Motion-Egomotion Discrimination and Motion Segmentation from Image-Pair Streams

David Demirdjian, Radu Horaud

Motion-Egomotion Discrimination and Motion Segmentation from Image-pair Streams

David Demirdjian and Radu Horaud

INRIA Rhône-Alpes and GRAVIR-CNRS, 655 Avenue de l'Europe, 38330 Montbonnot Saint Martin, FRANCE
E-mail: David.Demirdjian@inrialpes.fr; Radu.Horaud@inrialpes.fr

Given a sequence of image pairs we describe a method that segments the observed scene into static and moving objects while it rejects badly matched points. We show that, using a moving stereo rig, the detection of motion can be solved in a projective framework and therefore requires no camera calibration. Moreover the method allows for articulated objects.

First we establish the projective framework enabling us to characterize rigid motion in projective space. This characterization is used in conjunction with a robust estimation technique to determine egomotion. Second we describe a method based on data classification which further considers the non-static scene points and groups them into several moving objects. Third we introduce a stereo-tracking algorithm that provides the point-to-point correspondences needed by the algorithms. Finally we show some experiments involving a moving stereo head observing both static and moving objects.

1. INTRODUCTION

The problems of detection, description, and understanding of motion from visual data are among the most difficult and challenging problems in computer vision. At the low-level, 3-D motion must be analysed based on the 2-D appearance and time evolution features that are observable in images. At the high-level, the 2-D motion fields previously derived must be interpreted in terms of rigid, articulated, or deformable objects, discriminate between objects undergoing distinct motions, estimate the motion parameters, etc.

If the visual sensor moves as well, one more difficulty is added because one has to estimate egomotion (the motion of the visual sensor with respect to some static reference frame) in the same time as motion associated with the observed objects.

Existing techniques for motion/egomotion discrimination and motion segmentation roughly fall into two categories: methods using an image sequence and methods using the stereo-motion paradigm:

- *Image sequence analysis.* These methods rely either on the estimation of the optical flow or on point-to-point correspondences. In the former case the relationship between 3-D motion of a rigid body and the observed 2-D velocities is explored. In the latter case, such constraints as the epipolar geometry and the trifocal tensor are used. Points satisfying the same type of constraint are assumed to belong to the same rigid object. Therefore, the problem of motion segmentation becomes the problem of grouping together points satisfying the same constraint [17, 13, 9, 19, 15]. For example, in [13] this grouping is carried out by a clustering algorithm which uses a posteriori likelihood maximization. Other approaches use such techniques as the Hough transform [17] or robust estimators which are used incrementally [19].
- *Stereo-motion analysis.* These methods combine the relationship between 3-D motion and image velocities described above with stereo constraints such as the epipolar constraint in order to disambiguate the inherent ambiguity associated with optical flow [22, 23, 21, 11].

In this paper we address both the problem of egomotion/motion discrimination and the problem of motion segmentation. The approach is based on the stereo-motion paradigm. The visual sensor consists of a pair of cameras rigidly attached to each other – a stereo rig. The geometry of this sensor remains rigid over time. This allows one to represent the motion of the sensor (egomotion) with a 3-D projective transformation which is conjugated to a 3-D rigid transformation. Because of this relationship between projective and rigid transformations, metric calibration of the stereo rig becomes an irrelevant issue [7]. 3-D

projective transformations are represented mathematically by 4×4 homogeneous full rank matrices, or homographies. The estimation of such a homography is based on the fact that a moving rig observes a static scene. When both the rig is moving and the scene is not rigid (is composed of both static and moving objects), current homography estimation methods cannot be applied anymore [1].

Therefore, the main contribution of this paper is a method for estimating 3-D projective transformations associated with the sensor's motion when the observed scene is composed of both static and moving objects. The output of this resolution technique consists in the estimation of a homography associated with egomotion as well as the classification of the observed 2-D point correspondences into a set of inliers and a set of outliers. The inliers are compatible with the observed egomotion while the outliers are not. Therefore, the latter are farther examined by a hierarchical clustering algorithm which operated in the image plane and over a long sequence of image pairs.

Organization

The remainder of this paper is organized as follows. The projective motion is defined in Section 2. In Section 3 we describe a robust estimator that enables to estimate the projective motion associated with the sensor's egomotion and the motion segmentation algorithm is described in Section 4. The stereo tracking algorithm that provides point-to-point correspondences through a sequence of image pairs is presented in Section 5. In Section 6 we show some experiments with real data and finally the conclusions are summarized in Section 7.

2. PROJECTIVE MOTION OF A STEREO CAMERA PAIR

Consider a 3-D point M which is observed by a stereo-rig from two different positions – position x and position y. Let (u_x, v_x), (u'_x, v'_x) be the image coordinates of the projections of M when the rig is in position x and (u_y, v_y) and (u'_y, v'_y) be the image coordinates of the projections of the same point when the rig is in position y. The associated homogeneous coordinates of these points are $\boldsymbol{x} = (u_x\ v_x\ 1)^\top$, $\boldsymbol{x}' = (u'_x\ v'_x\ 1)^\top$ and $\boldsymbol{y} = (u_y\ v_y\ 1)^\top$ and $\boldsymbol{y}' = (u'_y\ v'_y\ 1)^\top$, where $\boldsymbol{v}^\top$ denotes the transpose of $\boldsymbol{v}$.

Throughout this paper it is assumed that the epipolar geometry of the stereo rig is known. Since the stereo rig has a fixed geometry it is possible to associate a 3-D projective basis to the rig and when the rig moves, this projective basis *physically* moves with the rig. Therefore there is a projective basis associated with each position of the rig. Let $\mathbf{P}$ and $\mathbf{P}'$ the 3×4 projection matrices associated with the left and right cameras. According to what has just been said, these matrices are fixed. The following equations hold:

$$\begin{cases} \boldsymbol{x} \simeq \mathbf{P}\boldsymbol{X} \\ \boldsymbol{x}' \simeq \mathbf{P}'\boldsymbol{X} \end{cases} \text{and} \begin{cases} \boldsymbol{y} \simeq \mathbf{P}\boldsymbol{Y} \\ \boldsymbol{y}' \simeq \mathbf{P}'\boldsymbol{Y} \end{cases} \tag{1}$$

where "$\simeq$" denotes the projective equality. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are 4-vectors denoting the projective coordinates of the physical point M in the 3-D projective basis $\mathcal{B}_x$ associated with position x and the 3-D projective basis $\mathcal{B}_y$ associated with position y.

Equation (1) can be solved using the triangulation technique introduced by Hartley and Sturm [6] and which allows to estimate $\boldsymbol{X}$ and $\boldsymbol{Y}$. The relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is:

$$\mu\boldsymbol{Y} = \mathbf{H}\boldsymbol{X} \tag{2}$$

where μ is an unknown scale factor and $\mathbf{H}$ is a 4×4 full-rank matrix representing a *projective transformation* of the 3-D projective space. This matrix is defined up to a scale factor and therefore it has 15 degrees of freedom associated with it. Equation (2) provides three linear constraints in the entries of $\mathbf{H}$ and therefore with five points in general position it is possible to solve linearly for $\mathbf{H}$. However, in [1] and in [7] it was pointed out that the solution obtained with linear resolution techniques is quite noise-sensitive and at least 15 to 20 points are required in order to stabilize the numerical conditioning of the associated measurement matrix.

When the stereo-rig has fixed geometry and undergoes rigid motion, it has been shown in [2] that the projective transformation $\mathbf{H}$ is conjugated to a rigid transforma-

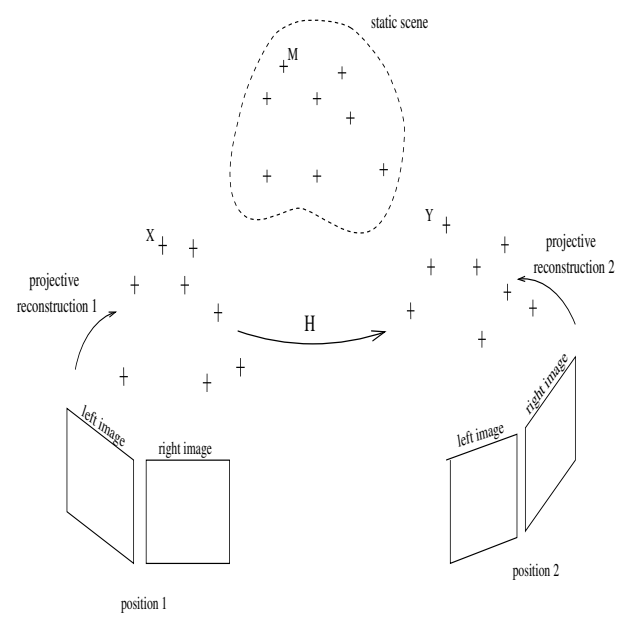

FIG. 1. Problem: a moving stereo rig observing a static scene

tion $\mathbf{D}$:

$$\mathbf{H} \simeq \mathbf{H}_u^{-1}\mathbf{D}\mathbf{H}_u$$

where $\mathbf{H}_u$ denotes the projective-to-metric upgrade. The above equation has been thoroughly studied in [8] and in [16]. $\mathbf{H}$ can be interpreted as a projective representation of the motion undergone by the camera pair – *projective motion* and may well be considered as an extension of affine motion [12] to the 3-D projective space:

DEFINITION 2.1. Consider a camera pair with known epipolar geometry which observes a 3-D rigid scene while it moves. The projective transformation between two projective reconstructions of the same 3-D scene obtained before and after the motion is called projective motion.

In theory, one can define projective motion without making the assumption that the stereo-rig has a fixed geometry. In practice, however, such an assumption is very useful. Indeed, the estimation of the epipolar geometry can be incrementally improved as new image pairs provide new left-to-right point correspondences.

3. ROBUST ESTIMATION OF PROJECTIVE MOTION

In order to estimate projective motion one may consider eq. (2) for $m \geq 5$ point correspondences. Therefore we obtain $3m$ linear equations which can be solved to determine the entries of $\mathbf{H}$. However, such a linear estimation method has two major drawbacks: (i) the method can deal neither with outliers (mismatched points) nor with non-rigid scenes (scenes that contain both static and moving objects), and (ii) the method minimizes an algebraic distance and hence it gives poor results for badly conditioned data. In particular, for $m = 5$ the method is very sensitive to noise [1].

To overcome these two drawbacks we introduce a new method based on robust estimation on one side and on minimizing an Euclidean error on the other side.

3.1. Robust methods in computer vision

Robust regression methods are widely used to solve various vision problems such as estimation of epipolar geometry [18] [24], estimation of the trifocal tensor [20] and so forth. Commonly used robust methods are M-estimators, least-median-squares (LMedS) [14], and random sample consensus (RANSAC) [3].

We wish to apply robust methods in order to compute projective motion $\mathbf{H}$ in the presence of outliers and/or non static scenes. Moreover, we would like to deal with situations where only 50 percents of the points composing the scene belong to static objects. Therefore we must choose a robust method which tolerates up to 50 percents of outliers. LMedS and RANSAC are the only methods tolerating such a rate of outliers. At first glance they are very similar. Data subsets are selected by a random sampling process. For each such subset a solution is computed and a criterion must be estimated over the entire data set. The solution yielding the best criterion is finally kept and used in a non linear process to refine both the solution and the sets of inliers/outliers. LMedS minimizes the median of the squares of the errors while RANSAC maximizes the number of inliers. Even if the criteria used by these two methods are quite different, in most practical applications, comparable results are obtained with both methods.

The main difference between LMedS and RANSAC resides in the outlier rejection strategy being used. The user must supply RANSAC with a threshold value while LMedS does not require such a threshold. Provided that this threshold is correctly selected, this feature enables RANSAC (i) to be more efficient in the presence of non homogeneous noise, (ii) to allow for 50 percents outliers and above, and (iii) to be more efficient because it can quit the random sampling loop as soon as a consistent solution is found. More detailed comparison/description of these algorithms can be found in [14].

In the framework of our application, the outliers may have two interpretations: they may either belong to independently moving objects or be "real outliers" (i.e. mismatched and/or mistracked points). As the stereo rig is observing a continuous flow of images, that means that the observed motions of independent objects may be small. In this case, we observe that LMedS often tends to choose an average model of all motions. As a result the set of selected inliers often contains some points of the moving objects and the set of outliers contains some points of the static scene.

The RANSAC algorithm performs better than LMedS in the framework of our application provided that the threshold for inliers/outliers selection t_c (in the inner loop of RANSAC) is carefully chosen. As a consequence we chose this algorithm for the estimation of the projective motion $\mathbf{H}$.

The choice of the threshold t_c is crucial for the success of RANSAC and is chosen such that $t_c^2 = 6.0\sigma^2$ where σ is the accuracy of the point location found by the stereo tracker described in Section 5. This threshold is observed to be often underestimated: the correct dominant projective motion is always found (contrary to what may happen using LMedS) but some points of the static scene may not be selected as inliers. However a completion is performed at the end of RANSAC by using a threshold t'_c slightly higher than t_c (with $6.0\sigma^2 \leq {t'_c}^2 \leq 9.0\sigma^2$). Using all the successive dominant motions in a sequence and averaging

the errors over time increases the performances of the robust algorithm (see Section 4).

Moreover the number of random samples N must be sufficiently large to guarantee that the probability of selecting a good subset is high enough, say this probability γ must satisfy $\gamma \geq 0.999$. The theoretical expression of this probability is $\gamma = 1 - (1 - (1 - \varepsilon_{out})^p)^N$ where p is the number of points that are necessary to compute a solution ($p = 5$ in our case) and ε_{out} is the number of outliers that are tolerated ($\varepsilon_{out} = 0.5$ in our case). By substituting all these numerical values in the above formula we obtain $N = 220$ as the minimum number of samples. However the expression of γ does not take into account the presence of noise on the inliers and a value of the order of 5 to 10 times larger than the theoretical one should be used for N. Hence for the robust method to be effective, the inner loop of the algorithm must be iterated at least 1000 times.

Moreover, remember that outliers have two physical meanings: they may well correspond either to mismatches or to moving objects. Therefore we must be able to distinguish between inliers and small motions. To conclude, the estimation step in the random sampling loop must be *fast* because it has to be run many times and must provide an estimation of $\mathbf{H}$ as *accurate* as possible.

3.2. A quasi-linear estimator

Let us devise an estimator for $\mathbf{H}$ that minimizes an Euclidean distance. In principle, such an estimator is non-linear because of the non-linear nature of the pinhole camera model. However, as described below, we have been able to devise a method which starts with a linear estimate of $\mathbf{H}$ and which incrementally and linearly updates the Euclidean error. Therefore, this method combines the efficiency of a linear estimator with the accuracy of a non-linear one. In practice it converges in a few iterations (2 to 3) and the solution thus obtained is very close to the solution that would have been obtained with a standard non-linear minimization method (see Figure 2).

The method described below can deal with a number of point matches equal or greater than 5. Within the robust method described above it is however desirable to use the minimal set of points – 5 points in our case.

With the notations already introduced in section 2 let $\boldsymbol{X}$ be the vector of 3-D projective coordinates obtained by reconstruction from its projections $\boldsymbol{x}$ and $\boldsymbol{x}'$ onto the first image pair. Matrix $\mathbf{H}$ maps these coordinates onto $\boldsymbol{Y}$ such that $\boldsymbol{Y} = \mu \mathbf{H} \boldsymbol{X}$, and matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$ reproject these coordinates onto the second image pair. Therefore we have the following estimated image points:

$$\alpha \hat{\boldsymbol{y}} = \mathbf{P}\mathbf{H}\boldsymbol{X} \tag{3}$$

$$\alpha' \hat{\boldsymbol{y}}' = \mathbf{P}'\mathbf{H}\boldsymbol{X} \tag{4}$$

The 3 vectors $\boldsymbol{y}$ and $\boldsymbol{y}'$ are defined up to a scale factor, α and α'. By dividing the first and second components of these vectors with their third component we get *estimated image positions* as opposed to $\boldsymbol{y}$ and $\boldsymbol{y}'$ which are *measured image positions*. The Euclidean distance between the measured point position $\boldsymbol{y}$ and the estimated point position $\hat{\boldsymbol{y}}$ is:

$$\varepsilon^2 = d^2(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \left(\frac{\hat{u}_y}{\hat{t}_y} - u_y\right)^2 + \left(\frac{\hat{v}_y}{\hat{t}_y} - v_y\right)^2 \tag{5}$$

with $\hat{\boldsymbol{y}}^\top = (\hat{u}_y \ \hat{v}_y \ \hat{t}_y)$ and $\boldsymbol{y}^\top = (u_y \ v_y \ 1)$.

Let us write matrix $\mathbf{H}$ as $\boldsymbol{h}$, a vector in $I\!R^{16}$ such that $\boldsymbol{h} = (H_{11} \ H_{12} \ldots H_{44})^\top = (h_1 \ldots h_{16})^\top$

By substituting eq. (3) into eq. (5) and with the notation:

$$w = \frac{1}{\hat{t}_y} = \frac{1}{(\mathbf{P}\mathbf{H}\boldsymbol{X})^{(3)}} \tag{6}$$

we obtain for the Euclidean error for the left image:

$$\varepsilon^2 = w^2 \left(\sum_{j=1}^{16} a_j h_j\right)^2 + w^2 \left(\sum_{j=1}^{16} b_j h_j\right)^2 \tag{7}$$

where the a_j and the b_j coefficients depend on $\boldsymbol{y}$, $\boldsymbol{X}$ and $\mathbf{P}$. Since we deal with an image pair the reprojected Euclidean error is:

$$e^2 = \varepsilon^2 + \varepsilon'^2 \tag{8}$$

For m point matches we obtain the following criterion:

$$E = \sum_{i=1}^{m} e_i^2 \tag{9}$$

$$= \sum_{i=1}^{m} \left(w_i^2 \left(\sum_{j=1}^{16} a_{ij} h_j\right)^2 + w_i^2 \left(\sum_{j=1}^{16} b_{ij} h_j\right)^2 + w_i'^2 \left(\sum_{j=1}^{16} a'_{ij} h_j\right)^2 + w_i'^2 \left(\sum_{j=1}^{16} b'_{ij} h_j\right)^2 \right) \tag{10}$$

It worth noticing that $\mathbf{H}$ is defined up to a scale factor. Then another constraint we can impose is $||\boldsymbol{h}|| = 1$. In order to find the matrix $\mathbf{H}$ or, equivalently, the unit vector $\boldsymbol{h}$ which minimizes the criterion E of eq. (10) we suggest the following incremental estimation method (notice that, by definition, the parameters w_i and w'_i are dependent of $\mathbf{H}$):

1. *Initialization*: Let $w_i(0) = 1$ and $w'_i(0) = 1$. Estimate $\mathbf{H}(0)$ using eq. (10);

2. *Evaluate* the parameters $w_i(k+1)$ and $\tilde{w}_i(k+1)$ using the current solution for $\mathbf{H}(k)$, i.e., eq. (6);

3. *Minimize* the criterion $E(k+1)$ of eq. (10) using standard weighted linear least-squares to estimate $\mathbf{H}(k+1)$;

4. *Stop test*: when $\frac{|E(k+1)-E(k)|}{E(k+1)+E(k)} < \varepsilon$ then stop, else return to step 2. Here we chose $\varepsilon = 10^{-4}$

The quasi-linear estimator requires low cost computation because each iteration of the loop only involves a standard weighted linear least-squares (based, in practice, on the Singular Value Decomposition technique). Moreover the quasi-linear estimator generally converges in two or three iterations.

Furthermore the quasi-linear estimator minimizes a geometric error and therefore it is less noise-sensitive than standard linear estimators [1] and appears to be well adapted when used in the inner loop of RANSAC: the error function associated with the inliers/outliers selection being defined by eq. (8).

3.3. Experiments with synthetic data

Experiments with simulated data are carried out in order to compare the quality of the results.

A synthetic 3-D scene consisting of 140 points is generated and placed at two different locations in the 3-D space. The 3-D points of each position are projected onto the cameras of a virtual stereo rig and Gaussian noise with varying standard deviation (from 0.0 to 1.6 pixels) is added to the image point locations. Data are normalized as described in [5] and three different methods are applied : the quasi-linear estimator, a standard linear method [1] and a classical non-linear optimization method, such as Levenberg-Marquardt, initialized with the quasi-linear estimator.

This process has been performed 100 times. The mean and standard deviation of the error function in eq. (8) for each method are shown on Figure 2. It shows that for noise under 1.0 pixel, the quasi-linear and the non-linear methods give very close results. It also shows the efficiency of the quasi-linear method in comparison with the standard linear method described in [1]. Furthermore it is faster: depending on the scene it is usually two to three times faster than the non-linear algorithm.

The convergence rates for varying levels of noise and number of points in the scene are studied for different methods: (i) the quasi-linear estimator (ii) the non-linear optimization method initialized with the standard linear estimator and (iii) the non-linear optimization method initialized with the quasi-linear estimator. Results are reported respectively on Figures 3, 4 and 5.

It shows that when the non-linear optimization method is initialized with the quasi-linear estimator, it always converges (as well as the quasi-linear estimator itself). On the contrary, when it is initialized with the standard linear estimator, it often falls in local minima. It can be explained by the fact that the quasi-linear estimator minimizes the same error as the non-linear optimization method, *i.e.* a geometric error in image space, whereas the linear estimator minimizes an algebraic error.

Therefore the quasi-linear estimator is a good compromise between accuracy and computation speed, fits very well with the estimation step of inner loops in robust algorithms like RANSAC or LMedS and provides a good initialization for non-linear optimization methods.

4. DETECTION OF MOVING OBJECTS

Let $\mathcal{P}_1, ..., \mathcal{P}_s, ..., \mathcal{P}_n$ be a sequence of image pairs gathered with a stereo rig. Let $\mathbf{H}_s$ be the projective motion associated with the sensor's egomotion between the image pairs $\mathcal{P}_s$ and $\mathcal{P}_{s+1}$.

The projective motions $\mathbf{H}_s$ are estimated with the robust estimator described in Section 3. In order to discriminate between static scene points from moving scene points we compute, for each tracked point, a global error over the whole sequence.

Let M be a 3-D point tracked through the pairs $\mathcal{P}_i$ to $\mathcal{P}_j$. For each t, $i \leq t \leq j$, eq. (8) defines the discrepancy $e(s)$ between the true motion of M and the motion predicted by $\mathbf{H}_s$. In other words, large $e(s)$ indicates that M is not a static point. In order to robustify this motion measure, we take the average $\tilde{e}^2$ of $e^2(s)$ over the image pairs in which M is observed, that is:

$$\tilde{e}^2 = \frac{1}{j-i+1} \sum_{s=i}^{j} e^2(s)$$

The observed scene points are then divided into two categories. Points M such that $\tilde{e} \leq t'_c$ are selected as static points (t'_c being the threshold defined in Section 3). The other points are considered as non-static points.

However these non-static points have two interpretations. On one side they may belong to moving scene objects and on the other side they may be "real outliers", i.e., mismatched and/or mistracked points.

In order to further classify the non-static points into points belonging to various moving objects and into real outliers we suggest to use data classification techniques. Generally speaking, such a technique groups the available data into several classes based on some metric. The data that we want to classify are the scene points denoted by M. Let M_1, ..., M_n be the non-static points found by the robust method just described.

The segmentation algorithm we propose here consists in grouping in the same cluster points being close to each other in all the sequence. However, since the 3-D reconstruction is projective one cannot define a metric in 3-D space. Therefore the distance we propose between points M is based on image point distance.

Let $x_k(s)$ and $x'_k(s)$ be the image projections of M_k respectively onto the left and right images of $\mathcal{P}_s$. If M_1 and M_2 are two points appearing together through the pairs $\mathcal{P}_i$ to $\mathcal{P}_j$, we define the distance between these two points as:

$$\delta(M_1, M_2) = \max_{i \leq s \leq j} \{d(\boldsymbol{x}_1(s), \boldsymbol{x}_2(s)), d(\boldsymbol{x}'_1(s), \boldsymbol{x}'_2(s))\}$$

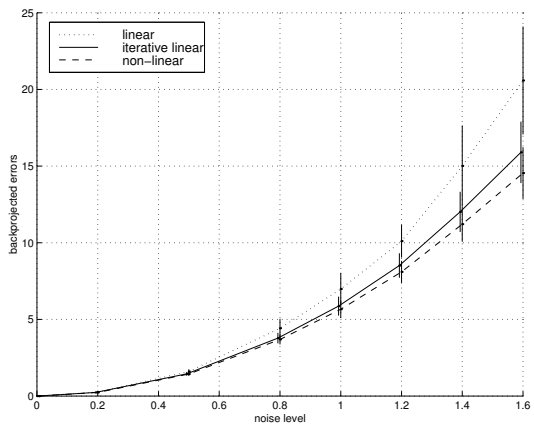

FIG. 2. Comparison between the different methods in the presence of image Gaussian noise.

nb. points	*noise level*						
	0.0	0.2	0.5	0.8	1.0	1.2	1.6
10 pts	100	100	100	100	100	100	99
30 pts	100	100	100	100	100	100	100
100 pts	100	100	100	100	100	100	100
300 pts	100	100	100	100	100	100	100

FIG. 3. Convergence rates (in %) of the quasi-linear estimator

nb. points	*noise level*						
	0.0	0.2	0.5	0.8	1.0	1.2	1.6
10 pts	100	100	87	68	68	63	61
30 pts	100	100	97	92	85	86	80
100 pts	100	100	99	99	89	91	85
300 pts	100	100	100	100	100	100	100

FIG. 4. Convergence rates (in %) of the non-linear method initialized with a standard linear estimator

This metric encapsulates the property that points which belong to the same moving object are close to each other in *all* the images in which they appear together.

In addition to the point-to-point metric defined above the classification algorithm needs a cluster-to-cluster metric. The latter is defined as a single linkage distance:

$$\Delta(\mathcal{C}_1, \mathcal{C}_2) = \min_{M_1 \in \mathcal{C}_1, M_2 \in \mathcal{C}_2} \delta(M_1, M_2) \tag{11}$$

where $\mathcal{C}$ denotes a cluster.

Therefore, the goal is to group within the same cluster those points which are close together and to throw out isolated points. Among the many data classification techniques available, the hierarchical clustering algorithm [10] with single linkage is well adapted for our purpose for several reasons. First, it does not need to know in advance the final number of clusters to be found, which means it does not need to know, a priori, either the number of moving objects present in the scene, or the number of real outliers. Second, it uses a simple stop procedure based on the minimum distance allowed between two clusters. Third, the method is fast because the cluster to cluster distances are efficiently updated.

At initialization there are as many clusters as there are points to be grouped. At each iteration of the algorithm the distances between all clusters are evaluated and the two clusters for which this distance is the smallest are merged together. The merging of clusters is thus repeated until the smallest distance is higher than a threshold t_s. It is worth noticing that if a dense matching is performed, a small value t_s can be confidently chosen.

Based on location only, the segmentation algorithm segments the scene into dense moving areas and contrary to many approaches it is able to successfully segment scenes in the presence of non rigid objects.

5. TRACKING WITH A RIGID STEREO RIG

In order to obtain point correspondences between many views, we propose a tracking algorithm that enables, from a sequence of image pairs gathered with a stereo rig, to (i) extract and track points along the sequence and (ii) incrementally estimate the epipolar geometry of the camera

nb. points	*noise level*						
	0.0	0.2	0.5	0.8	1.0	1.2	1.6
10 pts	100	100	100	100	100	100	100
30 pts	100	100	100	100	100	100	100
100 pts	100	100	100	100	100	100	100
300 pts	100	100	100	100	100	100	100

FIG. 5. Convergence rates (in %) of the non-linear method initialized with the quasi-linear estimator

pair. The points that are considered by the tracking algorithm are the interest points: these points are detected in all the images of the sequence by a corner detector [4].

A key idea of the approach is that, using a rigid rig, the epipolar geometry is constant over time and can therefore be estimated using a sequence of image pairs. The estimation of the epipolar geometry is well known to be subject to degeneracies when estimated from a single pair of images. Using several pairs of images enables to remove most of these degeneracies and therefore makes the computation of the epipolar geometry more stable and accurate.

The tracking algorithm is described below and illustrated on Figure 6.

Let $\mathcal{S}_1$ be a set of left-to-right correspondences associated with the image pair $\mathcal{P}_1$. These left-to-right correspondences are the projections of scene points $M_1 ... M_N$ that we want to track through the sequence.

Therefore the tracking algorithm consists in finding the sets $\mathcal{S}_s$ of left-to-right correspondences associated with the projections of the scene points $M_1 ... M_N$ onto the image pair $\mathcal{P}_s$.

The tracking is performed using an iterative approach. $\mathcal{S}_1$ is obtained using the robust estimator [24] and for all s, $\mathcal{S}_{s+1}$ is derived from $\mathcal{S}_s$ by the following way:

• For each match in image pair $\mathcal{S}_s$ we look for all the potential matches in image pair $\mathcal{S}_{s+1}$ such that they verify that (i) the four points associated with these two matches must have almost identical photometric profiles, and (ii) the epipolar constraint is verified. Based on these two constraints it is possible to select the best match in $\mathcal{S}_{s+1}$.

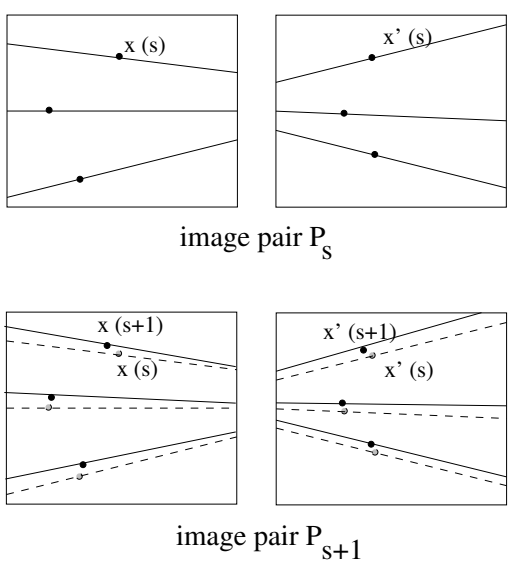

FIG. 6. Points tracked between two successive image pairs

• A robust computation of the epipolar geometry is performed using *all* the left-to-right correspondences available from $\mathcal{S}_1, ..., \mathcal{S}_{s+1}$ with the robust estimator [24]. This step enables to refine the estimation of the epipolar geometry over time.

• $\mathcal{S}_{s+1}$ is then updated: the correspondences which no longer satisfy the newly estimated epipolar geometry are removed (this case mostly arises when these points have been wrongly matched in the previous part of the sequence).

The tracking process then goes on until the end of the sequence and enables then to robustly:

• compute the epipolar geometry of the camera pair;
• match and track points between successive image pairs.

Moreover, an important feature of the tracking algorithm is that it enables to estimate the accuracy of point location σ introduced in Section 3.1 for the computation of the thresholds t_c and t'_c. σ is computed as the standard deviation of the errors of all the left-to-right correspondences of the sequence with respect to the epipolar geometry of the stereo rig.

6. EXPERIMENTS WITH REAL DATA

This section describes two experiments using real images. The same stereo rig has been used for each experiment . It consists of two similar cameras. The baseline is about 30 *cm.* and the relative angle between optical axes is between 5.0 *deg.* and 10.0 *deg.* (convergent configuration). The stereo rig has been moved while capturing sequences and the following process is applied to each sequence:

• Points are extracted and tracked with the tracking algorithm and the epipolar geometry of the stereo rig is estimated;
• The projective motions $\mathbf{H}_s$ associated with the sensor's egomotion are estimated;
• A global error $\tilde{e}$ is computed for each tracked point with respect to all $\mathbf{H}_s$ and used for selecting static/non-static points;
• the segmentation of outliers into different moving objects is performed.

Both sequences involve the same static scene: a robotic laboratory. In the first sequence, a single man is walking from left to right. In the second sequence, two men are walking (both from left to right). These stereo sequences (see Figures 7 and 8) consist each in nine image pairs that can be obtained at:

`http://www.inrialpes.fr/movi/people/Demirdjian/`

FIG. 7. Stereo sequence 1. Each column is an image pair of the sequence.

FIG. 8. Stereo sequence 2. Each column is an image pair of the sequence

FIG. 9. Detection of static points in sequences 1 and 2

It can be noticed that the static scene is composed of different levels of depth (walls, car, ...) and that the motions involved in the sequences are small. However in each experiment, the detection of the static points has been successful (see Figure 9). We can notice that in both sequences feet on the ground are sometimes detected as static but this can be explained by the fact that these feet have almost not moved in the sequence.

The threshold t_s required for the clustering algorithm has been fixed to 30 *pixels*. The evolution from the first to the last iteration of the clustering algorithm is shown on Figures 10 and 11. We can see that in each case, points

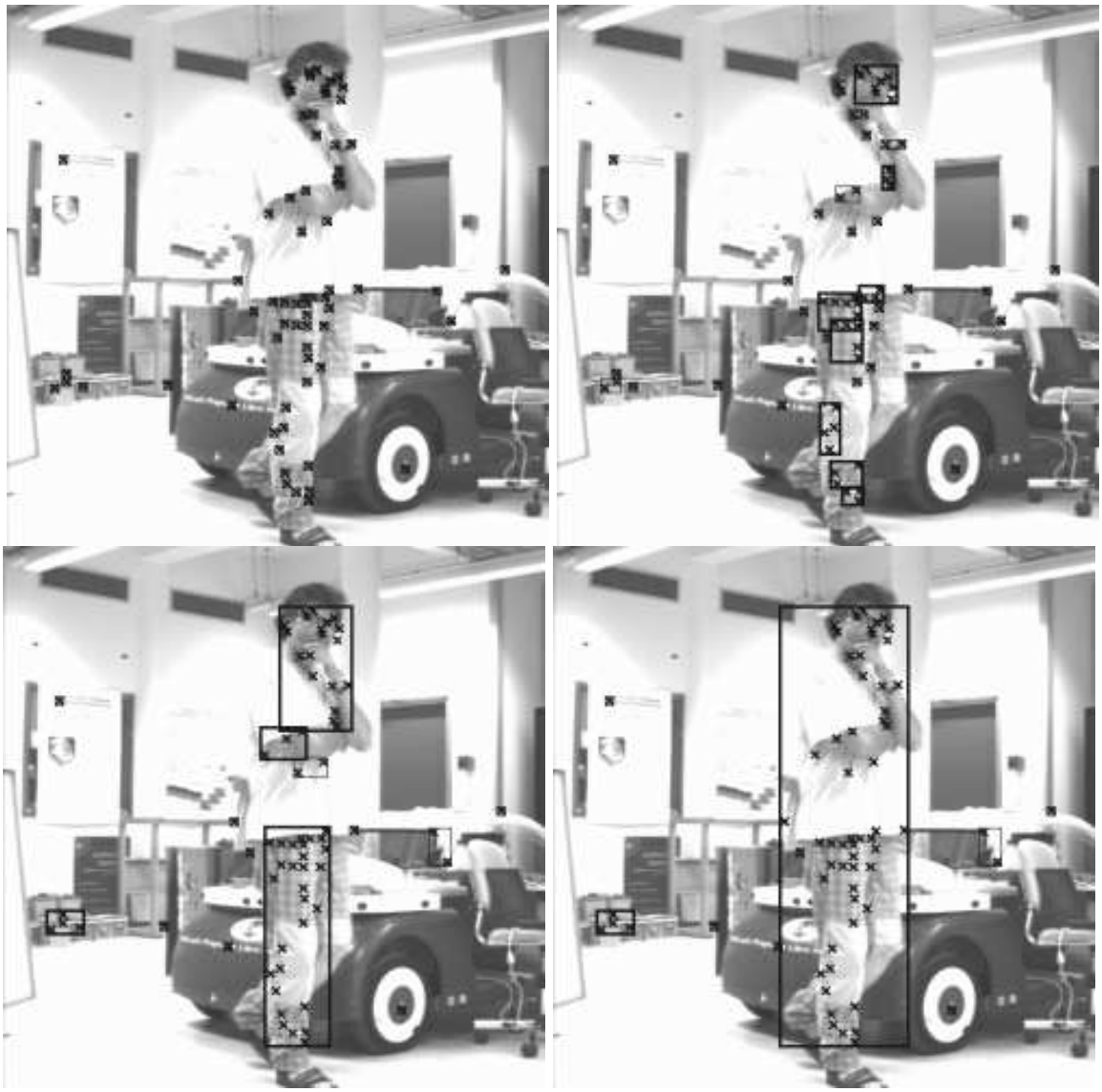

FIG. 10. Evolution of the clustering at iterations 1, 30, 62 and 69 (last) respectively

FIG. 11. Evolution of the clustering at iterations 1, 15, 37 and 42 (last) respectively

belonging to the same object are gathered in the same cluster. We can also notice that during the iterations of the clustering algorithm, the biggest clusters always correspond to parts of moving objects.

7. CONCLUSION

In this paper, we have described a method to detect moving objects with a moving stereo rig. Our approach is divided into three steps: (i) a stereo tracking process that simultaneously tracks points along a sequence of image pairs and robustly evaluates the epipolar geometry of the stereo rig, (ii) a robust *egomotion* estimation method based on 3D projective constraints, and (iii) moving object detection using image constraints.

We showed that, using a moving stereo rig, the detection of motion could be performed in a projective framework and therefore does not require any camera calibration. We improved the detection of static points in the case of small motions (i) by using RANSAC in conjunction with a quasi-linear estimator that accurately estimates projective motions (minimizing a geometric error) and (ii) by selecting inliers/outliers with respect to a global error estimated over the whole sequence.

We introduced a segmentation based on the detection of dense moving areas and we showed that this segmentation could be performed using a classical classification

gorithm has been chosen such that it benefits from the redundancy available from the multiple images of the observed sequence.

Finally the method needs no initialization and by this, we argue that the framework presented here can be used in many applications requiring an automatic moving object detection such as autonomous robotics or surveillance systems.

REFERENCES

1. G. Csurka, D. Demirdjian, and R. Horaud. Finding the collineation between two projective reconstructions. *Computer Vision and Image Understanding*, 75(3):260–268, September 1999.
2. F. Devernay and O. Faugeras. From projective to Euclidean reconstruction. In *Proceedings Computer Vision and Pattern Recognition Conference*, pages 264–269, San Francisco, CA., June 1996.
3. M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381 – 395, June 1981.
4. C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.
5. R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1064–1070, June 1995.
6. R. I. Hartley and P. F. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.
7. R. Horaud and G. Csurka. Self-calibration and Euclidean reconstruction using motions of a stereo rig. In *Proceedings Sixth International Conference on Computer Vision*, pages 96–103, Bombay, India, January 1998. IEEE Computer Society Press, Los Alamitos, Ca.
8. R. Horaud, G. Csurka, and D. Demirdjian. Stereo calibration from rigid motions. Technical Report RR-3467, INRIA, June 1998. Submitted to *IEEE Trans. on Pattern Analysis and Machine Intelligence*.
9. M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589, June 1998.
10. S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, (32):241–254, 1967.
11. P. J. Kellman and M. K. Kaiser. Extracting object motion during observer motion: Combining constraints from optic flow and binocular disparity. *Journal of the Optical Society of America A*, 12(3):623–625, 1995.
12. J. Koenderink and A. van Doorn. Affine structure from motion. *Journal of the Optical Society of America A*, 8(2):377–385, 1991.
13. W. MacLean, A.D. Jepson, and R.C. Frecker. Recovery of egomotion and segmentation of independent object motion using the em algorithm. In E. Hancock, editor, *Proceedings of the fifth British Machine Vision Conference, York, England*, pages 175–184. BMVA Press, 1994.
14. P. Meer, D. Mintz, A. Rosenfeld, and D.Y. Kim. Robust regression methods for computer vision: a review. *International Journal of Computer Vision*, 6(1):59–70, 1991.
15. J.M. Odobez and P. Bouthemy. robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.
16. A. Ruf and R. Horaud. Visual servoing of robot manipulators, part I: Projective kinematics. Technical Report RR-3670, INRIA, April 1999.
17. T.Y. Tian and M. Shah. Recovering 3d motion of multiple objects using adaptative hough transform. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1178–1183, October 1997.
18. P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. In P.S. Schenker, editor, *Sensor Fusion VI*, pages 432–442, Boston, 1993. SPIE volume 2059.
19. P.H.S. Torr and D.W. Murray. Stochastic motion clustering. In J.O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, volume 801 of *Lecture Notes in Computer Science*, pages 328–337, May 1994.
20. P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. In R.B. Fisher and E. Trucco, editors, *Proceedings of the seventh British Machine Vision Conference, Edinburgh, Scotland*, volume 2, pages 655–664. British Machine Vision Association, September 1996.
21. W. Wang and J.H. Duncan. Recovering the three-dimensional motion and structure of multiple moving objects from binocular image flows. *Computer Vision and Image Understanding*, 63(3):430–446, May 1996.
22. J. Weng, P. Cohen, and N. Rebibo. Motion and structure estimation from stereo image sequences. IEEE *Transactions on Robotics and Automation*, 8(3):362–382, June 1992.
23. J.W. Yi and J.H. Oh. Recursive resolving algorithm for multiple stereo and motion matches. *Image and Vision Computing*, 15(3):181–196, March 1997.
24. Z. Zhang, R. Deriche, O. D. Faugeras, and Q-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1–2):87–119, October 1995.