INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Utilisation des quasi-invariants géométriques pour l'appariement et la modélisation des images de segments de droite

Patrick GROS , Olivier BOURNEZ , Edmond BOYER

N 2608

Juillet 1995

PROGRAMME 4

ISSN 0249-6399

Utilisation des quasi-invariants géométriques pour l'appariement et la modélisation des images de segments de droite

Patrick Gros , Olivier Bournez , Edmond Boyer

Programme 4 — Robotique, image et vision
Projet Movi (Lifia–Cnrs)

Rapport de recherche n ° 2608 — Juillet 1995 — 40 pages

Résumé : L'appariement d'images consiste à retrouver dans deux ou plusieurs images les primitives qui représentent une même primitive de la scène observée. C'est un processus de base de la vision dès lors que l'on utilise plusieurs images. Dans cet article, nous proposons un algorithme d'appariement de deux images basé sur l'utilisation de quasi-invariants géométriques locaux. Une fois le premier appariement réalisé, une approximation projective du mouvement apparent et la géométrie épipolaire sont utilisés pour le compléter. Dans le cas où l'on dispose de plusieurs images, on peut, dans un premier temps, les apparier deux par deux. La dernière partie expose alors comment passer de ces appariements partiels à un appariement global de toutes les images. Les principaux avantages de la méthode présentée ici sont les suivants : la méthode montre une bonne robustesse dans le cas d'images bruitées, dans celui où les objets ne sont pas tous polyédriques, lorsque le mouvement apparent n'est pas infinitésimal, lorsque le mouvement de la scène n'est pas rigide, ou lorsque la caméra n'est pas calibrée ou que le mouvement qu'elle effectue entre les prises d'images n'est pas connu... La méthode présentée apparaît donc la seule utilisable dans de nombreux cas.

This work was supported by grants from Région Rhône-Alpes and Esprit- Bra project N° 6769 Second.

This work was done in the frame of a common project between CNRS, INRIA, Université Joseph Fourier and Institut National Polytechnique de Grenoble.

Unité de recherche INRIA Rhône-Alpes
46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1 (France)
Téléphone : (33) 76 57 47 77 – Télécopie : (33) 76 57 47 54

(Abstract: pto)

Using geometric quasi-invariants to match and model images of line segments

Abstract: Image matching consists in finding in two images the features which represent a same feature of the observed scene. It is a basic process of vision as soon as several images are used. In this paper, we propose a matching algorithm for two images, based on the use of local geometric quasi-invariants. Once a first match is made, a projective approximation of the apparent motion and the epipolar geometry may be used to complete it. In the case where we consider more than two images, these images may be matched two by two in a first step, and a global match is deduced in a second step: this is exposed in the last section. The main advantages of the method presented here are the following: it still works even if the images are noisy and the polyhedral approximation of the contours is not exact, if the apparent motion between the images is not very small, if the whole scene has not a single rigid motion, if the camera is not calibrated and the camera motion between the two shots is not known... It is thus usable in many cases where no other method is available.

1 Introduction

Image matching is one of the difficult basic problems of vision. It appears as soon as several images are considered. Let us take, for example, the problem of reconstruction: from two images of a three-dimensional object, we want to compute the geometry (i.e. the shape) of this object. We thus have to solve two problems:

1. given a point of the object, we have to find its projections in each image: this is the matching problem;

2. once both projections are known, the position of the object point has to be computed: this is the reconstruction itself.

The first of these two problems is usually set in a quite different way: in one of the two images, we consider a point m_1 which is the projection of the point M of the observed object. We search for the projection m_2 of M in the other image. m_1 and m_2 are said to be corresponding points, and this process in called image matching (to distinguish it from the image-model matching process).

Many methods have been published in the past to solve this problem. Many of them are designed to deal with grey level images: in this case, the algorithm tries to match every pixel of both images. Each pixel contains a value, the grey level, which encodes the light intensity received in one point of the sensor plane. These methods may be classified in three groups, following what is done in [9].

Correlation methods: they tend to measure by a correlation the similarity of two sub-images. With this aim in view, they consider a sub-image of the first image, i.e. a part of it, and they find the most similar sub-image of the second image. To limit the size of the search space, some constraints due to the way the images were taken may be used: epipolar geometry is a good example. Other limitations may be purely algorithmic: for example, the considered sub-images must have borders parallel to the image axis. Such methods have been used for a long time in photogrammetry [20, 10] and in computer vision [12]. Some improvements are regularly published [26, 18, 19, 15, 1, 11].

Relaxation methods: in these methods, a few matches are first guessed. Some constraints derived from these first matches are then used to compute other ones. This process is repeated until no new match can be done. Such algorithms were exhibited in the literature by Marr and Poggio [22, 23], with further improvements by Grimson [13, 14], and by Pollard et al. [28, 29].

Dynamic programming methods: in this case, the matching problem is set in the form of the minimization of a function of many discrete variables. Examples of such algorithms may be found in [3, 27].

When higher level features are available in images, such as edge points or line segments, the matching problem is slightly different. Firstly the features are less numerous and richer in information; particularly, they have geometric characteristics which may be used. Secondly these features are usually more reliable than the pixel grey level values. On the other hand, it is no more possible to obtain a dense match or reconstruction with such features. Another difficult problem is the relevance of the features: a non polyhedral shape represented by line segments may not be matched because the definition of the segments is very instable.

Several methods adapted to such features have already been proposed in the literature.

Prediction and verification methods: these methods are very close to the relaxation techniques. Some matches are first guessed; some constraints are deduced from them; they allow to verify the assumptions and to propose other matches. Such methods may be found in [24, 2].

Methods using sub-graph isomorphisms: if we consider the line segments present in an image as the edges or the vertices of a graph, we can use all the methods found in the literature about the computation of sub-graph isomorphisms. These methods imply not to have too much noise in the images, and do not use any geometric information. In addition, their complexity is often very high. Several examples may be found in [33, 17].

Methods using geometric invariants: if we consider that the images have been obtained by a pinhole camera, some projective invariants may be computed to characterize point and line configurations, and to match them according to the value of these invariants [31, 25]. Until now, these methods are mainly used in the case of planar or very simple scenes.

The method proposed in this paper concerns the case where the image features are line segments, possibly linked one to the other by their extremities. We use an approximation of the apparent motion of the features between the two images by a geometric transformation. We compute some invariants associated to this kind of transformation, for many configurations of points and segments in each image. These invariants are matched according to their values. A Hough like technique allow to

deduce the feature matches from the invariant matches. The invariants we use are called quasi-invariants because they are not invariant with respect to the apparent motion, but only with respect to an approximation of it. They only verify a local invariance criterion [4], but they have in general a better stability against noise than genuine invariants.

The main advantages of this method are the following:

1. it does not need any camera calibration, or that the camera motion or the epipolar geometry are known. In the opposite case, it would have allowed to use a prediction and verification technique. The apparent motion is not assumed to be very small, and the method is thus usable in cases where tracking methods would fail.

2. Between the two shots, the whole scene has not to have the same motion. The method still works if there are some mobile objects in the scene.

3. The method works with real noisy images. In such images, the polyhedral approximation of the contours is often not exact and all the segments are not stable. This is particularly the case when the scene contains curved objects. The topology of the graph formed by the segments is known to be very unstable. Using local quasi invariants makes our method robust towards this problem.

4. The use of geometric quasi-invariants associated with some special segment configurations allows not to consider all the possible configurations and then allows to keep the complexity much more smaller than that of sub-graph isomorphism techniques.

The method presented here then allows to deal easily with cases where no other method is usable in practice.

The paper is organized as follows. The first section describes the matching algorithm in the case of two images. The second one shows how this first match may be improved and completed using global constraints. The third one concerns the case of several images and shows how to go from image matching to object modeling.

2 Matching two images

2.1 Matching algorithm

This first section concerns the case where two images containing line segments are to be matched. These images are usually obtained from grey level images by edge extraction [6, 7], and polygonal approximation of the edge lines. Segment extremities may be used as junctions between segments, and each image may be seen as a graph of segments and vertices. FIG. 3 shows such images. The problem to solve is the matching problem between these segments and their extremities.

If we consider a coordinate system in each image, with the origin in the lower left corner of each image, it is clear that two corresponding features do not have the same coordinates in both images. The difference between them is called apparent motion, even if there is no temporal difference or order between the two images.

This apparent motion is not a classical geometrical transformation. As a matter of fact, two points of the scene can give a same projection in one image and two different projections in the other one; and no classical transformation can thus represent exactly this motion in the general case.

Our algorithm is based on the computation of an approximation of apparent motion by a single planar geometrical transformation, either a similarity or an affine transformation.

The different stages of this algorithm are the following ones.

1. One of the two kinds of approximation (extended Euclidean or affine) is chosen. The local invariants of some feature configurations are computed for the chosen kind of transformation: angles and length ratios for similarities, and affine coordinates for affine transformations.

2. These invariants are matched between the two images according to their values and to a few thresholds determined with respect of the image noise.

3. When two configurations have their invariants matched, the transformation (respectively a similarity or an affine transformation) which transforms the first configuration into the second one is computed.

4. All these transformations are represented in their parameter space: a four parameter space for similarities (two parameters for the translation, one for the rotation, and one for the scaling factor), and a six parameter space for affine transformations (four parameters for the linear part and two for the translation).

Correct matches define some transformations very close to the best approximation of apparent motion. The corresponding points in the parameter space are all in a small region of this space. On the other hand, the other matches are not correlated one with the other and their corresponding points are distributed in a very large region of space.

5. Thus we search for a small region of space where the point accumulation is maximal. This allows to compute an approximation of apparent motion and to separate the correct matches from the other ones.

6. Invariant matches give raise to configuration matches, and feature matches are deduced from the latter. In case of ambiguity, only the most probable matches are considered as correct: for example, if a feature a is matched 5 times to another feature b, and only once to a third feature c, the match (a, b) is the only one considered as correct.

These stages are explained in detail in the next paragraphs: the first one studies the case of the approximation by a similarity, the second one the case of the approximation by an affine transformation, and the third one shows some experimental results.

2.2 Approximation by a similarity

Similarities are the result of the composition of an isometry and a scaling. In the latter, we consider only direct similarities, i.e. the composition of a rotation, a translation and a scaling. These transformations define a geometry which can be called extended Euclidean (since Euclidean classically excludes scalings).

The two basic invariants of this geometry are the angle and the length ratio defined by any two segments. These invariants are also projective quasi-invariants [4]: they are stable by an infinitesimal projectivity and then by a "small" projective transformation. This explains the results obtained: even if the approximation of apparent motion by a similarity appears to be coarse, the obtained matches are correct as soon as the camera motion is not too large.

Stage 1: computing invariants

Similarity invariants may be computed from two segments or three points. To maintain a low complexity of the configuration enumeration and to insure a better validity of the approximation, only the pairs of segments having an extremity in

common are considered. Let S_1 and S_2 be two such segments and P be their common extremity. We denote P_1 and P_2 their other extremity (see FIG. 1).

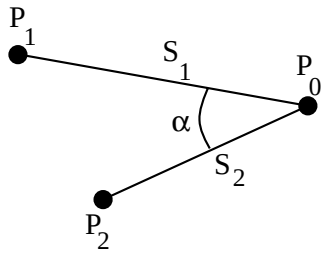

FIG. 1 - *Definition of invariants from two adjacent segments.*

Such a configuration is characterized by its angle α, a length ratio ρ, and a reliability measurement p, called its weight in the latter.

$$\begin{aligned} \alpha &= \arccos \frac{\overrightarrow{P_0P_1} \,.\, \overrightarrow{P_0P_2}}{\| \overrightarrow{P_0P_1} \| . \| \overrightarrow{P_0P_2} \|} \\ \rho &= \| \overrightarrow{P_0P_1} \| / \| \overrightarrow{P_0P_2} \| \\ p &= \| \overrightarrow{P_0P_1} \| + \| \overrightarrow{P_0P_2} \| \end{aligned}$$

To avoid any further ordering problem, the segments S_1 and S_2, and then P_1 and P_2, are chosen by convention such that the oriented angle $(\widehat{S_1S_2})$ is sharp and positive. This allows to simplify the further comparison tests between the invariants, and between the configurations. The use of a weight is justified by the fact that noise and texture usually give raise to short segments and that, on the other hand, long segments are often more significant and characteristic. These weights are especially used in the fifth stage of the algorithm.

Stage 2: matching invariants

The configuration invariants from the first image are compared with those from the second image. To realize this, we use two thresholds: $\alpha_{max} = 20°$ and $\rho_{max} = 1,5$. These thresholds are chosen with respect to the noise present in both images. A configuration of segments of the first image, whose invariants are (α_1, ρ_1), is matched to

a configuration of the second image, whose invariants are (α_2, ρ_2), if three conditions are satisfied:

$$|\alpha_1 - \alpha_2| < \alpha_{max} \qquad \rho_2/\rho_1 < \rho_{max} \qquad \rho_1/\rho_2 < \rho_{max} \tag{1}$$

At this stage, a configuration is often matched to one or two tens of other configurations.

From a practical point of view, this comparison can be sped up using the following method: the invariants (α_1, ρ_1) of the first image are sorted with respect to the value of α_1. The same is done for the invariants of the second image. The two lists obtained may thus be compared by a linear rather than a quadratic enumeration.

Stage 3: computing similarities

Among all the matches computed at the previous stage, only a small proportion is correct. To separate it from the incorrect matches, we use the fact that the correct matches correspond to the same apparent motion and thus define some approximations very close to that motion.

Two matched configurations provide enough information to compute the similarity which transform the former into the latter. Indeed, every match provides three point matches and two are sufficient to compute a direct similarity.

Let us consider two pairs of matched segments, whose extremities are respectively denoted (P_0, P_1, P_2) and (Q_0, Q_1, Q_2) (with the ordering conventions presented before). Let k, θ, t_x and t_y be the parameters of the similarity which transform the first configuration into the second one. They may be computed as follows:

$$\begin{aligned} k &= \frac{1}{2}\left(\frac{\| \overrightarrow{Q_0Q_1} \|}{\| \overrightarrow{P_0P_1} \|} + \frac{\| \overrightarrow{Q_0Q_2} \|}{\| \overrightarrow{P_0P_2} \|}\right) \\ \theta &= (\overrightarrow{P_0P_1}\widehat{Q_0Q_1}) + \frac{1}{2}\left[(\overrightarrow{Q_0Q_1}\widehat{Q_0Q_2}) - (\overrightarrow{P_0P_1}\widehat{P_0P_2})\right] \\ \begin{pmatrix} t_x \\ t_y \end{pmatrix} &= \overrightarrow{PQ_0} \quad \text{where} \quad P = H_k \circ R_\theta(P_0) \end{aligned}$$

R_θ represents the rotation of angle θ whose center is the origin, and H_k the scaling of factor k whose center is the origin too.

The use of a unique pair of points to compute t_x and t_y is justified by the fact that the intersection point of two segments is often more reliable than the other two extremities which may be linked to no other segment.

The similarity computed with this method provides an apparent motion approximation for the two configurations it is computed from.

Stages 4 and 5: filtering similarities

For every pair of matched configurations, a similarity is computed using the previous method. Each of them is defined by its four parameters (t_x, t_y, θ, k), and may be represented by a point in $\mathbb{R}^4$. Furthermore, each of these points is weighed by the sum of the weights of the configurations used to compute this similarity.

Incorrect matches are generally not correlated one with another: the corresponding similarities are thus represented by points distributed in a large region of $\mathbb{R}^4$. On the other hand, correct matches all define good approximations of apparent motion, and give points gathered in a small region of space: the aim of this stage is to search for this point or region.

To realize this, we use some boundaries t_x^{max}, t_y^{max}, θ^{max} and k^{max}. For each similarity $(t_x^0, t_y^0, \theta^0, k^0)$, we compute the sum of the weights of the similarities (t_x, t_y, θ, k) which verify:

$$|t_x - t_x^0| < t_x^{max} \qquad |t_y - t_y^0| < t_y^{max}$$
$$|\theta - \theta^0| < \theta^{max} \quad k^0/k^{max} < k < k^0.k^{max}$$

This defines an accumulation around each point in $\mathbb{R}^4$ and the heaviest one is considered as the correct one: only the similarities and the matches which give raise to a point in this heaviest accumulation are considered as correct.

From a practical point of view, the points in $\mathbb{R}^4$ are sorted according to their first coordinate: this allows to limit the number of comparisons. The four thresholds may be taken equal to 15 pixels, 15 pixels, 20 degrees and 1.5.

Stage 6: matching features

Each match of two configurations (P_0, P_1, P_2) and (Q_0, Q_1, Q_2) provides three possible point matches: (P_0, Q_0), (P_1, Q_1), (P_2, Q_2). Let us consider all the possible matches in which a point P is involved: only the matches forming the majority are assumed to be correct. The other ones, and the configuration matches from which they are deduced are suppressed. For example, from four point matches (P, Q_1), (P, Q_1), (P, Q_1), (P, Q_2), the match between P and Q_1 is assumed to be correct. The match (P, Q_2) and the configuration match from which (P, Q_2) was deduced are suppressed.

2.3 Approximation by an affine transformation

Let us now consider the case where apparent motion between the two studied images is approximated by an affine transformation, rather than by a similarity. The basic invariant of affine geometry is the length ratio of collinear segments. What was done with similarities and Euclidean invariants may be extended to affine transformations using invariants based on this basic affine invariant. In this section, we discuss only what differs between the similarity approximation and the affine approximation, but the general algorithm remains the same.

Stage 1: computing invariants

Three segments or four points are needed to compute an affine invariant, in the general case where these points or segments are not collinear. To limit complexity, we consider only two kinds of configurations, called Z or Y configurations.

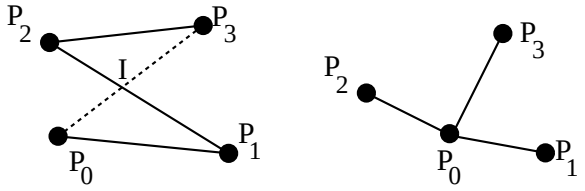

Fig. 2 - *Definition of invariants from a Z (left) or Y (right) three-segment configuration.*

From a Z configuration, such as that shown on Fig. 2, two invariants may be defined.

$$\rho = \| \overrightarrow{P_3 I} \| / \| \overrightarrow{P_0 I} \| \quad \text{et} \quad \sigma = \| \overrightarrow{P_2 I} \| / \| \overrightarrow{P_1 I} \|$$

To avoid any further ambiguity, the configuration points are labeled such that ρ is greater or equal to 1.

In the case of a Y configuration, as shown on the right of Fig. 2, the invariants are the affine coordinates of P_0 with respect to the three other points P_1, P_2 and P_3. These coordinates are defined as follows:

$$\begin{cases} a_1 P_1 + a_2 P_2 + a_3 P_3 = P_0 \\ a_1 + a_2 + a_3 = 0 \end{cases}$$

The configuration points are labeled in such a way that a_1 is smaller than a_2 and a_2 is smaller than a_3.

In both cases, the configuration is weighed by the sum of length of the three segments.

Stage 2: matching invariants

The matching process in the affine case follows exactly the same algorithm as in the case of similarities. The only difference comes from the matching conditions (1). For two Z configurations, whose invariants are respectively (ρ_1, σ_1) and (ρ_2, σ_2), the conditions are:

$$\rho_1 < k\rho_2 \quad \rho_2 < k\rho_1 \quad \sigma_1 < k\sigma_2 \quad \sigma_2 < k\sigma_1$$

For two Y configurations, whose invariants are respectively (a_1, a_2, a_3) and (b_1, b_2, b_3), they are:

$$|a_1 - b_1| < \varepsilon \quad |a_2 - b_2| < \varepsilon \quad |a_3 - b_3| < \varepsilon$$

In our experiments, k was taken equal to 2.2 and ε to 1.5.

Stage 3: computing affine transformations

Each configuration match provides four point matches. From these matches, it is possible to compute the best affine transformation which transforms the first configuration into the second one. Let T be that transformation. T may be computed using a least square method.

Let P_i^j be the i-th point of the j-th configuration and $(x_i^j y_i^j 1)^t$ be its homogeneous coordinates vector. Let us denote $\begin{pmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{pmatrix}$ the homogeneous matrix associated to T. We have the following equation:

$$\begin{pmatrix} x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ y_1^2 & y_2^2 & y_3^2 & y_4^2 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1^1 & x_2^1 & x_3^1 & x_4^1 \\ y_1^1 & y_2^1 & y_3^1 & y_4^1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

which can also be be written in the following form:

$$\begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ x_4^2 \end{pmatrix} = \begin{pmatrix} x_1^1 & y_1^1 & 1 \\ x_2^1 & y_2^1 & 1 \\ x_3^1 & y_3^1 & 1 \\ x_4^1 & y_4^1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ t_x \end{pmatrix} \text{ and } \begin{pmatrix} y_1^2 \\ y_2^2 \\ y_3^2 \\ y_4^2 \end{pmatrix} = \begin{pmatrix} x_1^1 & y_1^1 & 1 \\ x_2^1 & y_2^1 & 1 \\ x_3^1 & y_3^1 & 1 \\ x_4^1 & y_4^1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_3 \\ a_4 \\ t_y \end{pmatrix}$$

The least square resolution of these systems, using for example the Singular Value Decomposition [30], provides as a solution the transformation T which minimizes the expression $\sum_{i=0}^{3} \|P_i^2 - TP_i^1\|^2$. We chose this solution T as the best approximation of the apparent motion for the two considered configurations.

Last stages

As it was done with the similarities, affine transformations may be represented as points in $\mathbb{R}^6$. These transformations may be filtered in order to separate the correct matches from the other ones, and feature matches are deduced from configuration matches. To compare affine transformations, we use the following equations:

$$|a_i - a'_i| < \varepsilon_i \quad |t_x - t'_x| < \varepsilon_x \quad |t_y - t'_y| < \varepsilon_y$$

In practice, we chose ε_i equal to 1 and ε_x and ε_y respectively equal to 10 and 20.

2.4 Experimental results

In this section, the matching algorithm just described is illustrated by some results obtained with real images: this allows to validate the assumptions and approximations done.

FIG. 3 shows the result of the matching of two images. The approximation by similarities was used in the two upper images, and the matched points have the same number. The approximation by affine transformations was used in the two lower images (that are the same). There is no correspondence between the numbers in the upper images and the numbers in the lower ones.

The left image contains 133 points, the right one 106. We can consider that about 40 among these points are characteristic of the object. The other ones are due to noise, background or texture. 25 matches are obtained using the similarities , and 24 using affine transformations. All these matches are correct, although two of them in the case of similarities correspond to the image border. On the other hand, some points clearly belonging to the object are not matched. This lack is due to noisy segments (for example segments split in two parts by a vertex), or to regions where the approximation of apparent motion by a similarity or even an affine transformation is too rough (due to perspective effects for example).

These results clearly show that the quasi-invariants are robust enough, even if the apparent motion is visibly far from an exact similarity or an affine transformation.

FIG. 4 shows in the same manner other results with two other images. These images contain non polyhedral objects, for which the approximation of edges by

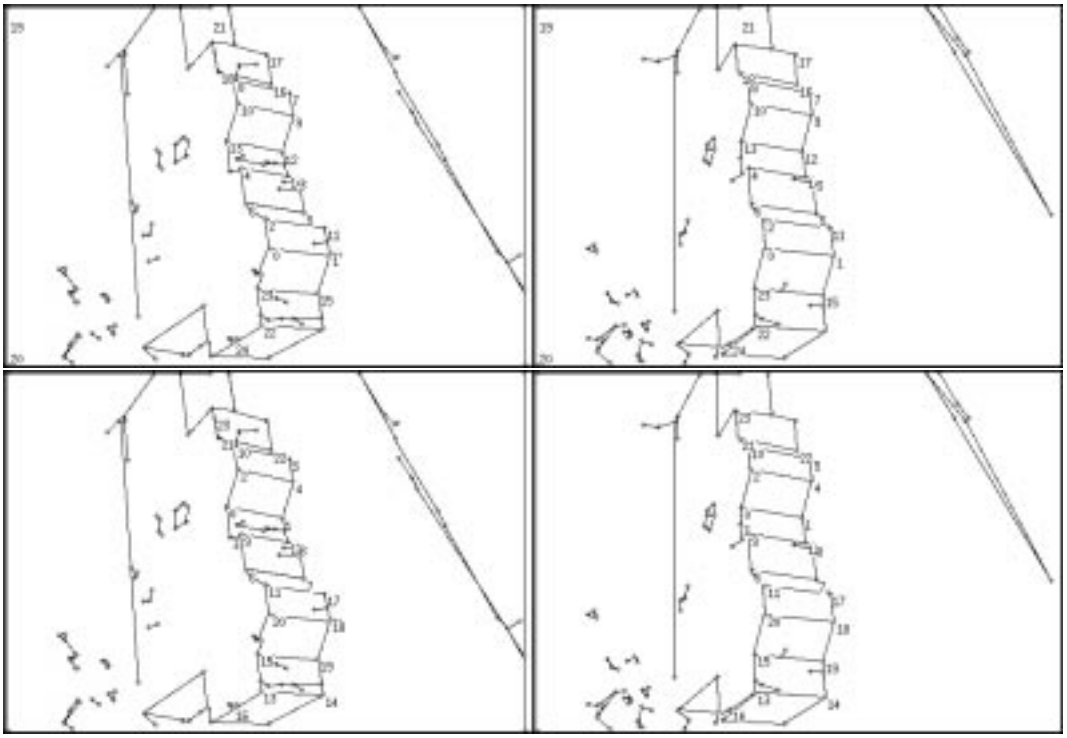

FIG. 3 - *A First example of matching, using similarities (top), and affine transformations (bottom).*

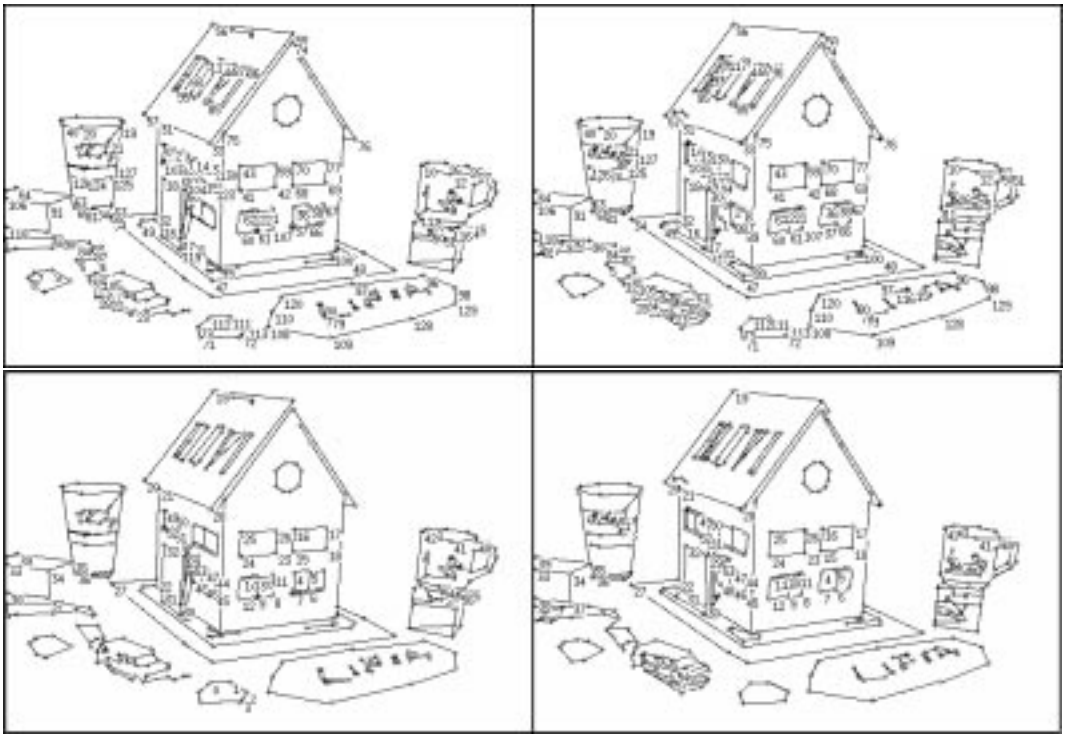

FIG. 4 - *Another example of matching using similarities (top) and affine transformations (bottom).*

segments is not stable and very noisy. The first image contains 322 points, the second one 352. We may consider that about 80 points are characteristic of the house. With similarities, 131 points were matched, with 15 outliers, and 15 errors at the windows: the windows are represented by repetitive rectangles and errors often occur in such situations. With affine transformations, 58 points are matched, with 8 outliers and 4 errors due to the windows.

2.5 Conclusion

The results obtained show the validity of the algorithm, especially its ability to deal with real noisy images containing polyhedral and even a few curved objects. Of course, the obtained match is not complete, and the next section shows how to improve these first results.

The method appears to be general and not linked to a special kind of local quasi-invariants. To demonstrate that, we used other kind of invariants, like Euclidean invariants computed from three segments. The results are similar.

On the other hand, another idea would be to use projective invariants, computed using cross ratios. In this case, the problem is that the computation of such invariants, and that of an homography between two configurations is very sensitive to noise, and it makes the search of an accumulation point in $\mathbb{R}^8$ extremely difficult. In fact these invariants require more precise data than ours. The use of a parametric corner detector may be a solution.

As a conclusion, we can say that our algorithm is robust, and that its robustness comes from the use of simple quasi-invariants. They appear to be a powerful tool to deal with images where noise forbids to use exact and more sensitive invariants. They make a big part of the originality and of the success of our method.

3 Improving image matching

The algorithm described in the previous section allows to do a first match between two images without any a priori information. The results may contain a few incorrect matches and miss many other matches: apparent motion approximation by a similarity or an affine transformation is too restrictive for some image regions.

Once a first match is done, other finer tools can be used to evaluate the validity of already computed matches, and to find new ones. A first tool is the approximation of apparent motion by a 2D homography. When the observed object is planar, this is no more an approximation, but an exact computation.

A second tool is epipolar geometry. This geometry is not a point to point correspondence, but a point to line one. On the other hand, it is exact for planar and non planar objects. Computation and use of these two tools are described in this section.

These tools may be particularly interesting in the case where the scene contains a mobile object in a fixe background. As soon as the match obtained is not too poor, the mobile object will fail to define the same epipolar geometry or homography as the rest of the scene. This opens new directions of applications to the matching method.

3.1 Projective approximation of apparent motion

As the use of a similarity or of an affine transformation to approximate apparent motion seems to be too rough in some image regions, a first idea is to use a transformation which has more parameters. In Klein hierarchy [21], the first class which contains the affine transformations is that of projective transformations (homographies). At least four points are necessary to define an homography between two images. The matching algorithm described in the previous section provides, in general, more than four correspondences, and this redundancy allows to suppress the computation instability described in the conclusion at paragraph 2.5.

Several methods are usable to compute an homography between two images from at least four matches.

Linear least squares method

The first and simplest method consists of a linear least squares computation, such as that done in stage 3 of paragraph 2.3.

Let (P_i^1, P_i^2) be the matches obtained previously, and T be the homography of homogeneous matrix $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$, which minimizes $\sum_i \|P_i^2 - T.P_i^1\|^2$. If the equations $P_i^2 = T.P_i^1$ are written as it was done in paragraph 2.3, T can be obtained by solving the system of equations by singular value decomposition.

Robust least median squares method

Real images are always noisy and the use of thresholds in the matching algorithm to take this noise into account provides some errors in the result. The least squares

method uses all the matches, and then all the errors. A few errors are sufficient to cause inaccuracy of the computed transformation, even at the points given to compute it. A least median squares method can correct this drawback by detecting outliers. The algorithm is the following:

1. Four point matches are chosen randomly.
2. The homography T defined by these four matches is computed using the linear method.
3. The error $\|P_i^2 - T.P_i^1\|^2$ is computed for every match (not only the four previous ones). These errors are sorted, and the value ε of one of the errors is associated with T, such that $x\%$ of the errors are smaller than ε and the other ones are greater. ε is the errors' median if x is equal to 50.

This process is repeated such a number of times that the probability to choose at least one set of four correct matches is very high. The final estimation of the homography is the transformation T whose associated value ε is the smallest.

This method allows to detect outliers in the data. As a matter of fact, the matches giving an error three times greater than the median, or twice as great as ε if x is greater than 50, may be considered as outliers and eliminated. It is clear that the best estimation of the homography is not based upon these outliers, and the final result is good enough for many applications. To have a more accurate result, it is possible to finish this computation using the next non linear method.

The number of times the three stages 1, 2, and 3 must be repeated and the percentage x depend upon the data. x indicates an estimated rate of outliers in the data. 50% may be taken as a default value, but the result is of course better with a better estimation of the real rate.

If y is this real rate of incorrect matches in the data, the probability to get at least one configuration of n correct matches among m is equal to [34]:

$$P = 1 - [1 - (1 - y)^n]^m$$

If we assume that $y = 40\%$ and if we want $P > 0.99$, 34 iterations are enough. As indicated in [34], this method can be improved by requiring that the chosen points are well distributed in the images. According to our experimental results, the additional time this requires is hardly justified by the improvement of the result.

Non linear optimization method

The two previous methods are not symmetric with respect to the images. As a matter of fact, the two images do not play the same role in the criterion $\sum_i \|P_i^2 - T.P_i^1\|^2$ to be minimized, and this may have consequences on the result.

This expression may be symmetrized:

$$\sum_i (\|P_i^2 - T.P_i^1\|^2 + \|P_i^1 - T^{-1}.P_i^2\|^2)$$

A linear resolution is no more possible. The result of the first or the second method may be used as a first estimation of the solution for a non linear optimization method, such as that of Levenberg-Marquardt [30].

The linear solution provides in general a good enough approximation of the real solution to insure a convergence of the algorithm in only a few iterations. On the other hand, the elimination of outliers by the least median squares method seems necessary to obtain accurate results in most of the cases. Finally, the non linear method is sensitive to the quality of the first estimation of the solution: this is yet another argument for the use of the robust method rather than that of the linear one.

Conclusion

In our experiments, we used the least median squares method which allow to deal with outliers. It gives much better results than the simple linear method. On the other hand, our data are not precise enough to have a real improvement of the results when using the non linear optimization method. It is sure that in all cases, the use of homographies gives better results with scenes which are not too deep.

3.2 Epipolar geometry computation

Epipolar geometry defines a relation between two images of the same scene. The fundamental matrix F sums up the information of epipolar geometry. This matrix has a 3×3 dimension and a rank 2; it verifies ${}^tP^2.F.P^1 = 0$ for every couple of matched points (P^1, P^2), represented by their homogeneous coordinates. Many methods of computation of this matrix from point matches have been proposed in the literature. They may be classified in three groups:

- linear methods, which allow to compute epipolar geometry directly from point matches;

- robust methods, which use the previous ones, and allow to detect and eliminate outliers.
- non linear methods, which allow to compute a very accurate result from a first estimation of the solution.

Here are short descriptions of the main methods:

Simple linear method: this first method consists in considering the problem as a simple system of linear equations. The fundamental matrix F is defined up to a scale factor and one of its coefficient may be taken equal to 1. The matrix is then found by solving the equations ${}^tP^2.F.P^1 = 0$, for at least eight different point matches, using the SVD method. This method has a few drawbacks: the system to be solved is ill-conditioned and the rank the computed matrix is often equal to 3 as soon as the data are not exact.

Hartley's method: the previous method may be significantly improved in several ways [16]: first, changing the coordinate basis allows to correct the ill-conditioning of the initial system. Second, when the system is solved by SVD, if the smallest singular value is put equal to 0 before the computation of the result, the obtained matrix will be singular.

Boufama's method: a new method was recently proposed [5]. Based on a different formulation of the problem, it allows by a simple resolution to find a singular fundamental matrix. In the case of a planar scene, the two previous methods provides meaningless results. Boufama's method provides in this case correct pencils of epipolar lines: every point of every epipolar line has its corresponding point on the corresponding epipolar line.

Least median squares method: as it was done with the homography computation, it is possible to use a robust method using the least median squares algorithm. The principle is the same in the case of epipolar geometry computation. This method may be improved in several ways, concerning the way the points are chosen, or the linear method used...Such improvements are described in [34].

Non linear optimization method: the first method could provide a non singular matrix. To correct this, the criterion $\sum_i \| {}^tP_i^2.F.P_i^1 \|^2$ may be minimized under the constraint $\det F = 0$. This may be done by using a non linear optimization algorithm such as that of Levenberg-Marquardt [30].

The main drawback of this method is due to the instability of the function $F \mapsto \sum_i \| {}^tP_i^2.F.P_i^1 \|^2$, which admits many local minima; the correct solution, i.e. the one that corresponds effectively to the camera motion, is even not always the global minimum of the function as soon as the images are noisy. To use this method, one must eliminate the outliers with the robust method first, and give a very good first estimation of the result.

For our matching purpose, we generally use the least median squares method with the linear Boufama's method. They provide robust epipolar pencils which can be efficiently used for matching. On the other hand, the epipoles are not used and their position has not to be determine precisely.

3.3 Corrections and improvements

The tools described in the previous paragraphs allow to significantly improve the results obtained with the matching algorithm presented in the first section of the paper.

- Computing epipolar geometry and the best approximation of apparent motion by an homography using robust methods allows to detect outliers. The mean-error found when computing the homography may be used to decide whether the scene is planar or not. In the former case, Boufama's method or the robust method associated with Boufama's one should be used to compute epipolar geometry. In the latter case, homography will be given less importance since it can not accurately approximate apparent motion.

- New matches may be deduced from those already done. Let s_1 be a segment which extremities are denoted a_1 and a_2. If only one of these to points, say a_1, is matched to a point b_1 of the other image, we can search for another point b_2 of this image, such that there exists a segment between b_1 and b_2 and, that a_2 and b_2 are compatible with respect to epipolar geometry and to the homography approximating apparent motion. If a_1 and a_2 are not matched, it is possible to look for a segment s_2 whose extremities b_1 and b_2 respect the epipolar and homography constraints with a_1 and a_2.

- Even more, we can search, among all the unmatched points of both images, for all the couples (a, b) which satisfy the epipolar and the homography constraints. This last method may give incoherent results with respect to image topology and should be used with much care.

FIG. 5 and 6 show results obtained from the matches shown on FIG. 3 and 4: epipolar geometry and the homography were computed using a robust method, and the outliers were detected and eliminated. Every error was suppressed. 24 and 90 matches respectively remain in these examples.

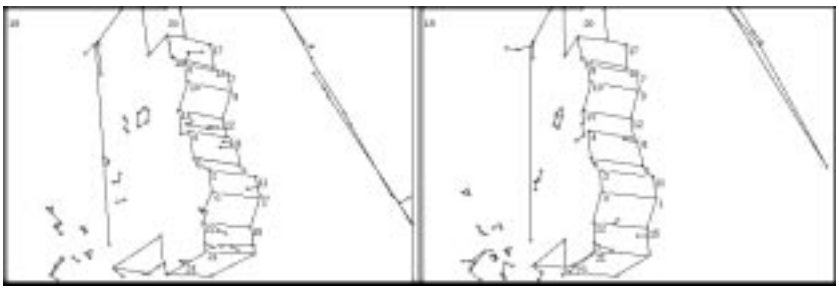

FIG. 5 - *The matches of* FIG. *3 after error suppression.*

FIG. 6 - *The matches of* FIG. *4 after error suppression.*

FIG. 7 and 8 show the results obtained after the improvement stage. The match is more complete: in the presented examples, 45 and 262 matches are respectively obtained.

It is of course possible that a few errors appear during this last stage. But they are not too rough from a geometrical viewpoint, because they must respect the two constraints.

On the other hand, some points are still not matched. This comes from the homography which was computed from a few match and which gives a good approximation of apparent motion only in a part of the image. In the remaining part, it forbids any new match.

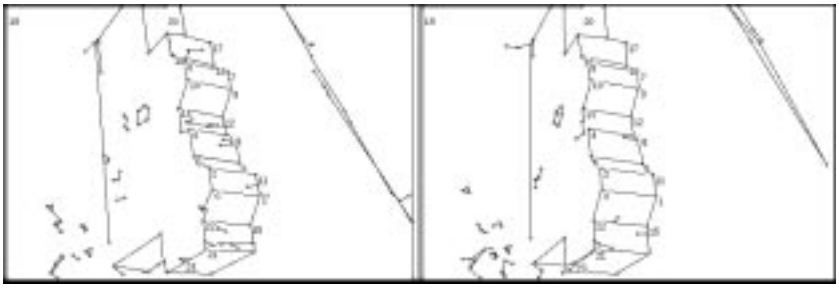

Fig. 7 - *Improvement of the matches shown on* Fig. *3*

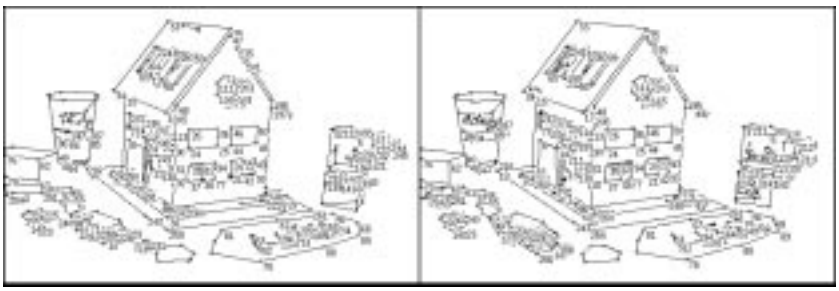

Fig. 8 - *Improvement of the matches shown on* Fig. *4*

3.4 Other experimental results

All the tools that were previously described may be mixed and used with different aims. In the example presented here, two images are taken from slightly different viewpoints, but two objects moved between the two shots.

These images are matched and this match is improved using the tools presented in this section. The result obtained after this first stage is shown on FIG. 9.

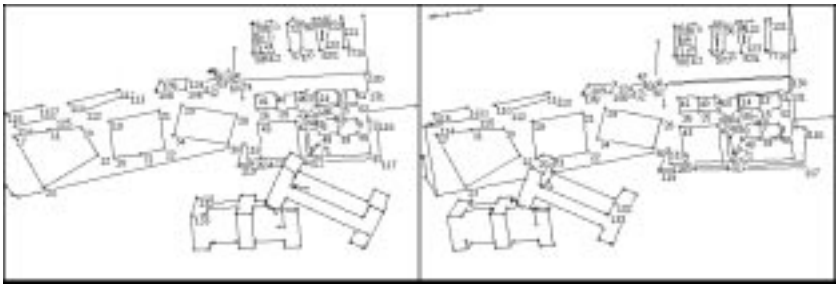

FIG. 9 - *Result of the first matching.*

In a second stage, we have kept in the images only the segments that were not matched and their extremities. These remaining segments are matched, and the match is improved. The result is shown on FIG. 10.

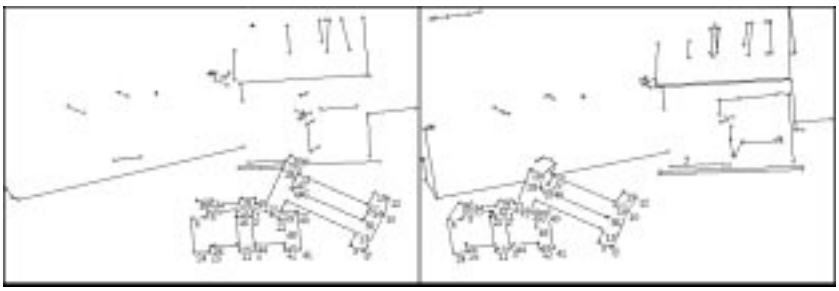

FIG. 10 - *Result of the second matching.*

This example shows the ability of our method to deal with mobile objects in a scene. In the first match, the objects in the background were matched. As a consequence of the use of a global constraint on the apparent motion, the mobile objects may not be matched. But when the matched features are removed, i.e. most of the background, the mobile objects can be matched. Such an ability seldom exists with other matching algorithms.

4 Matching more than two images

This last part of the paper is devoted to the case where more than two images must be matched together. To accomplish that, we propose an algorithm where the images are first matched two by two using what was previously described in the paper, and which deduces a global match of the images from these partial matches. This algorithm and a few experimental results are presented in the first paragraph of this section.

The second paragraph concerns the case where some images of an object are available, but do not represent the same aspect of it. We show how to compute a model for each aspect of the object. A few results are shown to validate this approach.

4.1 Matching n images

4.1.1 Algorithm principle

Let us assume that we want to match n images. These images are given in no particular order. In a first step, they are matched two by two, using what was described before. These matches may be summarized in a graph: the image features are the graph vertices, the feature matches are the graph edges. This graph is subsequently called the matches graph.

From this graph, we want to compute a hyper-graph which has the same vertices and whose hyper-edges represent global matches. Another way to state this problem is to search for a partitionment of the matches graph, such that each of the obtained components is a global match.

A first simple method: connectedness

A first simple idea consists in finding the connected components of the matches graph. This method provides bad results: each error makes a link between two components that should have remained separated and thus has a catastrophic effect. As soon as more than ten images are concerned, the biggest component may contain ten to twenty features of each image.

The Fig. 11 shows a part of the matches graph of a set of images. The features of all images were ordered such that all the features having the same number should go in the same global match. Right matches are shown with continuous lines and wrong matches with broken lines. In the Fig., three errors are enough to merge the four

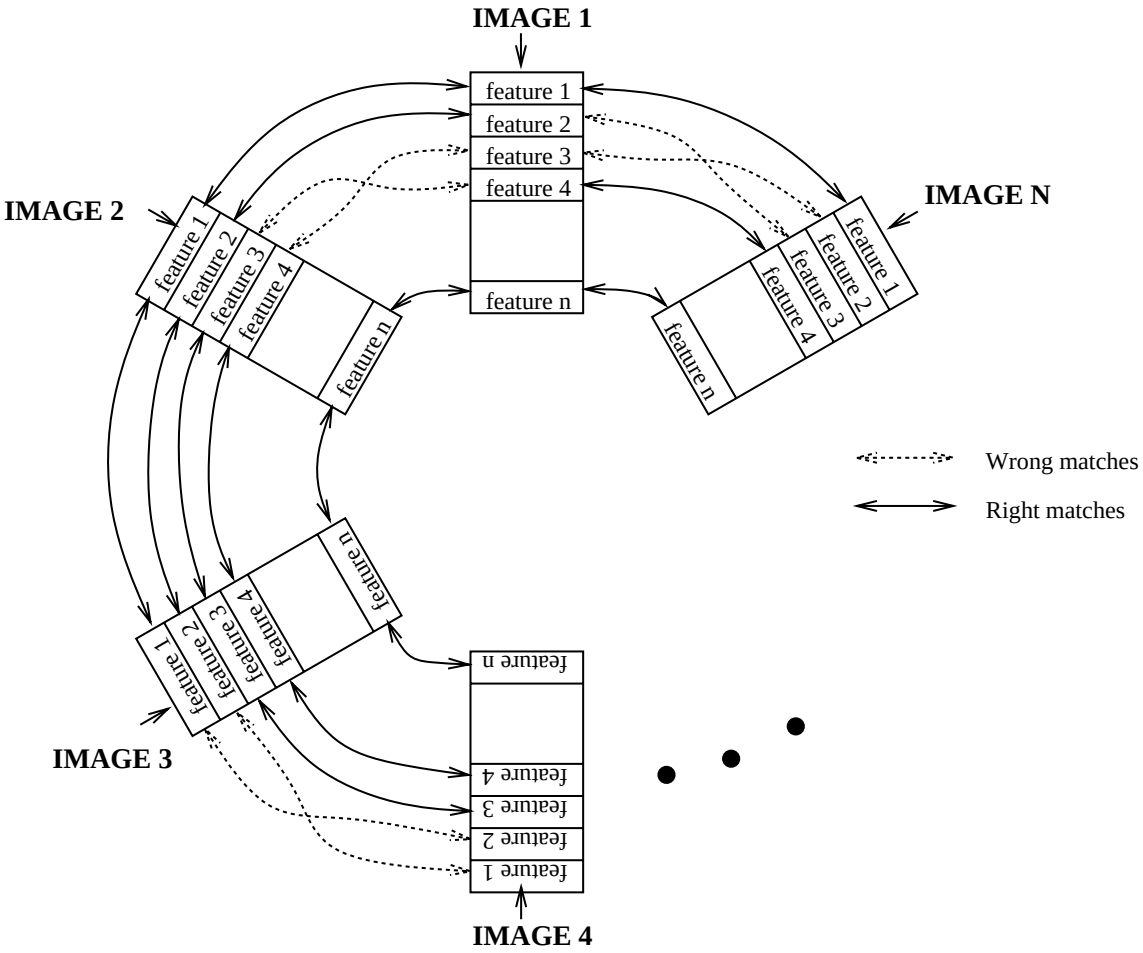

FIG. 11 - *When considering connected components, every error has a catastrophic effect.*

first global matches, and it is clear that every error can merge two such components and thus has a catastrophic effect.

The drawbacks of this method are twofold:

– the existence of a path between two features is a too weak relation, which does not take the available information redundancy into account. Let us consider two cases: in the first one, there exist only one path between two features f_1

and f_2, and the length of this path is 10; in the second case, there exist 10 paths of length 2 between these two features. It is clear that the probability that f_1 and f_2 are in the same global match is much bigger in the second case than it is in the first one. We would like the algorithm to take such an information into account.

- the constraints of the problem are neglected: it is unusual that two features of the same image must be in the same global match (it happens when a corner gives raise to two points rather than one). The case of 3 features of the same image may be a priori excluded. Ideal components should contain, in the best case, one feature of every image.

- as it was stated before, every error has a catastrophic effect.

Second method: strong connectedness

We want to eliminate these drawbacks and especially the last one, such that one error is not sufficient to merge to components. We define the notion of *strong connectedness*. Two features are said to be *k-strongly connected*, if there exists at least k paths of length smaller or equal than 2 between these two features in the matches graph.

It is then possible to search for the equivalence classes of the transitive closure of this relation for a given k. This method provides much better results than the first one, but raises the problem of choosing the threshold k.

Third method: strong connectedness and threshold adaptation

To correct this last problem, we propose a method to automatically adapt the threshold k. Let $C_k = \{c_k^i\}$ be the set of all components obtained with the relation of strong connectedness for a given threshold k. It is clear that the components of C_{k+1} form a partition of those of C_k. Each component c_{k+1}^i is included in a particular component c_k^j. All these components may then be organized in a tree: the root is the set of all image features; at the first level, the tree nodes are the components c_1^i; at level two are the components c_2^i... The edges between the tree nodes represent the relation of inclusion between the components of two consecutive levels.

The FIG. 12 presents such a tree. On the first line is c_0^1 which regroups all the components. The components obtained which k equal to 1, 2 and 3 are respectively on the second, third and fourth lines.

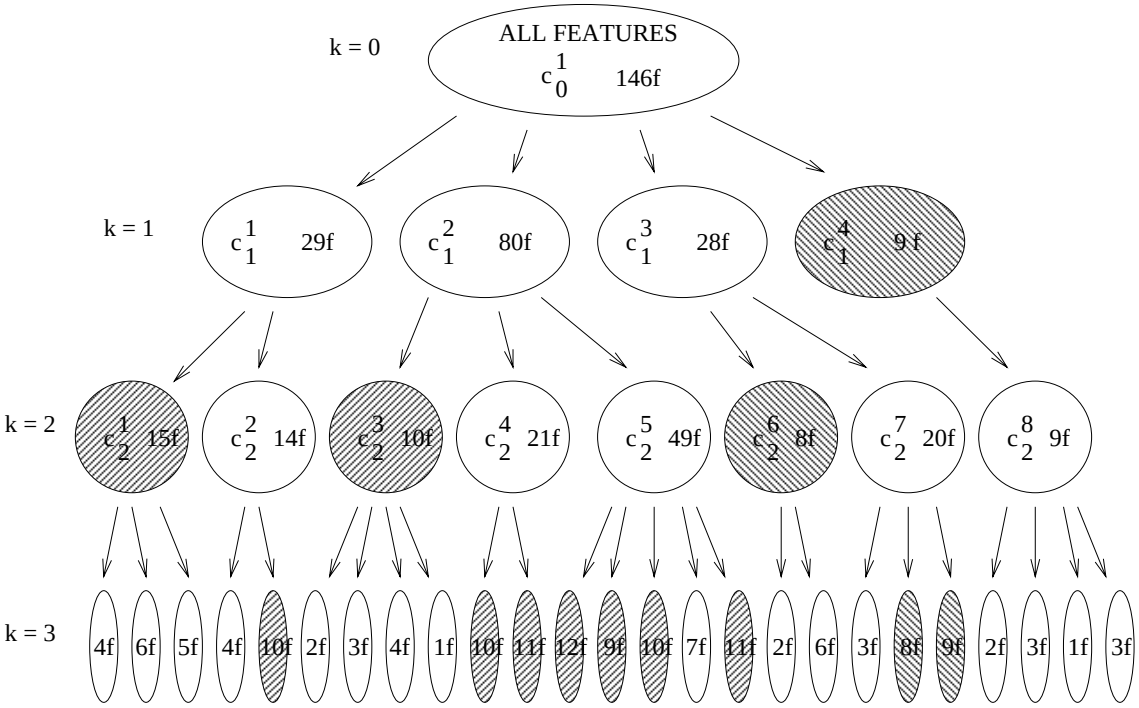

FIG. 12 - *An illustration of threshold adaptation. The component name is written in the first components, and the number of contained features in written in every component. The final components are hatched.*

We use the following algorithm, where n is the number of images and s_{top} et s_{down} are two thresholds, taken respectively equal to 1.2 and 0.8 in practice, for all examples.

1. The tree is examined from its root.

2. If an examined component c_k^i contains more than $n.s_{top}$ features, its sons are respectively examined. On FIG. 12, this is the case for c_1^1, c_1^2 and c_1^3 on the first line.

   If none of them may form a final component, c_k^i is a final component. This case is illustrated by c_2^1 on FIG. 12.

3. If c_k^i contains less than $n.s_{down}$ features, it is not a final component. This is the case for c_3^1, c_3^2 or c_3^3 on FIG. 12.

4. If c_k^i contains between $n.s_{down}$ and $n.s_{top}$ features, it is a final component. This case is illustrated by c_1^4 or c_2^3 on FIG. 12.

Two post-treatments are added to this search method.

Dilatation: let a final component have no final component as brother in the tree: if it does not contain a feature from an image, and if its father contains a feature from this missing image, this feature is added to the final component.

Reduction: let a final component contain two or more features of the same image: only the feature the most connected with the rest of the component remains in this component. Such a reduction will be applied to c_2^1 on FIG. 12.

4.1.2 Experimental results

We applied the previous algorithm, with strong connectedness, automatic threshold adaptation and the two post-treatments, to the graphs formed by point matches only. Segments matches were deduced from point matches afterwards.

Ten images of the object shown on FIG. 3 and 7 were matched. FIG. 13 shows the result of the global match on the images number 1, 4, 7, and 10 of the sequence. Points matched together have the same number.

During this computation, 68 global matches were found: 50 among them contain a feature of every image, 5 contain a feature of only 9 images, and 4 a feature from 8 images.

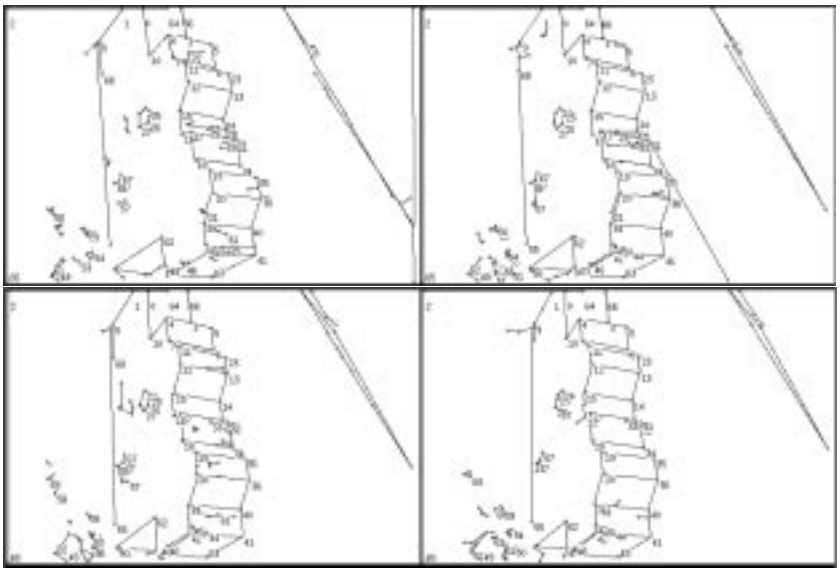

FIG. 13 - *Global match of a set of ten images (the images presented here have the number 1, 4, 7, and 10 in the sequence).*

4.2 Aspect modeling

This last paragraph illustrates the use of the different previous algorithms to model the different aspects of an object.

4.2.1 Modeling principle

From several images of an object, we want to establish automatically a model of all the aspects of the object which are represented in the image set. With this aim in view, we use the following algorithm.

1. The images are matched two by two.
2. A dissimilarity measurement is evaluated for every couple of images. The result is put in a dissimilarity matrix.
3. A hierarchical clustering algorithm is used to gather in components the images which represent a same aspect (or very close aspects) of the object.

4. In each of these components, a global match is realized using the partial matches computed at stage 1.

5. A model of each represented aspect is computed from these global matches. This ends the process.

Stages 1 and 4 have been already described. The three other stages are described in the latter.

Image dissimilarity

The aim of this measurement is to evaluate the similarity or dissimilarity of the contents of two images. This may be done by comparing what was matched between the two images with respect with their contents. This gives the following formula for two images I_1 and I_2:

$$d(I_1, I_2) = \frac{2}{3}\frac{\text{nbver}(I_1) \times \text{nbver}(I_2)}{\text{nbver}(I_1, I_2)^2} + \frac{1}{3}\frac{\text{nbedg}(I_1) \times \text{nbedg}(I_2)}{\text{nbedg}(I_1, I_2)^2}$$

In this formula, $\text{nbver}(I)$ and $\text{nbedg}(I)$ are respectively the number of vertices and the number of edges of image I; $\text{nbver}(I_1, I_2)$ and $\text{nbedg}(I_1, I_2)$ are respectively the number of vertex matches and of edge matches between the two images.

The value of the dissimilarity is thus 1 when the images are totally matched, and is equal to ∞ when the matching algorithm has totally failed. It is not a distance from a strict mathematical point of view, but this measurement is sufficient for a clustering algorithm.

FIG. 14 - *Six views of an object.*

For example, let us consider the six images shown FIG. 14. They were matched using the Euclidean invariants, and their distance matrix is:

$$\begin{pmatrix} 1 & 1.82859 & 11.8551 & 13.6304 & 27.2987 & 61.0833 \\ 1.82859 & 1 & 2.35642 & 24.6308 & 23.7482 & 15.5712 \\ 11.8551 & 2.35642 & 1 & 2.42964 & 18.4236 & 38.6125 \\ 13.6304 & 24.6308 & 2.42964 & 1 & 2.78519 & 41.875 \\ 27.2987 & 23.7482 & 18.4236 & 2.78519 & 1 & 2.67569 \\ 61.0833 & 15.5712 & 38.6125 & 41.875 & 2.67569 & 1 \end{pmatrix}$$

Hierarchical clustering

Numerous clustering methods are proposed in the literature [8, 32]. They usually require that either a threshold or the final number of clusters is given as an input to the algorithm.

As we do not have any idea of the number of clusters that should be obtained in our case, we preferred to use a method based on a threshold, which form a partition of the initial image set. It has the following stages.

1. Each image is put in a different set.
2. The two nearest sets (the less dissimilar ones) are merged if their dissimilarity is smaller than a given threshold.
3. The dissimilarity matrix is updated. The dissimilarity between two image sets is, by definition, the mean dissimilarity between the images of the first set and the images of the second one.
4. The stages 2 and 3 are repeated until no more sets can be merged.

Updating the matrix is very easy. As a matter of fact, if I, J and K are three image sets, and if $|L|$ denotes the number of elements of a set L, one may remark that:

$$\begin{aligned} d(I \cup J, K) &= \frac{1}{|I \cup J|.|K|} \sum_{i \in I \cup J, k \in K} d(i,k) \\ &= \frac{|I|}{|I \cup J|} d(I,K) + \frac{|J|}{|I \cup J|} d(J,K) \end{aligned}$$

Choosing the threshold is a very delicate thing. In our experiences, we had to adapt this threshold according to the image quality. Very noisy images need greater

threshold, between 10 and 20, and exact data, coming from CAD models for example, need a smaller one, around 2 or 3.

Modeling

At this last stage of the process, images are organized in components partitioning the initial image set. In each component, images are globally matched. The aim of this last stage is to establish a model of each component images.

In each of these components, an image is chosen. All the other images are transformed by an homography to be as close as possible to the chosen image (this homography was already computed during the matching improvement stage).

The model is a new image which contains the vertices and edges present in at least 60% of the component images. Each time a feature (and the features matched with it) is chosen to be part of the model, its position is computed as the mean position of the corresponding features in the transformed images.

The rate of 60% is smaller than the s_{down} threshold previously used, because the reduction post-treatment may reduce some global matches.

4.2.2 Experimental results

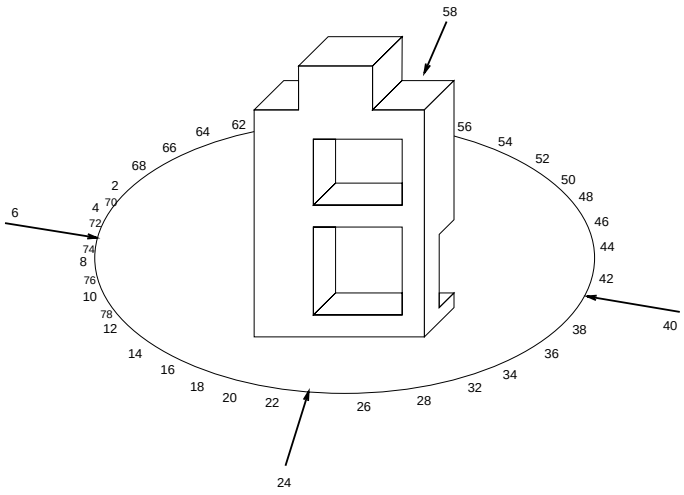

FIG. 15 - *78 images of an object.*

To test the algorithms just presented, we used the object shown on FIG. 15. Seventy eight images of this object were taken turning the camera around it, as shown on the FIG. With a threshold equal to 10 in the clustering algorithm, we obtained 10 image sets S_i: FIG. 16 shows the features extracted from four images of each set, and the model computed for this set.

Here are the images contained in each set:

$$\begin{aligned}
S_1 &= \{1,2,3,33,34,35,36,37,67,68,69,70\} \\
S_2 &= \{4,5,6,7,38,39,40,41,71,72,73,74\} \\
S_3 &= \{8,9,42,43,44,45,75,76\} \\
S_4 &= \{10,11,12,77,78\} \\
S_5 &= \{13,14\} \\
S_6 &= \{46,47\} \\
S_7 &= \{15,16,17,18,48,49,50,51,52\} \\
S_8 &= \{19,20,21,22,23,24,53,54,55,56\} \\
S_9 &= \{25,26,27,57,58,59,60,61,63\} \\
S_{10} &= \{28,29,30,31,32,62,64,65,66\}
\end{aligned}$$

It should be noticed that the object is almost symmetric with respect to a vertical plane and each set contains some images taken from both sides of the object.

To illustrate the modeling algorithm, we also used the images shown in the previous sections, which have more numerous features. The model shown on the left of FIG. 17 was computed from the ten images of the object shown on FIG. 3. This model contains 70 points.

The model shown on the right of FIG. 17 was computed from five images of the house shown on FIG. 4. This model contains 181 points.

5 Conclusion

We present in this paper several algorithms for image matching and object modeling, in the case where the images are formed of line segments and vertices or points between these segments: matching of two or several images, computation and modeling of the different aspects of an object. All these algorithms were illustrated by results obtained with real images.

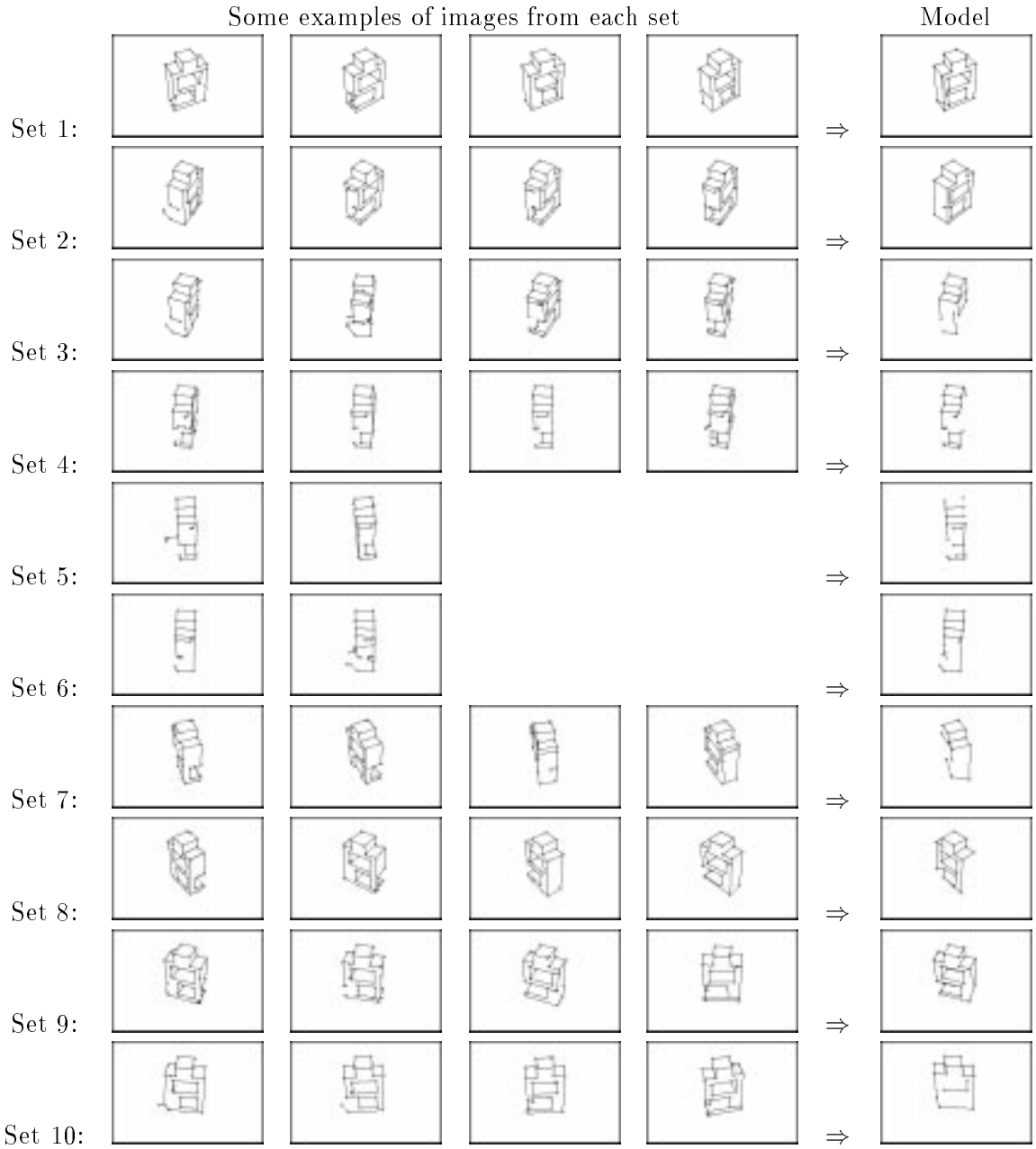

FIG. 16 - *Four images and the model of each set.*

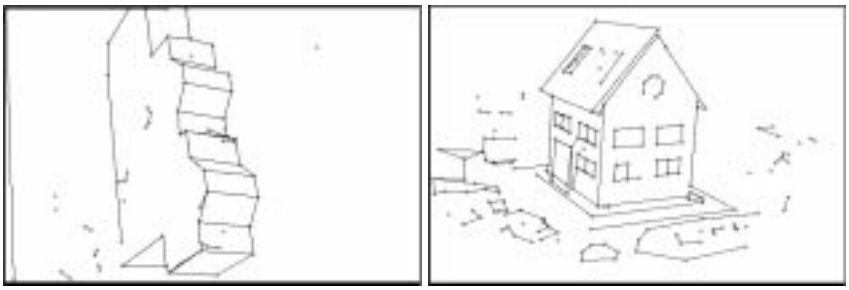

FIG. 17 - *The two obtained models.*

These algorithms are based on the use of local quasi-invariants and of a global constraint (all the matches have to define an approximation of the same apparent motion).

These tools allow:

- to use uncalibrated cameras or stereo rigs,
- to manage the case where there exists a mobile object in the scene,
- to work with noisy images and occluded objects,
- to keep a low complexity.

In a few words, it allows to deal with images where no other method is available, and the results clearly show the robustness of the methods.

This work may be extended in three directions.

1. The basic algorithm, that of two image matching, is based upon the existence of local quasi invariants. It ought to work with other features, such as curves for example.
2. The models we obtain summarize the stable and characteristic information about an object. Then it may be used to recognize this object in other images. The problem is to be able to index such models in a data base, and to have an efficient search algorithm.
3. The modeling process is quite close to the "learning from example" paradigm. The question is then to really show that our modeling technique is a learning process, i.e. that the models that are build allow a better and faster recognition than the initial images.

Références

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[2] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2):107–132, April 1987.

[3] H.H. Baker and T.O. Binford. Depth from edge- and intensity- based stereo. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 631–636, August 1981.

[4] T.O. Binford and T.S. Levitt. Quasi-invariants: theory ans exploitation. In *Proceedings of* DARPA *Image Understanding Workshop*, pages 819–829, 1993.

[5] B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, June 1995. to appear.

[6] J. Canny. A computational approach to edge detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[7] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2):167–187, 1987.

[8] E. Diday and J.C. Simon. Clustering Analysis. In K.S. Fu, editor, *Communication and Cybernetics*. Springer-Verlag, 1976.

[9] O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint*. Artificial intelligence. M.I.T. Press, Cambridge, MA, 1993.

[10] W. Forstner and A. Pertl. Photogrammetric standard methods and digital image matching techniques for high precision surface measurements. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice II*, pages 57–72. Elsevier Science Publishers, 1986.

[11] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 1990.

[12] D.B. Gennery. *Modelling the Environment of an Exploring Vehicle by means of Stereo Vision*. Ph.d. thesis, Stanford University, June 1980.

[13] W.E.L. Grimson. A computer implementation of the theory of human stereo vision. *Philosophical Tansactions of the Royal Society of London*, B292(1058):217–253, 1981.

[14] W.E.L. Grimson. Computational experiments with a feature based stereo algorithm. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34, 1985.

[15] A.W. Gruen. Adaptative least squares correlation: a powerful image matching technique. *S. Afr. Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187, 1985.

[16] R. Hartley. In defense of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, June 1995. to appear.

[17] L. Hérault. *Réseaux de neurones récursifs pour l'optimisation combinatoire.* Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 1991.

[18] M. Kass. A computational framework for the visual correspondence problem. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, Germany*, pages 1043–1045, August 1983.

[19] M. Kass. Linear image features in stereopsis. *International Journal of Computer Vision*, 1(4):357–368, January 1988.

[20] R.E. Kelly, P.R.H. McConnel, and S.J. Mildenberger. The gestalt photomapper. *Photogrammetric Engineering and Remote Sensing*, 43:1407–1417, 1977.

[21] F. Klein. *Le Programme d'Erlangen.* Collection "Discours de la méthode". Gauthier-Villars, Paris, 1974.

[22] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[23] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B 204:301–328, 1979.

[24] G. Médioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics and Image Processing*, 31:2–18, 1985.

[25] L. Morin, P. Brand, and R. Mohr. Indexing with projective invariants. In *Proceedings of the Syntactical and Structural Pattern Recognition workshop, Nahariya, Israel*. World Scientific Pub., 1994. to appear.

[26] H.K. Nishihara. PRISM, a pratical real-time imaging stereo matcher. Technical Report Technical Report A.I. Memo 780, Massachusetts Institute of Technology, 1984.

[27] Y. Ohta and T. Kanade. Stero by intra and inter-scanline search using dynamic programming. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[28] S. B. Pollard. *Identifying Correspondences in Binocular Stereo.* PhD thesis, University of Sheffield, 1985.

[29] S.B. Pollard, J.E.W. Mayhew, and J.P. Frisby. PMF: A stereo correspondence algorithm using a disprity gradient constraint. *Perception*, 14:449–470, 1985.

[30] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling W.T. *Numerical Recipes in C.* Cambridge University Press, 1988.

[31] C.A. Rothwell. Hierarchical object descriptions using invariants. In *Proceeding of the* DARPA–ESPRIT *workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pages 287–303, October 1993.

[32] G.L. Scott and H.C. Longuet-Higgins. Feature grouping by "relocalisation" of eigenvectors of the proximity matrix. In *Proceedings of the British Machine Vision Conference, Oxford, England*, pages 103–108, September 1990.

[33] T. Skordas. *Mise en correspondance et reconstruction stéréo utilisant une description structurelle des images.* Thèse de doctorat, Institut National Polytechnique de Grenoble, France, 1988.

[34] Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. Rapport de recherche 2273, INRIA, May 1994.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399