HAL
open science

Matching and Clustering: Two Steps Towards Automatic Model Generation in Computer Vision

Patrick Gros

To cite this version:

Patrick Gros. Matching and Clustering: Two Steps Towards Automatic Model Generation in Computer Vision. AAAI Fall Symposium Series: Machine Learning in Computer Vision: What, Why, and How?, Oct 1993, Raleigh, United States. pp.40–44. inria-00590020

HAL Id: inria-00590020

https://inria.hal.science/inria-00590020v1

Submitted on 3 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Matching and Clustering: two Steps towards Automatic Object Modelling in Computer Vision

Patrick Gros

Lifia – Imag – Inria Rhône-Alpes

46, avenue Félix Viallet - 38031 Grenoble Cedex 1 - France

Abstract

In this paper, we present a general frame for a system of automatic modeling and recognition of 3D polyhedral objects. Such a system has many applications for robotics: recognition, localization, grasping... Here we focus upon one main aspect of the system: when many images of one 3D object are taken from different unknown viewpoints, how to recognize those of them which represent the same aspect of the object? Briefly, is it possible to determine automatically if two images are similar or not? The two stages detailed in the paper are the matching of two images and the clustering of a set of images. Matching consists in finding the common features of two images while no

information is known about the image contents, the motion or the calibration of the camera. Clustering consists in regrouping into sets the images representing a same aspect of the modeled objects. For both stages, experimental results on real images are shown.

1 Introduction

This paper is concerned with the problem of automatic recognition of 3D polyhedral objects. Such a 3D object recognition system has two major parts: *object modelling* and *recognition*, i.e. matching of a new sensed image with an already constructed model. This model is usually stored in a model data basis.

Here we address the first part of the problem, object modelling. A camera takes many images of one object under different viewpoints; from these images we construct the views of this object, a view being a set of images representing the same aspect of the object. All the views form the object model. The aim of such a system is to reduce the information existing in the images, i.e. the size of the representation of the object. Such a reduction will allow a smaller size of the model database and then a greater speed for the recognition system. Typically an object is modeled from one hundred images and we construct about ten different views.

The applications of such a system in a robotic environment are numerous: recognizing objects allows a robot arm to grasp them, a mobile robot to avoid them when moving or to recognize its position according to high level markers. Furthermore, recognition is a bridge between low level environment description in terms of free space and shapes, and a high level description in terms of objects, rooms and ways. It thus should allow robot tasks to be described symbollically and it realizes a strong link between sensing and planning.

The current approaches to the modelling problem may be classified according to two criterions: the kind of data used to construct the model and the kind of model constructed. The data may be 2D or 3D, man made or obtained from a sensor. The model may be 2D or 3D. Such a classification is presented by Flynn et al. [FJ91] and is used here to compare the different systems.

3D man made data: they usually come from a CAD system. The data are made of a description of the object in terms of its geometrical and mechanical properties. The problem is thus to infer the object visual aspects from these data. The model building step using CAD data has been intensively studied, creating a new field of vision called CAD-based vision [Bha87].

2D man made data: another way of using CAD data is to compute the 2D

aspects of the modelled object [KD79, PPK92]. Each aspect is topologically different from the others and they are ordered in a graph called aspect graph according to their associated viewpoint. The model of the object thus consists of the set of all its aspects. Even simple objects may have several tens of different aspects.

3D sensed data: they concern mostly two fields of vision: medical imagery using 3D volumetric sensors and robotic applications using 3D range sensors. In the first case the sensor gives a complete 3D image, while it gives only a depth map from a given viewpoint in the second case. Surveys of these techniques are given by Besl [Bes88] and Nitzan [Nit88].

2D sensed data: these data are usually images of the object to be modeled, taken from different viewpoints. Modelling and recognition systems using such data are very numerous. They differ in the kind of information they extract from the images, and in the dimension of the model (2D or 3D). Connell and Brady [CB87] use intensity data, Arbogast [AM91] use occlusion contours, Mohr et al. [MVQ93] use points, Rothwell et al.[RZFM92] use numerical invariants associated with some configurations of points, lines and curves, Weiss uses differential invariants associated with algebraic curves [Wei92].

Our approach falls into this last category. The input consists of a large set

of images. These images represent the object to be modeled and are taken from different viewpoints. The aim of the method is to find out which of these images represent the same aspect of the object. Such images belong to the same view of the object, and all these "characteristic" views form the object model.

Our method relies upon the matching of images one with another: two images represent the same object aspect if they contain approximatively the same features and the same relationship between them. Thus we try to compare the contents of the different images. As the viewpoint changes between the different images, the location of the features within the images also changes and we try to estimate this motion in order to find a correspondence between the features of each image.

Our method models an object directly from what can be seen of this object in images. In this it differs from the methods based on CAD data. With these methods, the main problem is to infer visual information from geometrical properties. This inference is usually not satisfactory and is a weakness of the method. Furthermore, the use of aspect graphs adds another problem: the number of theoretical aspects of an object is much greater than the number of its visual aspects. Theoretical aspects very often differ only in unsignificant details. The complexity of these methods is a real obstacle. Bowyer gives a compete criticism of these methods [Bow91]. On

the contrary, our method has a pragmatic notion of aspect. The different aspects are separated according to their visual dissimilarities, and not to their topological differences.

With respect to the methods using 3D models computed from 2D sensed data, our method avoids the reconstruction and projection stages. The reconstruction consists of computing the 3D shape of an object from 2D information. The projection is the opposite operation, i.e. computing a 2D visual aspect of an object from its 3D model. These two stages are complex and sensitive to noise.

Our method is thus more natural: the data used for modelling are 2D sensed data, so are the images to be recognized. The built models stay as close as possible to this kind of data.

In this paper, we focus on two stages of the method. The matching of two images when no a priori information is known is studied in section 2 and section 3 concerns the clustering of similar images. Both sections show experimental results. Two directions of further work are discussed in the conclusion.

2 Matching sets of 2D features

2.1 *The matching algorithm: general description*

At this stage, our inputs are two images containing contours approximated by line segments. The aim of the matching is to find which segments of each image are the projections of the same edge of the 3D object. The output is a correspondence between the features (here the segments) of each image.

Matching is a prior stage to many algorithms and usually relies on one of the two following assumptions:

1. first assumption: the motion of the camera between the two viewpoints or that of the object if the camera is supposed motionless, is approximatively known and the location of one feature in an image may be deduced from the location of the corresponding feature in the second image. This assumption is done for example by the systems based on correlation techniques [Ana89, Fua90]. Another important case of systems using this assumption is that of tracking. The motion is supposed to be very small or very regular and the location of the features within an image of a sequence may be predicted from the knowledge of the previous images of the sequence [CS90, DF90].

2. second assumption: some of the features or group of features remain qualitatively similar. In this case, matching is based on the search of particular features configurations: small graphs of segments [SH92], the whole graph of all the segments [HHVN90], symmetric features [HSV90].

The first methods are quite limitated by their assumption: the motion has to be approximatively known. In many cases, especially those when the camera is not calibrated, the motion is not known at all, even if its kind (pure rotation or translation...) is known. This is also the case if the images are taken with different cameras. The second methods are sensitive to noise. In the case of the use of small graphs of segments, either these graphs are too big and their configuration is never perfectly conserved, or they are too small and are no longer discriminant.

In our method we also use small groups of features. We do not characterize them by topological properties but by geometrical ones. We do not consider the exact motion of the camera, but only the apparent change of location of the features within the images. If we superimpose these images, we can speak of apparent motion of the features. The method is based on the knowledge of the kind of this apparent motion and on the estimation of its parameters. The second principle of the method is that it is not worth spending computing time to match a small number

of features between two images which belong to two different views of the object. As we want to cluster similar images, it is sufficient to know that the matching is almost impossible, i.e. that the images represent different aspects.

The different stages of our matching method are the following.

1. We have two images containing line segments approximating contour curves. We assume that the apparent motion of the segments between the two images is a similarity (see next paragraph). We associate numerical invariants to the features. They are the angle and the length ratio defined by every pair of segments having an extremity in common.

2. The invariants and their corresponding segments are matched according to the value of the invariants: two pairs of segments of two images are matched if they define equal angles and length ratios. As there is some noise in the images, the equality is tested up to a noise threshold, in consequence of what all matches are not right.

3. To eliminate the wrong matches, a Hough transform technique is used, in order to evaluate the parameters of the apparent motion. As a matter of fact, the right matches correspond to the same apparent motion and the computation of this motion allows to recognize them. When two invariants are matched, there

is enough geometrical information to compute the transformation [GQ92]. In our case, when two pairs of segments are matched, it is possible to compute the parameters of the similarity which transform one of the two pairs into the second one. Such a computation is done for all the matches done at stage 2, whether they are right or wrong. In this way, each match gives a point in the transformation parameter space.

4. The points corresponding to wrong matches are distributed almost uniformly in the parameter space. This is because they are not correlated. On the contrary, the points corresponding to right matches define all the same real transformation parameters up to a noise factor. Thus they give many points in a small region of the space. This "accumulation point" may be found easily: all the points are projected on each of the space coordinate axis. A convolution computation allows to find the interval of each axis which contains the maximum number of projected points. These intervals are the projections of the accumulation point. All the matches which give a transformation, whose parameters are not in these interval, are eliminated.

5. The match between the individual segments are deduced easily from the matches of segment pairs.

This method allows matching images with no a priori information about the images, and it is more powerful than tracking or correlation methods. Furthermore, it uses only very small groups of features and is thus more robust to noise than the methods based on a topological description of the images. Its only limitation comes from its incapacity to match images representing different aspects of the observed object, but this is not a problem for our modeling method. On the other hand, it is based on local features and it is not sensitive to partial occlusion or to the eventual existence of a background visible in the images.

The next paragraph justifies the use of similarities. After that, some experimental results are shown on real images and some techniques of image correction to ameliorate the matching are presented.

2.2 *Comments on the choice of similarities*

The apparent motion is not any classical planar transformation (Euclidean, affine, projective). But in many cases, it can be approximated precisely by one of these transformations. When the object is flat, the transformation is projective. Thompson et al. [TM87] show that the perspective distorsions are negligible if the depth of the 3D object is at least ten times smaller than its distance to the camera. The

apparent motion is a similarity if two conditions are satisfied: firstly the object is planar and orthogonal to the principal axis; secondly, the principal axis of the camera dœs not move between the two shots.

In practice these assumptions are not strictly satisfied. The invariants we use are the angle and length ratios of each pair of segments having an extremity in common. The use of other segments could be considered, but it would increase the combinatorics of the computation, Furthermore, when the assumptions are not strictly observed in general, they can be so locally. Our experimentations show that the observance of the assumptions is not too strict. For example, the invariance of the principal axis may be practically understood as "this axis must not rotate of more than 15 degrees".

2.2.1 Mathematical considerations

Let us give a more mathematical argumentation about the choice of similarities. We consider a classical perspective projection model for the camera. The following frames are orthonormal (see Figure 1): an object centered frame (O, X, Y, Z), an image frame (o, u, v), the image orientation being provided by the direction of the optical axis, a camera frame (C, x, y, z) where C is the optical center, Cz is the optical axis, Cx being parallel to ou.

Ten parameters are needed in order to determine the projection of an object onto the image: six parameters for defining motion between the object and the camera, i.e. three angles (α, β, γ) of rotation around each axis and three scalars (a, b, c) for a translation, two parameters give the scale factor between the camera frame and the image frame (if we consider square pixels these two parameters reduce to one, k), the last two parameters define the translation between the image frame origin and the intersection between the image and the optical axis, (d, e).

Transformation equations. An object point M projects onto an image point m and this transformation can be written as a matrix with homogeneous coordinates:

$$\begin{pmatrix} u_m \\ v_m \\ w_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix} \begin{pmatrix} x_M \\ y_M \\ z_M \\ t_M \end{pmatrix}$$

The coefficients of this matrix can be expressed in terms of the parameters just described:

$$a_{11} = k\cos\gamma\cos\beta + d\sin\beta, \quad a_{21} = k\sin\gamma\cos\beta + e\sin\beta, \quad a_{31} = \sin\beta$$

$$a_{12} = -k\cos\gamma\sin\beta\sin\alpha - k\sin\gamma\cos\alpha + d\cos\beta\sin\alpha$$

$$a_{22} = -k\sin\gamma\sin\beta\sin\alpha - k\cos\gamma\cos\alpha + e\cos\beta\sin\alpha$$

$$a_{32} = \sin\alpha\cos\beta$$

$$a_{13} = -k\cos\gamma\sin\beta\cos\alpha - k\sin\gamma\sin\alpha + d\cos\beta\cos\alpha$$

$$a_{23} = -k\sin\gamma\sin\beta\cos\alpha - k\cos\gamma\sin\alpha + e\cos\beta\cos\alpha$$

$$a_{33} = \cos\alpha\cos\beta$$

$$a_{14} = ka + dc, \quad a_{24} = kb + ec, \quad a_{34} = c$$

Restrictive assumptions. We introduce now some restrictions onto the projection parameters in order to simplify these equations. The effect of these restrictions is to insure that the various images of the same object depict the same characteristic view and to allow the estimation of the transformation parameters without any point-to-point correspondence.

First we assume that the perspective effects are weak. One can consider this hypothesis as valid whenever the ratio between the approximate size of the object and the distance from the object to the viewer is 0.1 or less[TM87]. Mathematically this translates into $t \ll z$ or $a_{i4} \approx 0, i = 1, 2, 3$.

Second we assume that the relative displacement (between two views of the same object) is such that the same characteristic view is seen in both images. Mathematically, this is equivalent to restrict "lateral" rotations and translations:

$\alpha \approx 0, \quad \beta \approx 0, \quad a \ll c, \quad b \ll c$

The projection transformation becomes:

$$\begin{pmatrix} u_m \\ v_m \\ w_m \end{pmatrix} = \begin{pmatrix} k\cos\gamma & -k\sin\gamma & d \\ k\sin\gamma & k\cos\gamma & e \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_M \\ y_M \\ z_M \\ t_M \end{pmatrix}$$

The object to viewer transform is the composition of perspective projection and a 2-D direct similarity. This last transform is composed of a scaling, a rotation, and a translation. If we denote by S the similarity matrix and by P the projection matrix we have: $m = SP(M)$.

For two different views we have two different similarities but the same projection: $m_1 = S_1P(M)$ and $m_2 = S_2P(M)$. The similarity being invertible we immediately obtain the mapping of points from one view onto points from the other view: $m_2 = S_2S_1^{-1}(m_1)$. This mapping has four parameters associated with it: a scaling factor k, an angle of rotation γ, and two scalars d and e defining a translation.

2.2.2 *The case of other image transformations*

The distorsion of the projection of a 3D object in different views cannot actually be modelled by an image transformation. However, as affine or projective image

transformation offers more parameters than similarities (6 in the affine case, 8 in the projective one), they provide a way to get a better approximation of the observed transformation.

Nevertheless, such complex transformations have more complex invariants: length ratios of collinear points and affine coordinates for the affine transformations, and cross-ratios and projective coordinates for the projective ones. Such invariants may be computed choosing 3 or 4 points as a reference frame and another point whose coordinates are computed in that frame. The first problem for an effective computation is the choice of these points. Even if we restrict the possible configurations to the points lying on some particular subgraphs, the combinatorics remains high. Secondly, these invariants are not always very robust to noise [Mor93].

2.3 *Experimental results*

In this paragraph, we provide some results which show that the algorithm runs well even if the assumptions are not strictly respected.

Figure 2 shows an example using the algorithm. The original images are shown on the left, the features extracted from these images are shown in the middle and the features which are matched are on the right. The segments which are not matched

are usually broken in several smaller segments in one of the images and not in the other one.

This example shows clearly that the assumptions are not too strict. Between the two shots, the principal axis of the camera has rotated of more than 15 degrees and the algorithm still runs correctly.

The second example (Figure 3) shows what happens when the images are too far from the theoretical assumptions. Almost nothing is matched, though a few segment matches are correct. This demonstrates that wrong matches do not form any accumulation points in the transformation space and that the right matches will be found even with much noise in the images. That also shows that the main limit of the algorithm is the invariance of "the invariants" and not the principle of the algorithm itself.

2.4 *Image correction*

The algorithm just presented allows the features in two images which have similar geometrical properties to be matched. Some other features have not been matched because they are affected by noise. Comparing the unmatched features, it is then possible to find some of the effects of the noise and to correct them.

Some examples of these corrigible effects are T-junctions and split junctions. As a first match is already done, it may be carried on by topological considerations. For example, if two junctions are matched, the segments which go through this junction should probably be matched. If they are not, we look for T-junctions, split-junctions, collinearities...If such an error exists in one image and not in the other one, it is corrected and we may carry on the matching.

The justification of such corrections comes from statistical properties. In most cases, T-junctions, split junctions and collinearities which cannot be matched are due to noise, rather than by the object itself. Furthermore, it is possible to exhibit some causes for them [GM92]: the fact that the depth of focus is finite, the passage from 2D to 3D or electronic noise.

The corrections are shown on Figure 4. On the left of the figure are shown the noised structures, and the corrected ones are shown on the right.

Figure 5 shows an example of correction. The two upper images are two views of an object. They are very similar, but the noise is very different. The left lower image shows the elements which are matched when no correction is done. The last image shows the features matched when some corrections are done.

2.5 *Conclusion on matching*

The matching algorithm presented here is very simple. The approximations of the apparent motion are quite unrefined and the invariants used are only based on 2D transforms. This makes the algorithm robust and allows it to run with no prior information. Of course, the quality of the matches can be improved easily using topological information for example.

3 Clustering of an image set

This stage of the modelling process consists of grouping into sets or clusters the images representing neighboring aspects of the object. This is done by computing a measure of likeness between images and by using a classical clustering method.

3.1 *Measure of the likeness between images*

With a set of images of a same object, the algorithm of the previous section allows to match all the pairs of images. When the images are similar, many features are matched; when they are not, the matching is very poor. It is then possible to measure the similarity of two images according to the proportion of features matched. The measure may be called a distance, but not in a mathematical sense.

The formula we use for this measurement is:

$$d(I1, I2) = \frac{a}{d}\frac{nbseg_1.nbseg_2}{nbseg^2_{matched}} + \frac{b}{d}\frac{nbvrt_1.nbvrt_2}{nbvrt^2_{matched}} + \frac{c}{d}\frac{sumdeg_1.sumdeg_2}{sumdeg^2_{matched}} \quad (1)$$

$nbseg_1$, $nbseg_2$ and $nbseg_{matched}$ are respectively the number of segments of the first image, that of the second image, and the number of segments matched between the two images. $nbvrt_1$, $nbvrt_2$, $nbvrt_{matched}$ have similar meanings about vertices. $sumdeg_1$ and $sumdeg_2$ are respectively the sums of the degrees of all the vertices of the first image and the same sum for the second image. $sumdeg_{matched}$ is the sum of the degrees of the matched vertices: when two vertices, one of each image, are matched, their degree is the number of pairs of matched segments going through these vertices. a, b and c are three coefficients which have to be experimentally determined. According to our experiments, we take $a = b = 2$ and $c = 1$. d is equal to $a + b + c$.

3.1.1 Experimental results

Figure 6 displays eight images of a same object. All pairs of images were matched and the likeness of these image pairs was computed. The distance matrix obtained

for these eight images is the following:

$$\begin{pmatrix} 1.00 & 2.26 & 5.85 & 75.65 & 80.20 & 25.59 & 103.68 & 34.39 \\ & 1.00 & 2.10 & 63.40 & 27.98 & 98.97 & 30.41 & 115.00 \\ & & 1.00 & 3.93 & 17.86 & 49.79 & 59.74 & 73.18 \\ & & & 1.00 & 1.92 & 2.29 & 11.36 & 5.96 \\ & & & & 1.00 & 2.02 & \infty & 21.22 \\ & & & & & 1.00 & 1.8 & 6.96 \\ & & & & & & 1.00 & 7.00 \\ & & & & & & & 1.00 \end{pmatrix}$$

The symbol ∞ means that the matching process failed to find common features between the two considered images. The exact value of big numbers has not much significance. It only shows that the images differ a lot. The pertinence of this measure is shown by the results of the clustering.

3.2 *Clustering of a set of images*

The method we use to regroup the images is a classical agglomerative method: each image is put in a different cluster; their distance is that of the images. The two nearest clusters are grouped if their distance is inferior to a threshold; the distances between the clusters are updated; the distance between two clusters is equal to the

mean of the distances of the images of the each cluster; The process is repeated until no new grouping is possible.

This method forms a partition of the initial set of images, what is not necessary, and there is a threshold to determine. The partition constraint gives in fact a way to compute a threshold automatically. If we consider the likeness of the clusters which are grouped at each step of the process, we obtain a sequence of positive numbers which has a gap (this is an experimental verification). Here are for example some sequences obtained with different sets of real images (boldface numbers locate the gap):

- 1.115 1.242 1.257 1.367 1.380 1.425 1.425 1.541 1.757 1.849 2.120 2.13 0 2.485 **3.939** **36.097** 471.307 905.698
- 3.537 3.690 4.309 4.435 4.704 5.026 **10.488** **47.163** 75.783

3.2.1 Experimental results

Let us consider the sequence of image of Figure 6. The clustering algorithm gives the following groups: $\{I1, I2, I3\}$, $\{I4, I5\}$ and $\{I6, I7, I8\}$

To test our algorithms on a more significant set of images, we took 80 images of a same object as shown on Figure 7. The first twenty images are taken every 2.5

degrees, the other ones every 5 degrees. The clustering process gives 7 groups as follows:

Group 1: images 1 2 3 4 33 34 35 36 37 38 39 68 69 70 72 73

Group 2: images 5 22 23 24 25 26 28 58 59 60 61 62

Group 3: images 6 7 8 9 10 40 41 42 43 44 74 75 76 77 78

Group 4: images 11 12 13 14 15 27 45 46 47 48 49

Group 5: images 16 17 50 71

Group 6: images 18 19 20 21 51 52 53 54 55 56 57

Group 7: images 29 30 31 32 63 64 65 66 67

Figure 8 shows two images of each group and the features extracted from these images. As the object is almost symmetric, each cluster collects images of both sides of it. As similarities are used to compute the matching, only images taken with neighboring viewpoints are gathered in one group. For example, the groups 2 and 6 contain some images topologically very similar, but these images have very different invariants for the similarities. To avoid this problem (if we want to avoid it), we should use other transformations like affine transformations or collineations.

The result is not totally perfect. Images number 5, 27 and 71 are not in the group expected, but this is due to the noise of the images. The effect of this clustering noise will be attenuated at the modelling stage with the introduction of reliability

coefficients for each feature of a group.

3.3 Conclusion on clustering

The clustering process is based on a very simple algorithm for the computation of the similarity measure and for the clustering itself. The experimental results are good, even with a big set of images. The main problem of this algorithm is its complexity. As it implies to match all the pairs of images, the complexity is $O(n^2)$ where n is the number of images. Fortunately, this stage is off-line!

4 Conclusion

In this paper, we have detailed two stages of a modeling method: how to match two images to find their common features, and how to use this matching to regroup the images of a set of images, which represent similar aspects of the 3D modeled object. These stages are based on the estimation of the apparent motion using invariants associated with small groups of features.

Its main advantages are its robustness to image noise and partial object occlusions and its generality: it dœs not need any a priori information about the object, the motion of the camera between the two shots or any calibration of the camera.

The main extension of this algorithm concerns the used features. The conditions necessary for these features are the possibility of a segmentation, a parametrization which allows the computation of invariants, and the possibility to organize these features into a structure. B-splines approximating the contours would be a good candidate. The main difficulty will be to compute reliable invariants for them: it is the subject of the CCE ESPRIT-BRA VIVA project.

It should be noticed that even with simple invariants and approximations, it is possible to deal with numerous images of polyhedral complex objects because of the robustness of the method. This will allow this method to be used for practical robotic applications. That's the aim of the CCE ESPRIT-BRA SECOND project.

Acknowledgements. Marie Legendre, Marie-Hélène Malissen and Gudrun Socher are gratefully acknowledged for their participation to this project, so are Radu Horaud, Françoise Veillon and Roger Mohr for their fruitful comments. This work has been sponsored by the ORASIS project as part of the PRC Communication Homme-Machine and by the CEC through the ESPRIT-BRA 3274 (the FIRST project). Patrick Gros acknowledges support from Région Rhône-Alpes.

References

[AM91] E. Arbogast and R. Mohr. 3D structures inference from images sequences. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(5):749, 1991.

[Ana89] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[Bes88] P.J. Besl. Active optical range imaging sensors. Springer-Verlag, New York, USA, 1988.

[Bha87] B. Bhanu. Guest editor's introduction. *Computer (Special Issue on CAD-Based Robot Vision)*, August 1987.

[Bow91] K. Bowyer. Why aspect graphs are not (yet) practical for computer vision. In *Proceedings of the* IEEE *workshop on Direction on automated* CAD*-based Vision, Maui, Hawaii, USA*, pages 97–104, 1991.

[CB87] J.H. Connell and M. Brady. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31:159–183, 1987.

[CS90] J.L. Crowley and P. Stelmazyk. Measurement and integration of 3D structures by tracking edges lines. In O. Faugeras, editor, *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, pages 269–280. Springer-Verlag, April 1990.

[DF90] R. Deriche and O. Faugeras. Tracking line segments. In *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, pages 259–267. Springer-Verlag, April 1990.

[FJ91] P.J. Flynn and A.K. Jain. CAD-based computer vision: from CAD models to relational graphs. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(2):114–132, February 1991.

[Fua90] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 1990.

[GM92] P. Gros and R. Mohr. Automatic object modelization in computer vision. In H. Bunke, editor, *Proceedings of the workshop "Advances in Structural and Syntactic Pattern Recognition", Bern, Switzerland*, volume 5 of *Series on Machine Perception and Artificial Intelligence*, pages 385–400. World Scientific, August 1992.

[GQ92] P. Gros and L. Quan. Projective Invariants for Vision. Technical Report RT 90 IMAG - 15 LIFIA, LIFIA–IRIMAG, Grenoble, France, December 1992.

[HHVN90] L. Hérault, R. Horaud, F. Veillon, and J.J. Niez. Symbolic image matching by simulated annealing. In *Proceedings of the British Machine Vision Conference, Oxford, England*, pages 319–324, September 1990.

[HSV90] R. Horaud, T. Skordas, and F. Veillon. Finding geometric and relational structures in an image. In *Proceedings of the 1st European Conference on Computer Vision, Antibes, France*, Lecture Notes in Computer Science, pages 374–384. Springer-Verlag, April 1990.

[KD79] J. Koenderink and A.V. Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[Mor93] L. Morin. *Quelques Contributions des Invariants Projectifs à la Vision par Ordinateur*. PhD thesis, Institut National Polytechnique de Grenoble, January 1993.

[MVQ93] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *Proceedings of the Conference on Com-*

puter Vision and Pattern Recognition, New York, USA, pages 543–548, June 1993.

[Nit88] D. Nitzan. Three-dimensional vision structure for robot applications. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 10(3):291–309, 1988.

[PPK92] S. Petitjean, J. Ponce, and D.J. Kriegman. Computing exact aspect graphs of curved objects: algebraic surfaces. *International Journal of Computer Vision*, 9(3):231–255, 1992.

[RZFM92] C.A. Rothwell, A. Zisserman, D.A. Forsyth, and J.L. Mundy. Fast recognition using algebraic invariants. In J.L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, chapter 20, pages 398–407. MIT Press, 1992.

[SH92] H. Sossa and R. Horaud. Model indexing: the graph-hashing approach. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Urbana-Champaign, Illinois, USA*, June 1992.

[TM87] D.W. Thompson and J.L. Mundy. Three-dimensional model matching from an unconstraint viewpoint. In *Proceedings of* IEEE *International*

Conference on Robotics and Automation, Raleigh, North Carolina, USA, pages 208–220, 1987.

[Wei92] I. Weiss. Noise-resistant projective and affine invariants. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Urbana-Champaign, Illinois, USA*, pages 115–121, June 1992.

List of Figures

1 The geometric setup. 32

2 A first example of matching 32

3 A second example of matching 32

4 Noise correction 33

5 An example of correction 33

6 8 images of a same object 33

7 Eighty views of an object 34

8 Two images of each group 35

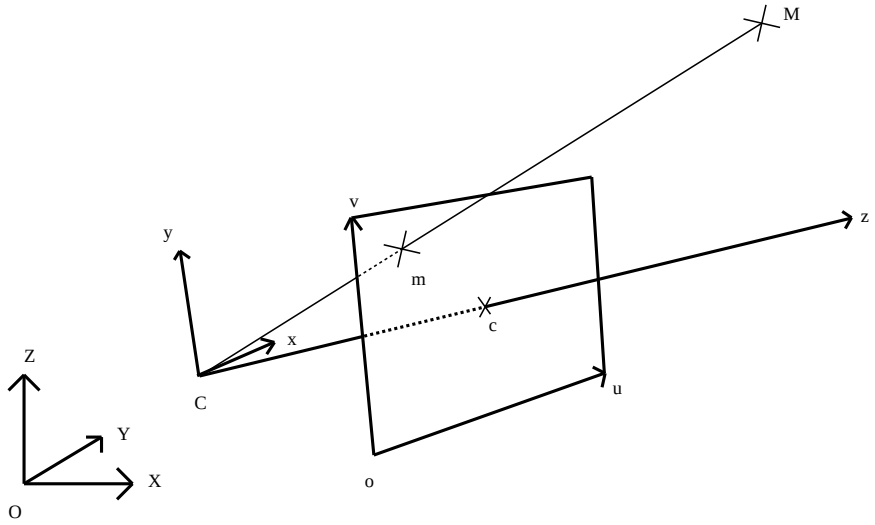

Figure 1: The geometric setup.

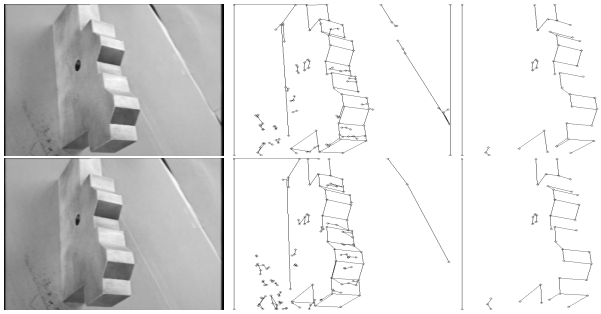

Figure 2: A first example of matching

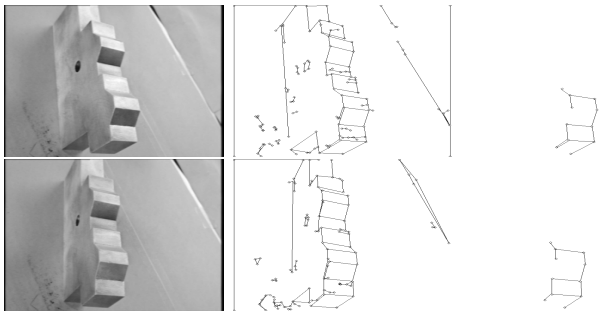

Figure 3: A second example of matching

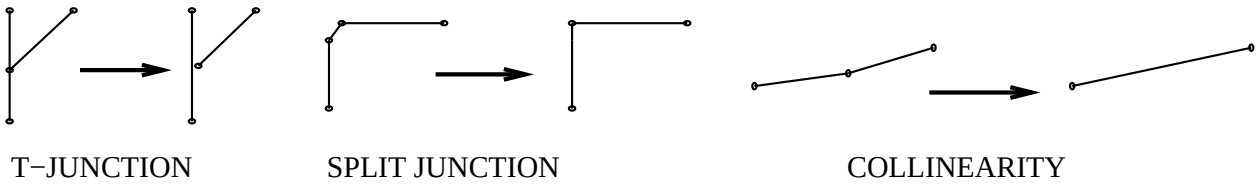

Figure 4: Noise correction

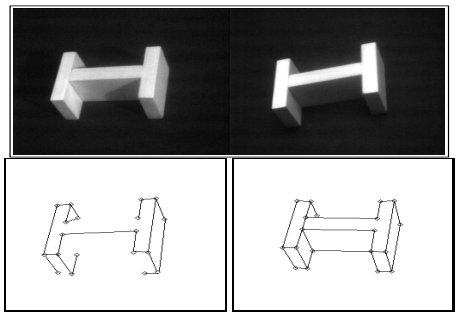

Figure 5: An example of correction

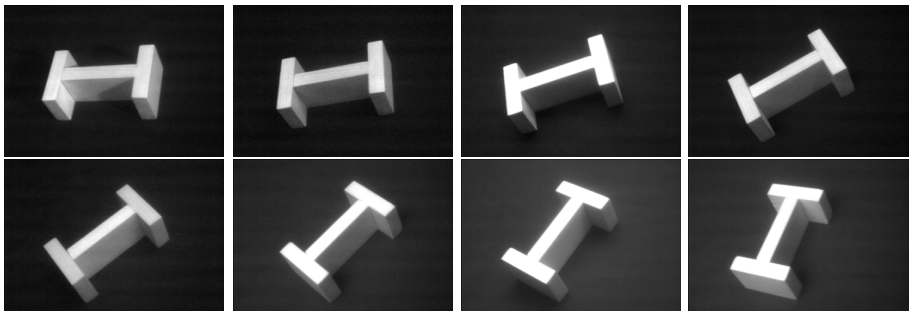

Figure 6: 8 images of a same object

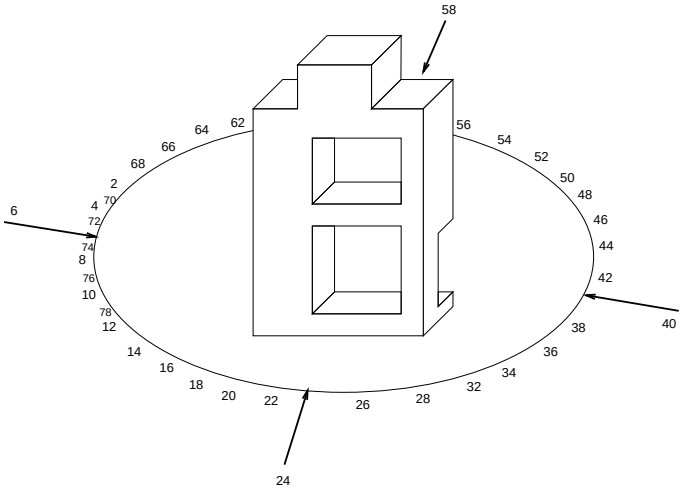

Figure 7: Eighty views of an object

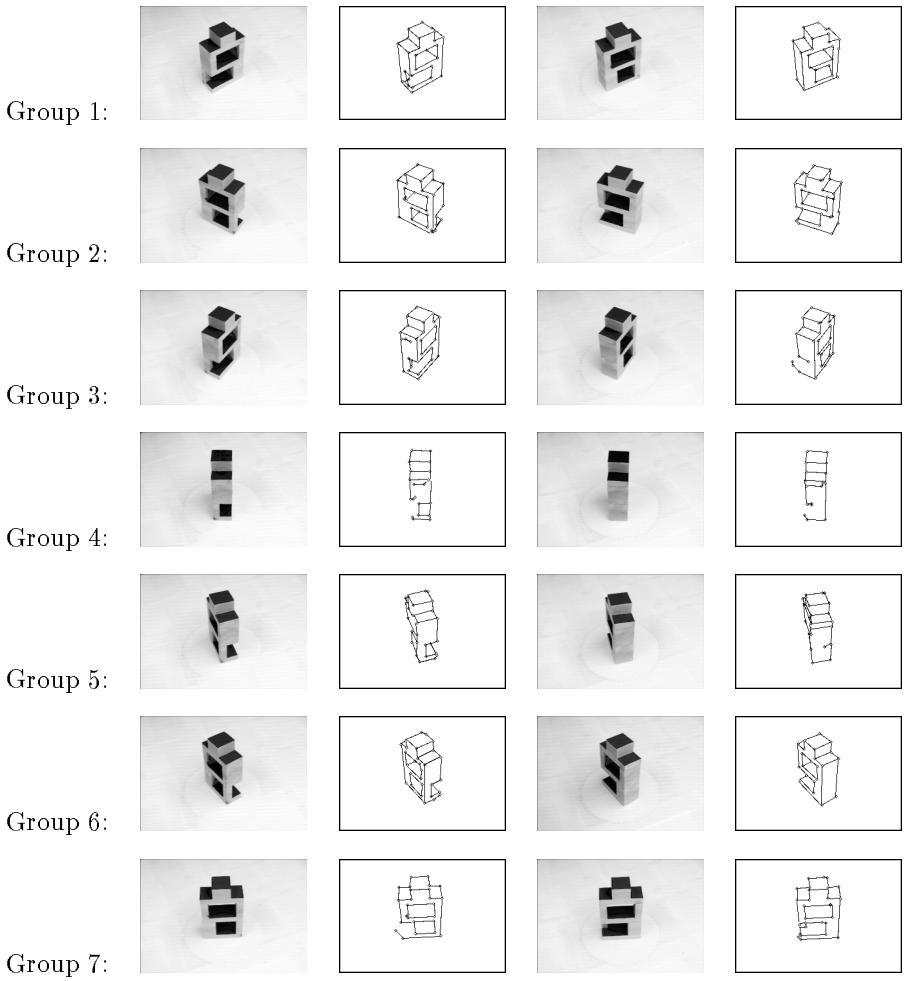

Figure 8: Two images of each group