# Detection and Tracking of Objects in Airborne Video Imagery

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 detection and tracking of moving objects in a video stream obtained from a moving airborne platform. The approach is based on the compensation of the image flow induced by the motion of observation platform and the detection and tracking of moving regions. The use of such an approach leads us to deal with stabilization inaccuracies, false alarms and non detection of moving objects and tracking difficulties due to partial occlusion or stop and go motion. Our approach use a hierarchical, feature based stabilization scheme and normal component of the residual flow for detecting moving objects. These objects are tracked using a dynamic template for each object, and extracts trajectories as the result of a graph searching algorithm. The proposed framework shows that an integration of well known tools and an efficient description of the moving objects can give very accurate detection and tracking of moving objects.

**Keywords:** Dynamic vision, motion, Image stabilization, ego-motion, object tracking, graph searching.

## 1 Introduction

We address the analysis of a video stream obtained from a moving airborne platform. The context of this work is video surveillance and monitoring. Partial automation of this work is crucial, as further deployment of such platforms is bound to overwhelm human analysts. Furthermore, boredom and fatigue strongly affect the performance of human operators. While the goal of our project is the semantic description of behaviors, the current paper only discusses detection and tracking of moving image regions.

We propose to perform these tasks in three steps:

- compensation of the image flow induced by the motion of observation platform (also called image *stabilization*)
- *detection* of moving regions in each frame,
- *tracking* of moving regions in time.

The issues addressed in this paper relate to:

- 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.

Furthermore, we do not use predefined templates for detection and tracking, in order to keep our system general.

We start by reviewing some of the most relevant work in section 2, then explain why we estimate the image induced by the observer, rather than observer motion itself, in section 3, and relate the two. In section 4, we present our hierarchical, feature based stabilization scheme, and show an example of the resulting image mosaic, which is a byproduct of the processing. Section 5 describes the detection module, which makes use of the normal component of the residual flow. Section 6 gives the details of the tracking module, which extracts a dynamic template for each object, and extracts trajectories as the result of a graph searching algorithm. Section 7 presents some results on actual image sequences, and we discuss limitations and extensions in section 8

## 2 Previous work

As pointed in the introduction, the innovative aspect of the presented work relies on the detection of moving object through image flow compensation. Taken separately, these two topics have been extensively studied in the past years. Image compensation or ego-motion estimation techniques are based on the approximation of the general perspective model using a parametric model [8, 10, 15]. Such a parametric model is estimated using all image pixels or using a small set of feature points. The accuracy and the performance of these methods are enhanced using a multiresolution approach to avoid local minima and reduce the computation time. Here, we use an iterative affine parameter estimation approach based on feature points only.

The detection of moving objects in the warped image sequence is performed by computing the residual motion. This can be achieved by using temporal gradients [4] and optical

flow techniques [2, 9]. A temporal integration of the variations can also be used, as proposed by [3], where the temporal derivatives of the image sequence are accumulated in order to locate regions where motion occurs. Our approach is based on the optical flow normal component, which takes into account the mapping functions used to warp the image sequence to the reference frame. This allows to take into account the change in metric between the processed frame and the reference one, and therefore increases the accuracy of the moving objects detection.

The detected regions allows to derive a 2D dynamic template of the moving object through its temporal coherence. This dynamic template allows us to increase the accuracy of the tracker algorithm in the case of motion segmentation errors and occlusions.

## 3 Egomotion estimation

The ego-motion estimation is based on the camera model which relates 3D points to their projection in the image plane. The framework we use is to model the image induced flow, instead of the 3D parameters of the general perspective transform [8, 10, 13]. 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.

Let $P = (X, Y, Z)$ denotes a 3D point in the world coordinate system and $p = (x, y)$ be the corresponding position of $P$ in the image plane defined by the $x$ and $y$ axes and perpendicular to the $Z$ axis that contains the point $(0, 0, f)$. The perspective projection of the point $P$ in the image plane is: $p = (x, y) = (fX/Z, fY/Z)$ where $f$ is the camera focal length.

Any arbitrary motion of the camera can be described by a translation $T$ and a rotation $R$, so that a point $P_0$ in the 3D world coordinate system at time $t_0$ has a new position $P_1$ at time $t_1$ given by: $P_1 = R_{01}P_0 + T_{01}$.

Using the rigid transform and the perspective projection, we can derive the following equation relating the points $p_0$ and $p_1$:

$$\begin{pmatrix} x_1 \\ y_1 \\ f_1 \end{pmatrix} = \frac{f_1}{f_0}\frac{1}{Z_1} R_{01} \begin{pmatrix} x_0 \\ y_0 \\ f_0 \end{pmatrix} + \frac{f_1}{Z_1} \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} \quad (1)$$

where we have set $Z_0 = 1$. Representing the rotation matrix by the Euler angles $(\omega, \varphi, \psi)$, we can derive the equation:

$$Z_1 = sin\varphi x_0 - sin\omega cos\varphi y_0 + cos\omega cos\varphi f_1 \quad (2)$$

The general perspective projection given by equation (1) is non linear, so the use of such an equation is not suitable for real time application, as it requires the use of iterative optimization techniques [12]. Instead, we use a parametric approximation of the general perspective projection and therefore, an estimate of the accuracy of the approximate model have to be established. In the case of the affine and pan-tilt zoom models, we can derive the relationship between the coefficients of the model to the general perspective projection equation.

### 3.1 The affine model

Recovering the rotation and translation parameters from a perspective projection leads to the difficult solution of a non linear set of equations. The proposed approaches are iterative and are often applied to a set of images, rather than to a video stream. However, an affine approximation of the general perspective model can be used in some specific cases. Indeed, when the distance between the camera center and the object is large compared to the depth of the scene being viewed, the perspective equations can be well approximated through an affine model.

Let us consider the affine model given by equation:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \mathcal{T} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \quad (3)$$

which describes the relationship between the projection of the 3D points $P_0$ and $P_1$.

We can relate the parameters of the affine model to the Euler angles $(\omega, \varphi, \psi)$ used in the general case by identifying the affine model with the model given by equation (1). We have the following equations:

$$\begin{cases} a = \frac{1}{Z_1}\frac{f_1}{f_0} cos\varphi cos\psi \\ b = \frac{1}{Z_1}\frac{f_1}{f_0} (sin\omega sin\varphi cos\psi + cos\omega sin\psi) \\ c = \frac{1}{Z_1}\frac{f_1}{f_0} cos\varphi cos\psi \\ d = \frac{1}{Z_1}\frac{f_1}{f_0} (-sin\omega sin\phi sin\psi + cos\omega cos\psi) \end{cases} \quad (4)$$

and

$$\begin{cases} e_1 = \frac{f_1}{Z_1} (-cos\omega sin\varphi cos\psi + sin\omega sin\psi + T_x) \\ e_2 = \frac{f_1}{Z_1} (cos\omega sin\varphi sin\psi + sin\omega cos\psi + T_y) \end{cases} \quad (5)$$

with $Z_1$ given in Eq. (2).

We can use this set of equations to derive the relationship between the affine parameters and the Euler angles used in the rigid transformation (1). Indeed, if the scalars $a, b, c$ and $d$ are non zero, we have the following equation:

$$\begin{cases} tan\psi = \frac{c}{a} \\ sin\varphi tan\omega = \frac{b-c}{(b-c)tan\psi + d} \\ \frac{cos\varphi}{cos\omega} = \frac{a}{(b-c)tan\psi + d} \end{cases} \quad (6)$$

An analytical solution of this set of equations leads to: $\left\{\phi = 2\arctan(t), \omega = -2\arctan\left(\frac{x(t^4-1)}{2t(-t^2+1+y+t^2y)}\right)\right\}$ where $x = sin\varphi tan\omega$, $y = \frac{cos\varphi}{cos\omega}$ and $t$ is the root of the equation: $(x^2Z^8 + (4-4y^2)Z^6 - (2x^2+8+8y^2)Z^4 + (4-4y^2)Z^2 + x^2) = 0$.

These equations describes the relationship between the affine model and the perspective projection. They can be used to correct the errors of the approximation model. Indeed, a more accurate location of the projection can be derived from the perspective equations. Note that, such an approach requires a good estimate of the focal lengths $f_0$ and $f_1$.

### 3.2 The Pan-Tilt Zoom model

The pan-tilt model is a three degrees of freedom model, where we assume that the camera is on a fixed tripod but allows rotations of the camera along the $X$ and $Y$ axes, and a change in camera focal length. The equations of the model can be derived from the general perspective projection equation (1). The following equation describes the relationship between the projection of two points $p_0$ and $p_1$ representing the projection, on the image plane of the two points $P_0$ and $P_1$:

$$\left\{\begin{array}{l} x_1 = \frac{1}{Z_1}\frac{f_1}{f_0}\left(cos\varphi x_0 + sin\omega sin\varphi y_0 - cos\omega sin\varphi f_0\right) \\ y_1 = \frac{1}{Z_1}\frac{f_1}{f_0}\left(cos\omega y_1 + sin\omega f_0\right). \end{array}\right. \tag{7}$$

where $Z_1$ is given by equation (2).

In the case of rotations of small amplitude, and if the distance between the camera center to the object is large compared to depth of the scene being viewed, a good approximation is given by the zoom and translation model described by the following equation:

$$\left\{\begin{array}{l} x_1 = sx_0 + t_x \\ y_1 = sy_0 + t_y \end{array}\right. \tag{8}$$

The parameters $s, t_x$ and $t_y$ of the pan-tilt-zoom model can be obtained by identifying the equation (8) with equation (7):

$$\left\{\begin{array}{l} s = \frac{f_1}{f_0} \\ t_x = -f_1 cos\omega tan\varphi \\ t_y = f_1 tan\omega \end{array}\right. \tag{9}$$

Henceforth, we can derive the relationship between the parameters $s, t_x$ and $t_y$ and the Euler angles $\omega$ and $\varphi$:

$$\left\{\begin{array}{l} \omega = atan\left(\frac{t_y}{f_1}\right) \\ \varphi = -atan\left(\frac{t_x}{f_1 cos\omega}\right) \end{array}\right. \tag{10}$$

These equations can also be used to evaluate the accuracy of the zoom and translation approximation of the pan-tilt zoom model. Indeed, we can rewrite Pan-Tilt-Zoom equations (7) in terms of the parameters $s, t_x$ and $t_y$:

$$\left\{\begin{array}{l} x_2 = \frac{f_1}{f_0}\frac{cos\varphi}{Z_1}\left(\frac{f_1}{f_0}x_1 - \frac{t_x t_y}{f_1^2}y_1 - t_x\right) \\ y_2 = \frac{f_1}{f_0}\frac{cos\omega}{Z_1}\left(y_1 + \frac{f_0}{f_1}t_y\right) \end{array}\right. \tag{11}$$

and use these points locations to increase the accuracy of the coefficients estimation.

## 4 Image sequence stabilization

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 $I_0$. Several authors have proposed different approaches to solve this problem. The most common models involve affine [7, 10] or quadratic approximation of the motion [7]. However, for processing a few number of frames, typically different pictures of a scene, some authors do consider the general perspective equations [13] since the computation time is not an important issue. Among these methods we can distinguish two types of approaches: intensity or feature based approaches. The intensity approaches rely on *all* image pixels to recover the set of parameters, while the feature based approach extract a set of relevant points in the frames and derive the parameters according to the matching of these points. This second approach is less time consuming and allows fast and accurate registration of the frames.

Recovering the parameters of the geometric transformation amounts to minimizing the least square criterion:

$$\begin{aligned} E &= \sum_i \left(I_0(x_0, y_0) - I_1(x_1, y_1)\right)^2 \\ &= \sum_i \left(I_0(x_0, y_0) - I_1(\mathcal{T}(x_0, y_0))\right)^2 \end{aligned} \tag{12}$$

This criterion leads to a global non linear minimization scheme which could be handled by a Levenberg-Marquardt method [12]. An alternative solution can be obtained iteratively using all image points and a coarse to fine approach [11]. Both approaches are computationally expensive.

Our approach is based on multigrid matching of features points extracted from the reference and target frames. There are several ways to define a feature point and each definition is context dependent. However, corners or high curvature points, are commonly used as feature points in matching algorithms. The extraction of corner points can be done through the use of a *corner model* [15] or a rough estimation can be obtained through image partial derivatives. In this paper, we extract the feature points by considering a partition of the image into a regular grid and extract in each cell the point which

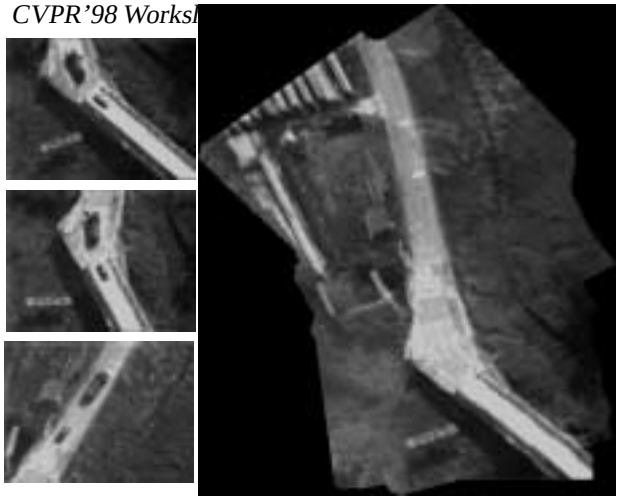

Figure 1: *Mosaic obtained by the hierarchical feature based approach.*

maximizes: $|\nabla_x I| + |\nabla_y I|$. Moreover, we compute a pyramid representation of each frame, and feature points are extracted at all resolutions. This coarse to fine approach allows to recover large displacements between two adjacent frames of the image sequence.

The features points selected in the reference image $(x_0, y_0)$ and in the target image $(x_1, y_1)$ are matched using local correlation. This gives us a set of pair of points from which we can derive the parameters of the motion model by minimizing the criterion given by eq. 12. Using the affine model, the minimum for $E$ in eq. (12) is obtained by solving the set of linear equations:

$$\begin{pmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 \end{pmatrix}_i \begin{pmatrix} a & b & t_x & c & d & t_y \end{pmatrix}^T = (x_1, y_1)_i^T \quad (13)$$

where $i = 1..N$ is the index of the feature points. This system of equations solved at each level of the pyramid by propagating the feature points and the parameters obtained at the previous level. Since this approach only works if all the points fit the motion model, we have adapted the basic algorithm [12, 8] to also handle the presence of independently moving objects in the scene. This is achieved by scaling the coefficients of the motion equations inversely proportional to the temporal derivatives. Moving objects are not registered when the images are warped according to the camera induced motion field. Therefore, pixels corresponding to moving objects have high temporal derivatives, and consequently less weight. Figure 1 illustrates a byproduct of the processing, a mosaic obtained from a video stream.

## 5 Detection of moving objects

Stabilization allows us to create a synthetic image sequence in which we have cancelled the motion field induced by the displacement of the observer. The detection of moving objects is accomplished by computing the residual motion in this stabilized image sequence. This motion can be derived through different schemes such as temporal gradients [4] or optical flow [9, 2]. These detected variations are due to 3D structures, not properly handled by the affine model, or the moving objects in the scene. These regions are usually small and cannot be used to infer the 3D geometry of the moving object. However, we can use the redundancy of the information and its temporal coherence in order to locate and track moving objects in the scene. Our method is based on a 2D multiframe motion analysis for locating and tracking moving objects.

Locating the regions of the image where residual motion occurs is done using the normal component of the optical flow field. Normal flow is derived from image spatio-temporal gradients of the stabilized image sequence. Consequently, each frame of the image sequence is obtained by mapping the original frame to the selected reference frame. Indeed, let $\mathcal{T}_{ij}$ denotes the warping of the image $i$ to the reference frame $j$, then the stabilized image sequence is defined by $\mathcal{I}_i = I_i(\mathcal{T}_{ij})$. These two frames do not, in general, have the same metric since, in most cases, the mapping function $\mathcal{T}_{ij}$ differs from the identity, and therefore influences the computation of the normal flow. The mapping function is defined by the following equation:

$$\mathcal{T}_{ij} = \prod_{k=i,..,j+1} \mathcal{T}_{k,k-1} \quad (14)$$

The optical flow equation associated to the image sequence $\mathcal{I}_i$ is:

$$\nabla\mathcal{I}_i \nabla^T \mathcal{I}_i w = -\nabla\mathcal{I}_i \frac{d\mathcal{I}_i}{dt} \quad (15)$$

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

$$\begin{aligned} &\nabla\mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij}) \nabla^T I_i(\mathcal{T}_{ij}) \nabla^T \mathcal{T}_{ij} w = \\ &-\nabla\mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij}) (I_{i+1}(\mathcal{T}_{i+1,j}) - I_i(\mathcal{T}_{i,j})). \end{aligned} \quad (16)$$

The normal flow $w_\perp$ is then defined by:

$$w_\perp = -\frac{(I_{i+1}(\mathcal{T}_{i+1,j}) - I_i(\mathcal{T}_{i,j}))}{\|\nabla\mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})\|} \cdot \frac{\nabla\mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})}{\|\nabla\mathcal{T}_{ij} \nabla I_i(\mathcal{T}_{ij})\|} \quad (17)$$

Although $w_\perp$ does not always characterize image motion, due to the aperture problem, it allows to locate accurately moving points. Indeed, the amplitude of $w_\perp$ is large near moving regions, and becomes null near stationary regions. Figure 2 illustrates the detection of a moving vehicle in a video stream taken from an airborne platform.

Another advantage of using such a formulation is that all computations are performed on the original image sequence, and not on the warped images. The approach presented in this section allows us to include the detection algorithm within the stabilization step.

Figure 2: *Tracking of a car going through a road block. The red rectangle corresponds to the bounding box of the moving object, and the green lines displays its motion direction.*

## 6 Tracking moving regions

The motion computation method presented above uses two consecutive frames to detect moving objects. Such an approach can be satisfactory when dealing with a fixed camera, or with objects having continuous motion. In video surveillance applications, these requirements may not be satisfied, since the objects in the scene stop and resume moving, or may become partially occluded. Moreover, the camera motion has been compensated, and the remaining residual motion may be due to the inaccuracy of the model to characterize properly the projection of a 3D object. Consequently, tracking moving objects consists in matching the different detected regions in order to derive a path describing their trajectory. This step requires a good representation of the detected regions and a similarity measure which can be used for properly matching these regions. Moreover, a reliable approach for tracking moving object must cope with false alerts while characterizing the moving objects.

### 6.1 Matching and data structures

The normal component given by equation (17) allows, given a pair of frames, to locate points of the image where a motion occurs. These points are merged into regions by considering a thresholded value of the normal component of the optical flow. These regions are then labeled using 4-connectivity. Each of these connected components represents a region of the image where a motion was detected.

The detection of the moving objects in the image sequence gives us a set of regions which represents the locations where a motion was detected. Tracking moving objects amounts to matching these different regions in order to determine the trajectories of the objects. The matching of these regions can be done through different approaches such as template matching [5] or correlation [14]. However, in video surveillance, little information about the moving object is available, since the observed objects are of various types. Also, the size of the moving objects can be small and therefore unsuitable for template matching.

Tracking moving objects based on three frames is very hazardous since it may leads to erroneous matches, and is unstable in dealing with false alarms due to the inaccuracies of the ego-motion compensation algorithm. In this paper, 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. In order to extract the templates we make a weak assumption on the shape of the objects: the shape of the objects must be coherent during a small period of time (typically 3 to 5 frames). This assumption is usually satisfied in video surveillance applications, since the frame sampling is high (10 to 15 Hz), and the moving objects have a coherent structure. Temporal integration of the detected objects over a number of frames allows the characterization of the moving objects by computing the motion, its direction and the trajectory of each object. In the presence of ego-motion, The temporal integration of each moving object can be done once the tracking is achieved. Indeed, we first have to locate and match the moving objects before deriving, from a temporal integration scheme, the shape and the direction of the moving objects.

### 6.2 Representation of moving objects

Each pair of frames gives us a set of regions where residual motion was detected (see Figure 3, left). These regions can be related in the time sequence through a correlation. This is done by computing the correlation between a region at time $t$ and a set of regions at time $t + 1$ located in its neighborhood, each region may have multiple matches. The size of this neighborhood is obtained from the objects motion amplitude. A graph representation, as shown in Figure 3 is used in order to represent the moving regions and the way they relate one another. Each node is a region and each edge represent a possible match between two regions in two different frames. Figure 4 illustrates the attributes associated to each node. 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 grey level correlation between a pair of regions.

Such a graph representation gives an exhaustive description of the regions where a motion was detected, and the way

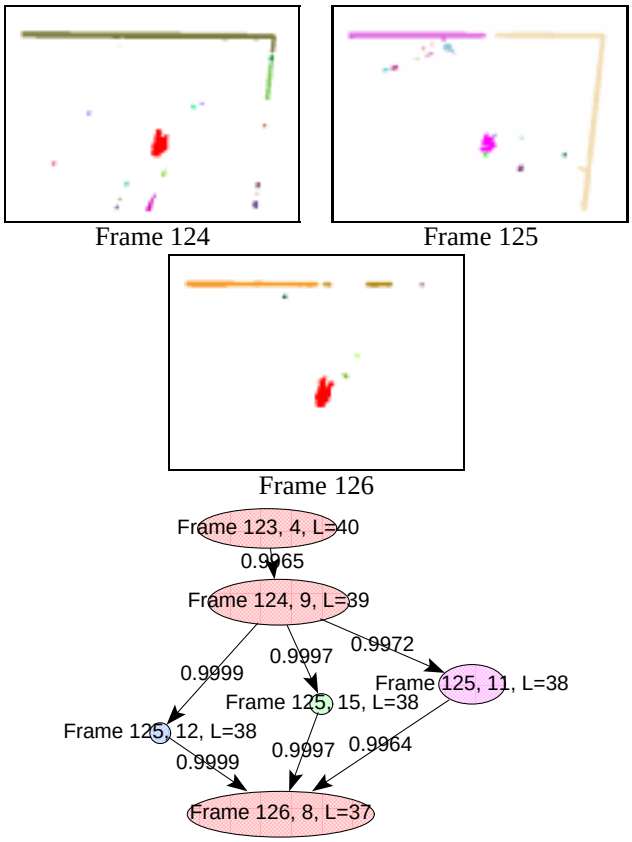


Figure 3: *Detected regions and associated graph.*

these regions relate to one another. This description is quite efficient to handle situations where a single moving object is detected as a set of small regions. This situation arises 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 smaller regions. These small regions should be merged into a larger region, or have a trajectory of their own. In both cases, these regions belong to the same connected component of the graph and are therefore related.

A final step is needed to complete the graph description. In video surveillance applications, objects often stop then resume moving. Consequently, such an object will be described through several connected components of the graph. However, we can merge these connected components by making use of the temporal coherence 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 5, where the object not detected is in red. The obtained graph is a Directed

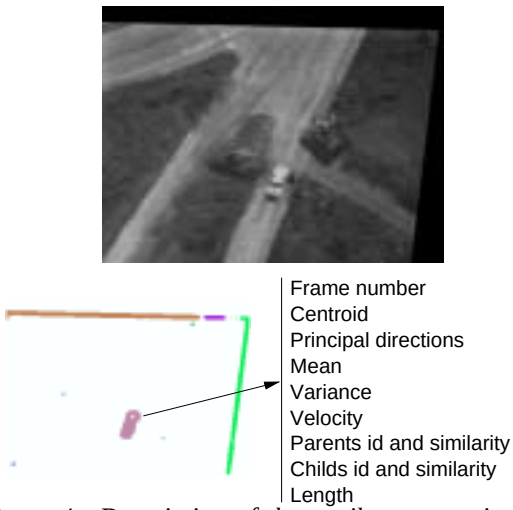


Figure 4: *Description of the attributes associated to each node of the graph. Each color represents a moving region.*

Acyclic graph where, each connected component represents a moving object.

### 6.3 Extraction of object trajectory

The extraction of object trajectory from the graph amounts to extract an optimal path along each connected component. The purpose is to automatically extract the trajectories of all moving objects. Since we have neither a source, nor a goal node, in the following we will consider each graph node without a parent as a potential source node, and each node without a 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 patches. However, we have also to take into account the location of each node, since nodes describing the same object are likely to be close one to another. Therefore we assign for each edge connecting region $i$ to $j$ the following cost:

$$c_{ij} = \frac{C_{ij}}{1 + d_{ij}^2} \tag{18}$$

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

The edge cost given by equation (18) 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 not the optimal path, but a suboptimal solution. 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.

The graph described in the previous section allows us to represent all the moving objects in the processed image

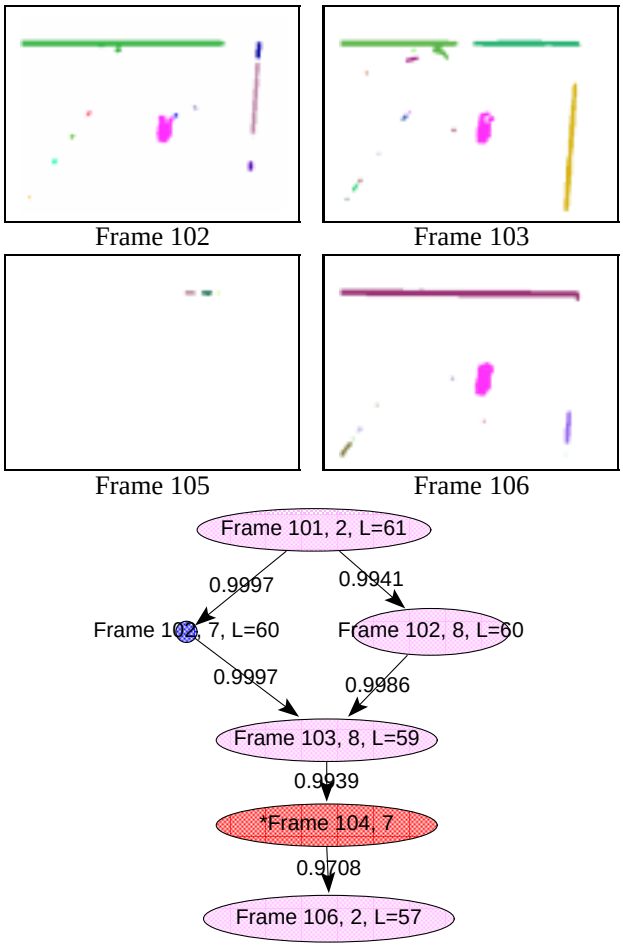


Figure 5: *Propagation of the nodes in order to recover the description of undetected objects.*

sequence. Furthermore, 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. 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:

$$l_i = max\{l_j, j \in successor(i)\} + 1 \qquad (19)$$

with the initial estimate: $l_i = 1$, if $successor(i) = 0$.
The combination of the cost function (18) and the length of each node allows us to define a new cost function for each node. This cost function recover the optimal path among the paths starting at the node being expanded. The cost function

Figure 6: *Trajectory of the car and the locations where it stopped (in blue).*

associated to the edge connecting the node $i$ to the node $j$ is then defined by:

$$\mathcal{C}_{ij} = l_j c_{ij} \qquad (20)$$

where $c_{ij}$ is defined by (18) and $l_j$ is the length of the node $j$ defined by equation (19).

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 6, where a trajectory of the car is shown.

## 7 Experimental results

The presented method was tested on a set of video streams taken from an airborne platform and acquired at 15Hz. The main objective is to help a tele-operator in monitoring several scenes. We illustrate the approach on two video streams, in which objects of different size are moving simultaneously. The reader is encouraged to view the mpeg files **Movie-1** and **Movie-2** contained in the floppy. The first stream represents the monitoring of a road block. The objective is to detect and track (see Figure 2) the cars going through this road block and locate the positions where the cars stop. These locations are shown as blue rectangles. A trajectory of the car is also displayed in the figure 6.
The second example is more subtle. The scene monitored by the airborne platform is a bridge where some human activity takes place. In this image sequence, the size of the moving objects is very small, typically 4 to 5 pixels width by 5 to 6 pixels height. Furthermore, the egomotion estimated with the affine model is corrupted by the 3D geometry of the bridge. In this case, the extraction of a dynamic template in the image sequence and the graph representation of the different occurrences of this template allows us to extract almost perfectly the trajectory of the people in the scene, as displayed in figure 7.

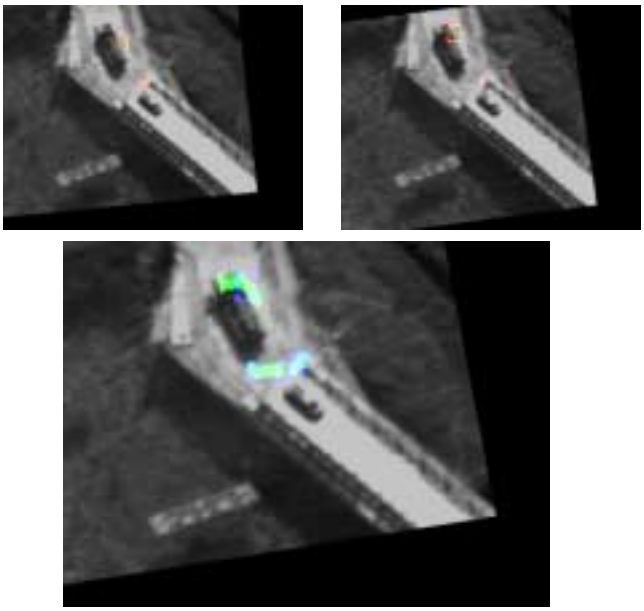

Figure 7: *Tracking of small objects (people) on a bridge.*

## 8 Conclusion

We have addressed several problems related to the analysis of a video stream. The framework proposed is based on a three steps approach image stabilization, detection and tracking of moving objects. Special consideration was given to the inaccuracies or the failures of each module to handle some specific situations such as the false alarm of the detection module due to 3D structures, not properly handled by the affine model. We plan to further refine 3D effects using the epipolar constraint [6].

The goal of our project is the semantic description of behaviors. This description is context based so the next step is the integration of contextual information into the framework [1].

## References

[1] F. Brémond and M. Thonnat. Issues of representing context illustrated by video-surveillance applications. *International Journal of Human-Computer Studies, Special Issue on context*, 1997.

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

[3] 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.

[4] 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.

[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] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *ECCV*, pages 17–26, Cambridge, April 1996.

[7] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *Image Understanding Workshop*, volume 1, pages 639–647, New Orleans, May 1997. DARPA.

[8] 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.

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

[10] C. Morimoto and R. Chellappa. Fast 3D stabilization and mosaic construction. In *CVPR*, pages 660–665, Puerto-Rico, June 1997. IEEE.

[11] S. Peleg and H. Rom. Motion based segmentation. In *ICPR'90*, volume 1, pages 109–113, 1990.

[12] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE WACV'94*, pages 44–53, December 1994.

[13] R. Szeliski and H.Y. Shum. Creating full view panoramic image mosaics and environment maps. In *SIGRAPH97*, Los-Angeles, August 1997. ACM.

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

[15] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2D mosaic from a set of images. In *CVPR*, pages 420–425, Puerto-Rico, June 1997. IEEE.