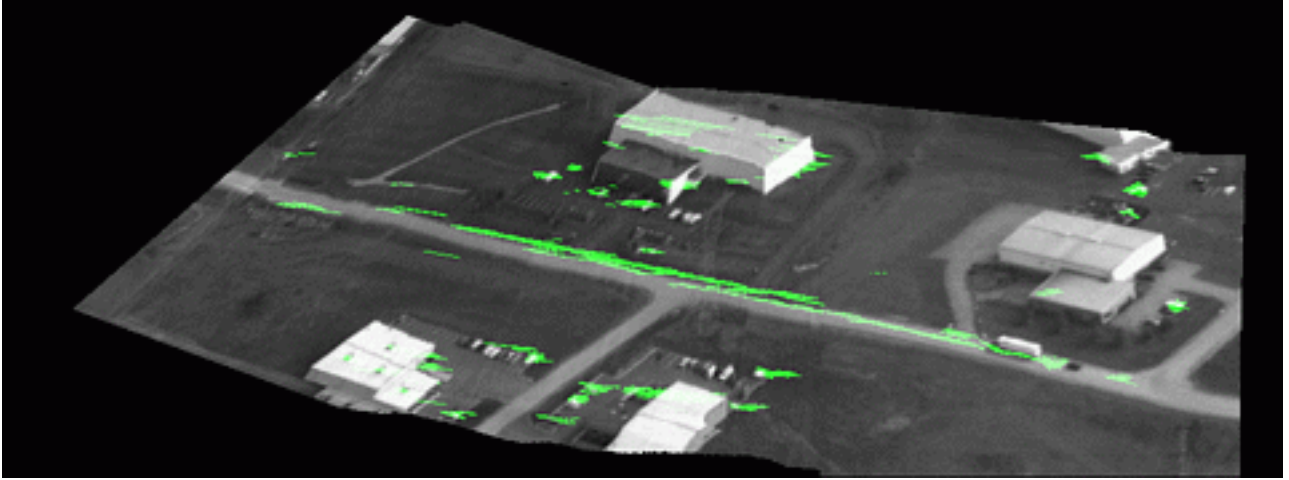


Figure 5: Trajectories of the truck and the car mapped on the generated mosaic.

successor as a potential goal node.

Defining an optimality criterion to characterize an optimal path is equivalent to associating to each edge of the graph a cost. Each edge of the graph corresponds to a match between two regions and has a cost which is the similarity measure between the connected nodes. Therefore, a set of properties associated to each node such as the gray level distribution, the centroid and the dynamic template of the object are used in order to infer a robust path. These properties are merged in the following cost associated to each edge of the graph:

$$c_{ij} = \frac{C_{ij}}{1 + d_{ij}^2} \quad (6)$$

where, C_{ij} is the gray level and shape correlation between regions i and j , and d_{ij} represents the distance between their centroids.

The edge cost given by equation (6) allows to extract the local optimal path. Indeed, a graph search algorithm based only on the edge cost will provide a sub-optimal solution since there are no constraints on the destination or goal node that have to be reached. In the different experiments led, we have observed that this criterion yields a part of the trajectory. The goal source was selected based on the highest value of the cost regardless of the other nodes belonging to the same connected component.

In the graph description used each connected component of this graph represents a moving object in the scene and the location of each node in the graph allows to characterize how far this node is from a potential goal node: a newly detected region. Such a characterization is done by assigning to each node the maximal length of graph's path starting at this node. The computation of the *node's length* is carried

very efficiently by starting at the bottom of the graph, i.e. nodes without successor, and assigning for each parent node the maximum length of his successors plus one. The length of a node i is given by the following equation:

$$l_i = \max\{l_j, j \in \text{successor}(i)\} + 1 \quad (7)$$

with the initial estimate: $l_i = 1$, if $\text{successor}(i) = 0$.

The combination of the cost function (6) and the length of each node allows us to define a new cost function for each node. The cost function associated to the edge connecting the node i to the node j is then defined by:

$$\mathcal{C}_{ij} = l_j c_{ij} \quad (8)$$

where c_{ij} is defined by (6) and l_j is the length of the node j defined by equation (7). This cost function recovers the optimal path among the paths starting at the node being expanded.

The extraction of the optimal path is done by starting at graph's nodes without parent and expanding the node with maximal value of \mathcal{C}_{ij} . This approach is illustrated in Figure 5, where the trajectories of a truck and a car are displayed. The AVI files of the processed video streams are available at http://iris.usc.edu/home/iris/icohen/public_html/tracking.htm.

6 Evaluation and Quantification

Our approach is based on a temporal integration of the moving objects over a certain number of frames which we call: the system's *latency time* (set here to five frames). This latency time, or delay, helps us in selecting the moving regions, and distinguish these blobs from inaccuracies

due to the compensation of the camera’s motion. Moreover, the confidence in the extracted moving region increases as new occurrences of the objects are detected in the processed frames. Indeed, the length (see eq. 7) associated to each graph’s node (i.e. moving region) represents the number of frames in which the object was detected. This scalar value allows us to discard detected blobs which are due to misregistration of the motion compensation algorithm, since these regions have no temporal coherence, characterized by a small length. Table 1 gives some results obtained over several set of video streams acquired by the Predator UAV (Unmanned Airborne Vehicle) and VSAM (Video Surveillance and Activity Monitoring) platforms. These video streams represent a variety of scenes involving human activity, and were used to evaluate the performance of our system.

The numerical values represent the outputs obtained at different stages of processing. The “Moving Objects” column represents the true number of objects moving in the video stream, and was provided by the user. The next two columns represent the output of the detection and tracking sub-modules respectively. As we can see, the number of regions detected is fairly large compared to the number of moving objects. These numbers correspond to the number of regions where the normal flow field was larger than a given threshold (10^{-5} , in all the experiments). The detection column gives the distribution’s plot of the number of these regions over the processed sequence. Also, the associated mean and variance are given as indicative values. The temporal integration of these regions, over a set of frames, allows us to reduce this number of regions (given in the fourth column) and discard *false detections*, since regions due to noise are not temporally coherent. However, some inaccuracies of the egomotion model, or the presence of a parallax can cause some regions to have a coherent temporal signature. Finally, the column “paths”, represents the number of trajectories considered as valid, *i.e.* coherent temporal regions detected for more than 10 frames, which represents the latency time used in the tracking. In some cases, the number of trajectories is larger than the number of moving objects in the stream. This is due to object trajectories being fragmented into several paths, and to failures in matching similar regions representing the same object. The remaining trajectories are due to regions with good temporal coherence which do not correspond to moving objects, and are, mostly, due to strong parallax.

Finally, we have defined two metrics for characterizing the *Detection Rate* (DR) and the *False Alarm Rate* (FAR) of the system. These rates, used to quantify the output of our system, are based on:

- TP (true positive): detected regions that correspond to moving objects,
- FP (false positive): detected regions that do not correspond to a moving object,
- FN (false negative): moving objects not detected.

These scalars are combined to define the following metrics:

$$DR = \frac{TP}{TP + FN} \quad \text{and} \quad FAR = \frac{FP}{TP + FP}$$

These metrics are reported in table 1. As the number of moving objects is small, these measurements may have large variances. This table shows that the large number of moving objects generated by the detection is reduced by the tracking, leading to a perfect detection rate in all examples. The large FAR in the last two experiments is due to 3D structures. In this case a further processing is needed in order to distinguish motion from parallax.

7 Conclusion

We have addressed several problems related to the analysis of a video stream. The framework proposed is based on a graph representation of the moving regions extracted from a video acquired by a moving platform. The integration of the detection and tracking in this graph representation allows to dynamically infer a template of all moving objects in order to derive a robust tracking in situations such as stop and go motion and partial occlusion. Finally, the quantification of the results through the definition of the metrics *DR* and *FAR* provides a confidence measure characterizing the reliability of each extracted trajectory.

The obtained results will be improved by further processing the false alarms in order to discard the trajectories due to regions with good temporal coherence which do not correspond to moving objects, and these are, typically, regions due to strong parallax.

References

- [1] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, pages 928–934, Puerto-Rico, June 1997. IEEE.
- [2] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.
- [3] W.E.L. Grimson, L. Lee, R. Romano, and C. Stauffer. Using adaptive tracking to classify and monitor activities in a site. In *CVPR98*, pages 22–31, 1998.


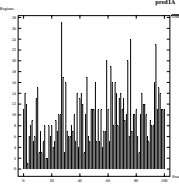

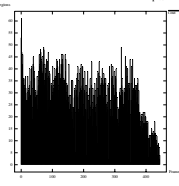

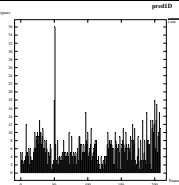

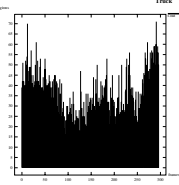

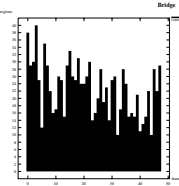
video stream	Moving Objects	Detection			Tracking		Metrics	
		detected regions	mean	σ	Regions	Paths	DR	FAR
	1		9	5	1	1	1.	0.
	2		29	15	3	3	1.	0.2
	4		6	3	4	5	1.	0.
	2		34	11	10	5	1.	0.8
	7		22	8	15	12	1.	0.53

Table 1: Quantitative analysis of the detection/tracking modules

- [4] I. Haritaoglu, D. Harwood, and L.S. Davis. W4S: A real-time system for detecting and tracking people in 2 1/2-d. In *ECCV98*, 1998.
- [5] D.P. Huttenlocher, J.J. Noh, and W.J. Rucklidge. Tracking non-rigid objects in complex scenes. In *ICCV*, pages 93–101, Berlin, Germany, May 1993.
- [6] S.S. Intille, J.W. Davis, and A.F. Bobick. Real time closed world tracking. In *CVPR97*, pages 697–703, 1997.
- [7] M. Irani, P. Anandan, and S. Hsu. Mosaic based representation of video sequences and their applications. In *ICCV*, pages 605–611, Cambridge, Massachusetts, June 1995. IEEE.
- [8] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *ICCV98*, pages 107–112, 1998.
- [9] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real time video. In *WACV98*, pages 8–14, 1998.
- [10] R. P. Wildes and L. Wixson. Detecting salient motion using spatiotemporal filters and optical flow. In *DARPA Image Understanding Workshop*, volume 1, pages 349–356, Monterey, November 1998.
- [11] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, Stockholm, Sweden, May 1994.