Detecting and Tracking Moving Objects in Video from an Airborne Observer

Isaac Cohen Gérard Medioni
University of Southern California
Institute for Robotics and Intelligent Systems
Los Angeles CA 90089-0273
{icohen|medioni}@iris.usc.edu http://iris.usc.edu/Outlines/vsam-project.html *

Abstract

We address the detection and tracking of moving objects in a video stream obtained from a moving airborne platform. The approach is based on two modules; first compensates the image flow induced by the motion of observation platform to detect moving objects and track these regions using a dynamic template derived from their temporal coherence. A graph representation of these objects allows us to characterize objects trajectories as an optimal search path. We also quantify the performance of such a system by providing a confidence measure reflecting the accuracy of each module. This framework shows that the integration of well known tools and an efficient description of the moving objects can give very accurate detection and tracking of moving objects.

1 Introduction

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 the video stream is based on an 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 detection and tracking moving objects in the context of video surveillance. We deal with video streams obtained from a moving airborne platform, this more general case allows us to evaluate the proposed approach for processing, arbitrary video stream acquired by a moving platform. We propose a two stage approach for detecting and tracking moving objects:

- compensation of the image flow induced by the motion of observation platform (also called image *stabilization*). We use a feature based approach for estimating parameters of the affine warping model. The *detection* of moving regions in each frame is obtained by using the normal flow field.
- *tracking* of moving regions in time. We infer, using the temporal coherence of the object over a number of frames, a 2D template of the moving object. This dynamic template is then used to track the objects over frames.

The development of such a system for objects detection and tracking requires us to address the following issues:

- inaccuracies of the stabilization module due to poor image quality, noise spikes and the presence of 3D distortions.
- false alarms and non detection of the detection module, due to failures of the stabilization module, or the very small size of the objects (i.e. humans).
- tracking difficulties, due to failures of the previous modules, and partial occlusion of the objects, or stop and go motion.
- quantifying the obtained results in order to evaluate the detection and tracking algorithms used and associate a confidence measure to the obtained objects trajectories.

2 Egomotion Estimation

The egomotion estimation is based on the camera model which relates 3D points to their projection in the

*This research is supported by the Defense Advanced Research Projects Agency (DARPA) under contract DAAB007-97-C-J023, monitored by US Army, Fort Monmouth, NJ.

image plane. The framework we use model the image induced flow, instead of the 3D parameters of the general perspective transform [Irani *et al.*, 1995] The parameters are estimated by tracking a small set of feature points in the sequence. 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.

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 estimation of the parameters of a geometric transform $\mathcal{T}$ is done by minimizing the least square criterion:

$$E = \sum (I_0(x_0, y_0) - I_1(\mathcal{T}(x_0, y_0)))^2. \quad (1)$$

We choose an affine model which approximates well the general perspective projection while having a low numerical complexity.

3 Detection of Moving Objects

The estimation of the geometric transform parameters models the motion of the moving platform. We can therefore cancel the motion field induced by the displacement of the observer prior to detecting the moving objects in the scene using temporal gradients [Halevi and Weinshall, 1997], accumulated gradients [Davis and Bobick, 1997] or optical flow [Irani *et al.*, 1992; Cohen and Herlin, 1996] techniques. Image variations are characterized through the normal component of the optical flow field. Normal flow is derived from image spatio-temporal gradients using the geometric transform mapping the original frame to the selected reference frame or to the previous one. Let $\mathcal{T}_{ij}$ denotes the warping of the image i to the reference frame j, then the stabilized image sequence is defined by $\mathcal{I}_i = I_i(\mathcal{T}_{ij})$, and the normal flow $w_\perp$ is then defined by:

$$w_\perp = -\frac{(I_{i+1}(\mathcal{T}_{i+1,j}) - I_i(\mathcal{T}_{i,j}))}{\|\nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})\|} \frac{\nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})}{\|\nabla \mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})\|} \quad (2)$$

The normal component given by equation (2) allows, given a pair of frames, to locate regions of the image where a motion occurs by merging points having a normal flow larger than a given threshold. The detected regions correspond to 3D structures not properly handled by the affine model and to moving objects in the scene. However, we show in the following that the use of the temporal coherence of the detected regions allows to discard noisy blobs. Figure 1 illustrates the detection of moving vehicles in a video stream taken from the VSAM airborne platform.

Figure 1: *Tracking of several cars on a bridge.*

4 Tracking Moving Regions

Inferring objects trajectory requires to match regions detected in two (or more) consecutive frames. In airborne video imagery, the matching has to deal with false detections due to errors in image stabilization and with objects in the scene which stop and resume moving, or may become partially occluded. Therefore matching the detected regions in order to derive a trajectory requires an appropriate representation of the detected regions and a similarity measure to match these regions.

4.1 Matching and Data Structures

Tracking moving objects amounts to match these different regions in order to determine the trajectories of the objects. Different approaches to the tracking problem such as template matching [Huttenlocher *et al.*, 1993] or correlation [Zabih and Woodfill, 1994] can be used. However, in video surveillance, the size of the regions can be small and therefore unsuitable for template matching. We propose instead to infer, from the detected regions, a 2D template of the object. Such a dynamic template is extracted by using the temporal coherence of the object over a number of frames (5 frames here).

A graph representation, as shown in Figure 2, is used in order to represent the moving regions and the

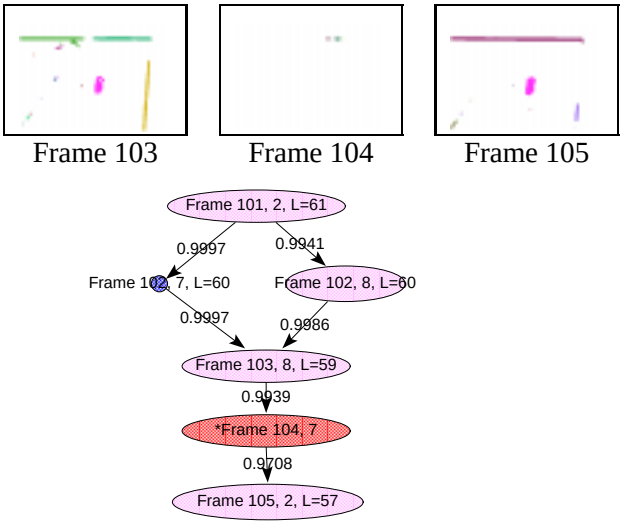


Figure 2: *Illustration of the graph representation of detected moving objects*

way they relate one to another. Each node is a region and each edge represent a possible match between two regions in two different frames. We assign to each edge a cost which is the image grey level correlation between the pair of regions.

4.2 Extraction of Object Trajectory

The extraction of object trajectory from the graph amounts to extract an optimal path for each connected component. The purpose is to automatically extract the trajectories of all moving objects as new frames are acquired. Therefore, each region newly detected is matched to the previous ones and is considered as a potential goal node. On the other hand, each graph node without a parent is a potential source node. The source and the goal nodes define the origin and the end of each extracted path.

Defining a criterion to characterize an optimal path is equivalent to associating to each edge of the graph a cost. This cost must also take into account the location of each node, since nodes describing the same object are more likely to be close one to another. We assign for each edge connecting region i to j the following cost:

$$c_{ij} = \frac{C_{ij}}{1 + d_{ij}^2} \tag{3}$$

where, C_{ij} is the correlation between regions i and j,

Figure 3: *Trajectory of the car and the locations where it stopped (in blue).*

and d_{ij} represents the distance between their centroids.

The edge cost given by equation (3) allows to extract the *local* optimal path. Since there is no fixed goal node, the use of a local criterion such the edge cost usually provides a suboptimal solution rather than the optimal path. In the different experimental results, 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. As new objects are detected and associated to the previous ones, the size of connected component increases and for each node of the graph we can associate a measure reflecting the number of frames in which a similar region was detected. This 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:

$$l_i = max\{l_j, j \in successor(i)\} + 1 \tag{4}$$

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

The combination of the cost function (3) and the length of each node allows us to define a new cost function for each node. This cost function produces the optimal path among the paths starting at the node being expanded. The cost function associated to the edge connecting the node i to the node j becomes:

$$\mathcal{C}_{ij} = l_j c_{ij} \tag{5}$$

where c_{ij} is defined by (3) and l_j is the length of the node j defined by equation (4). 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 3, where a trajectory of the car is shown.

5 Quantification and Evaluation Issues

Our system tries to bridge the gap between low level primitives such as regions, trajectories and behaviors of interest. Using such a framework for surveillance system requires more than a binary recognition of a specific behavior. Instead, such a complex inference requires a measure of belief and indications of reliability and confidence. Given that the two modules: detection and tracking feed on each other, it is necessary to characterize errors in each of these modules and their ripple effects. We have designed our system so that each module expects temporary failures of the previous one, and attempt to compensate for them using temporal coherence.

Few attempts were made in computer vision to evaluate such an integrated system where each component has its own inaccuracies and limitations. Our approach, based on a simultaneous processing of the detection and tracking and the efficient representation of the objects through a graph allows us derive a confidence measure after each of these tasks. Also, we choose here to evaluate each of the modules independently. The issues to be addressed are:

- **Detection**: Do we detect all the objects? Among the detected regions, how many of these are false alerts? What are the minimal size, speed of the moving objects that can be detected?
- **Tracking**: The temporal integration of the detected objects and their inter-relationship allows us to infer from a collection of blobs, representing the moving objects, a set of paths corresponding to the trajectories of the moving objects. How accurate are these trajectories? What is the stability of the trajectories inferred with regard to failure to detect the moving objects in one or several frames, stop and go motion, occlusion,…
- **Combined quantification**: inferring a confidence measure of the output.

5.1 Quantifying Moving Objects Detection

The detection of moving objects, as described in section 3, is performed after compensating for the motion of the platform. The moving objects are detected as regions where a residual motion subsists. The extracted regions are then due to moving objects present in the scene, to the inaccuracies of compensation algorithm used and/or to the presence of parallax. Consequently, the number of regions extracted by the detection algorithm is larger than the number of moving objects in the scene and cannot be considered for quantifying the accuracy of the detection algorithm unless a static camera is used, or a perfect egomotion estimation is achieved.

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 deciding which are 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. 4) associated to each graph's node (ie 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 which is characterized by a small length.

Table 1 gives some results obtained over several set of video streams acquired by the Predator UAV and VSAM platforms. These video streams represent a variety of scenes involving human activity (see figures 1 and 3), 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 detec-

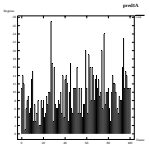
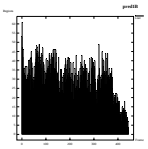
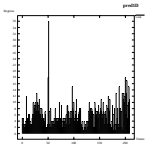
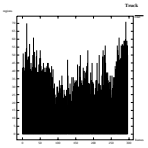
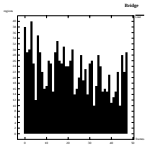
video stream	Moving Objects	Detection	Tracking		Metrics	
		detected regions	Regions	Trajectories	DR	FAR
	1	 $\bar{x} = 9, \sigma = 5$	1	1	1.	0.
	2	 $\bar{x} = 29, \sigma = 15$	3	3	1.	0.2
	4	 $\bar{x} = 6, \sigma = 3$	4	5	1.	0.
	2	 $\bar{x} = 34, \sigma = 11$	10	5	1.	0.8
	7	 $\bar{x} = 22, \sigma = 8$	15	12	1.	0.53

Table 1: *Quantitative analysis of the detection/tracking modules*

tion 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 the *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 "trajectories", 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 situations, this number 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.

5.2 Evaluation of the Tracking

The temporal integration of the detected objects and their inter-relationship allows us to infer from a collection of blobs, representing the moving objects, a set of paths corresponding to the trajectories of the moving objects. The size of these paths (i.e. the number of node belonging to the path) allows us to easily filter out the regions where a temporal variation was detected with no coherence over time.

Deriving, from this set of paths, a measure allowing to evaluate the tracking module is difficult since

several issues have to be considered: Among the detected objects, how many were tracked efficiently? For each tracked object how many paths form its trajectory? These issues are relevant in the case of stop and go motions or occlusion. Indeed, in the first case the trajectory of the object is fragmented into a set of paths. These paths have to be merged into a single trajectory in order to recognize the stop and go motion that occur for example in a checkpoint. The second case, partial or total occlusion, is more subtle since, before merging the collection of paths, one has to identify the object being tracked in order to recognize its different occurrences in the video stream. In table 1 we display for a set of video streams, the number of paths detected and the number of moving objects in the scene.

5.3 Quantification: Confidence Measure Definition

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:

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

These detection rates 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 submodule is reduced by the tracking submodule, leading to a perfect detection rate in all examples. The large FAR in the last two experiments is due to 3D structures. We are in the process of adding a filtering step to differentiate motion from parallax (and infer the 3D at the same time). This should drastically reduce the FAR.

6 Conclusion

We have addressed several problems related to the analysis of a video stream. The framework proposed is based on two modules to perform detection and tracking of moving objects. The integration of these modules and the use of the temporal coherence of the moving objects yields an efficient description of the objects and an accurate description of their trajectories. The definition of a metric for each of these module 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

[Cohen and Herlin, 1996] I. Cohen and I. Herlin. Optical flow and phase portrait methods for environmental satellite image sequences. In *ECCV*, pages 141–150, Cambridge, April 1996.

[Davis and Bobick, 1997] 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.

[Halevi and Weinshall, 1997] G. Halevi and D. Weinshall. Motion disturbance: Detection and tracking of multi-body non rigid motion. In *CVPR*, pages 897–902, Puerto-Rico, June 1997. IEEE.

[Huttenlocher *et al.*, 1993] 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.

[Irani *et al.*, 1992] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV*, pages 282–287, May 1992.

[Irani *et al.*, 1995] 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.

[Zabih and Woodfill, 1994] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, Stockholm, Sweden, May 1994.