Using quasi-invariants for automatic model building and object recognition: an overview.

Patrick Gros
Lifia - Inria Rhône Alpes
46, avenue Félix Viallet, 38031 Grenoble Cedex 1, FRANCE

Abstract

We address the problem of automatic model building for further recognition of objects. Our initial data are a set of images of an object. In a first stage, these images are put into correspondence using quasi-invariants, epipolar geometry and an approximation of the apparent motion by an homography. The different aspects of the objects may thus be computed and each aspect gives raise to a partial model of the object. In a second stage, these models and indexed in a data base which is used for recognition. This work is based on the idea that aspect graphs may (should?) be learned from examples rather than computed from CAD models, and that a planar representation associated with geometric quasi-invariants is a relevant tool for object recognition.

1 Introduction

This paper takes place in the frame of object recognition. This problem may be stated as follows:

- a system contains some "knowledge" about a collection of objects;
- when a new image is given to it, the system can determine if some of those objects are present in this image.

Usually the knowledge contained by the system is a set of what is called models, one for each objet, organized in a data base. When designing such a system, one has thus to decide which information is necessary to recognize an object, how this information can be obtained, i.e. how the models have to be computed.

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 keep away from 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.

The current approaches to this modeling problem may be classified according to two criterions: the kind of data used to construct the model and the dimension of the model constructed. 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 modeled object [KD79, PPK92]. Each aspect is topologically different from the others and they are organized 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. Modeling 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] uses occlusion contours, Mohr et al. [MVQ93] use points, Rothwell et al.[RZFM92] use numerical invariants associated with some configura-

tions of points, lines and curves, Weiss uses differential invariants associated with algebraic curves [Wei92].

Our approach falls in the last category. It relies on the idea that aspect graphs should be learned from examples rather than computed from CAD models, and that a planar representation associated with geometric quasi-invariants is a relevant tool for object recognition.

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 approximately 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 complete 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. computation 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 modeling 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 the two main stages of the method. The algorithm to find the correspondence between two images is described in section 2; section 3 explains how to go from these correspondences to the model, and particularly shows the learning ability of this process.

2 Matching sets of 2D features

2.1 The matching algorithm

The aim of 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 approximately known and the location of one feature in an image may be deduced from the location of the corresponding feature in the second image. Foe example, this is assumed 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.

Our method is based on the following idea: matching would not be a problem if corresponding features were in the same place in the two images to be matched. This difference of position is called "apparent motion". This motion is not a classical geometric transformation because

two different features in one image can correspond to only one feature in the second one. On the other hand, in many cases, it can be approximated by a transformation like a similarity or an affine transformation. Our method consists in computing such an approximation. It does not assume that the camera motion is known or that it is very small. It also works with noisy images or occluded objects.

The different stages of our matching method are the following:

1. We have two images containing line segments approximating contour curves. We assume that segments' apparent motion between the two images is a similarity (resp. an affine transformation). We associate numerical invariants to some feature configurations: angle and length ratio defined by every pair of segments having an extremity in common (resp. affine coordinates associated with every set of four connected segment vertices).

2. Invariants and their corresponding segments and vertices are matched according to the invariants' value. As there is some noise in the images, equality is tested up to a noise threshold, in consequence of what all matches are not right.

3. To eliminate wrong matches, a Hough transform technique is used, in order to evaluate the parameters of the approximation of the apparent motion. As a matter of fact, the right matches define the same approximation and the computation of this motion allows to recognize them. When two invariants are matched, there is enough geometrical information to compute the transformation. In our case, when two configurations are matched, it is possible to compute the parameters of the similarity (resp. the affine transformation) which transform one of the two configurations 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 and define the best estimate of apparent motion. Every match which gives a transformation far the best estimate is eliminated.

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

2.2 Experimental results

In this paragraph, we provide some results that show that the algorithm is robust, even when the apparent motion is far from an exact similarity or affine transformation.

Figure 1 shows the match obtained with two images. The corresponding vertices have the same number in the two images. The two upper images were matched using similarity invariants, while we used affine invariants in the two lower images. The first image contains 132 vertices, the second one 105. With the similarity invariants, we obtained 24 correspondences and 25 with the affine invariants. In both cases, all the correspondences are right.

After this first matching stage, it is now possible to compute the epipolar geometry of the two images, or an approximation of the apparent motion by a homography. This information may then be used to detect any eventual wrong match, and to find other correspondences.

3 From matching to modeling

3.1 The modeling algorithm

The matching algorithm is the central point of our method. As a matter of fact, it allows to find the characteristic and robust features of one object represented in two noisy images. A model of this object to be used in a recognition system has to contain this information: the robust and characteristic features that will appear in every image where the same aspect of the object is visible.

The only problem is that two images may not be sufficient to recover the whole characteristic structure of one aspect of the object if they are very noisy. In the case where we have three or more images, we use the following stages:

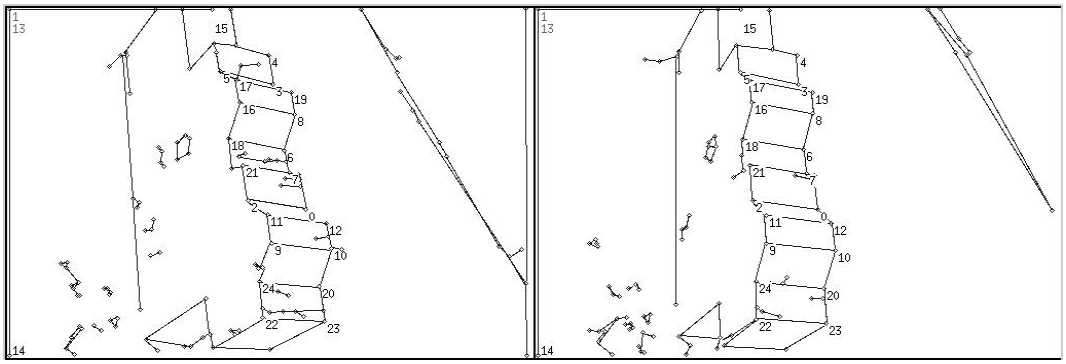
1. the images are matched two by two;

2. as incoherences may occur, we compute a global match;

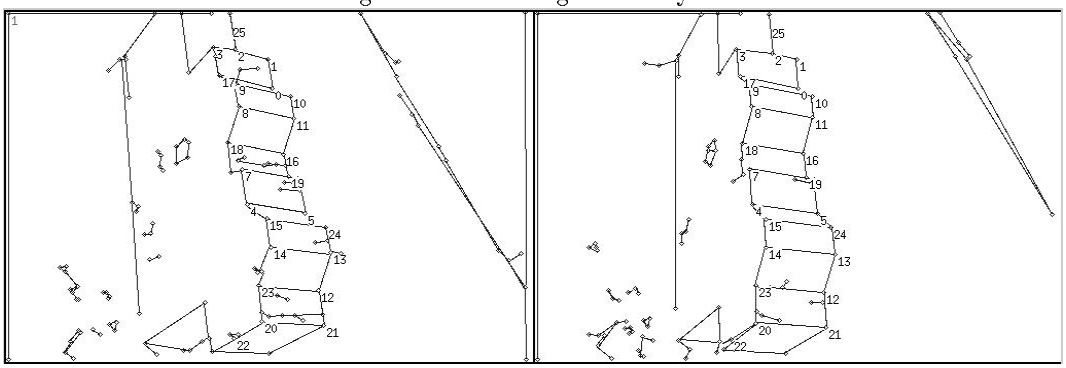
3. the features that appear at least in 60 percents of the images are put in the model; the position of the modeled feature is the average of the corresponding features in the images, after correction of the apparent motion by an homography.

This algorithm assumes that the three images represent approximately the same aspect of the object. In the opposite case, we add two more steps:

1. the images are matched two by two and the distance between two images is defined as the percentage of matched features;



Two images matched using similarity invariants



Two images matched using affine invariants

Figure 1: Experimental matching results

2. all the distances are put in a distance matrix and an agglomerative clustering algorithm is used to gather the images according to the aspect of the object they represent;

3. a global match is computed for all the groups found at the previous step;

4. one model is deduced from each group; in this case the model of an object is composed of a set of partial models.

In both algorithms, the most difficult step is the one consisting in going from the first matches to a global match. To do that, we represent the different matches as a graph; the vertices of this graph are the features of each image, and the edges represent the correspondences. We then consider the different connected components. They usually contain to many features, and we cut them up in more strongly connected parts until they reach an acceptable size. Two vertices a and b are strongly connected if they are connected and if there exist some other vertices c_i which are connected with both a and b. The degree of connectedness is then given by the number of the existing vertices c_i. A component is of acceptable size if it contains at most one feature of each image and at least $0.8n$ features where n is the number of images.

3.2 Towards recognition

As this part of the work is still under development, this paragraph only gives an overview of the subject. For each object we want to recognize, we learn its model from a set of images, taken from various viewpoints. All the models all gathered in a model base.

The recognition problem may then be solved as a correspondence problem between a model and a new image. The matching algorithm presented in section 2 may be adapted with this aim in view.

For a kind of transformation (affine or similarity), the invariants of all models are put in a single table. Some invariants are also computed from the image and compared with the first ones. When a correspondence is found, that gives a vote to a model. This vote is expressed as a point in a transformation parameter space. When all the possible correspondences are made, we count the coherent votes of each model, following the method presented in section 2.1.

This allows to predict the models which have the greatest probability to appear in the image. This has to be followed by a verification stage. The features corresponding to the model are removed from the image, and the process may be repeated to find another object.

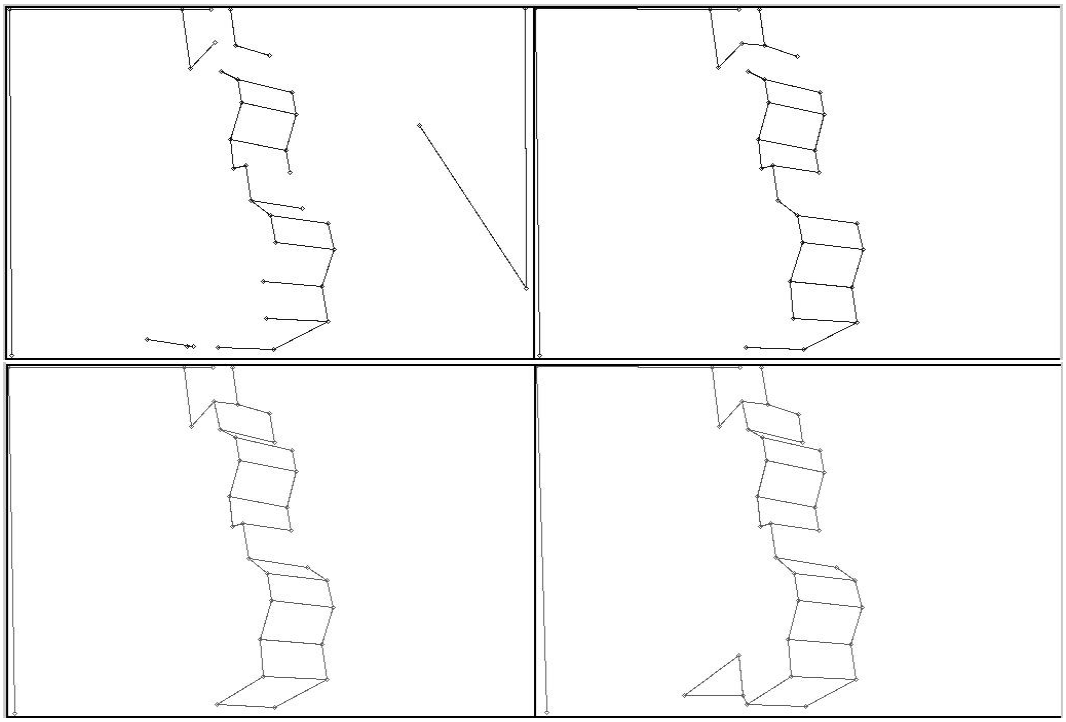


Figure 2: Models of an object computed from respectively 2, 4, 6, and 8 images of it.

3.3 Experimental results

Some results obtained with the second algorithm of section 3.1 are provided in [Gro93]. So, we show here some results using the first one.

The model of an object is computed from respectively 2, 4, 6, and 8 images of it. These images are of the same quality as the ones shown on Figure 1. Figure 2 shows the results.

It is clear that the most numerous the images are, the best the result is. The model is learned from the images, and all noise is eliminated. A few segments of the object are missing: one could thought that were a important part of the object structure; in fact, they are not reliable with respect to the acquisition device.

4 Conclusion

In this paper are presented a algorithm for image matching, in the case where images contain line segments, and a method to construct a model from a set of images of an object.

Even if the learning ability of this last method is clear according the provided results, the use of such models in a recognition system causes other difficulties, like fast indexing, that have still to be studied.

The main contributions of this work are:

- the use of local quasi invariants as a robust and discriminant feature in images; they have proven to be more usable than topological structures like subgraphs or even than exact invariants which are sensitive to noise and difficult to compute for non trivial objects;

- a new matching algorithm which works for images containing segments, even if the motion of the camera is unknown; this geometric method is a real alternative to the often used correlation and relaxation techniques;

- a method for "modeling from examples", which can compute a model for the main aspects of an object without computing the aspect graph of the object from a CAD model.

Acknowledgments. This work has been sponsored by European Esprit project No 6769 (the SECOND Project). The author also acknowledges Edmond Boyer and Olivier Bournez for their participation to the project, and Françoise Veillon for her fruitful comments.

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 PAMI*, 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.

[Gro93] P. Gros. Matching and clustering: two steps towards automatic model generation in computer vision. In *Proceedings of the AAAI Fall Symposium Series: Machine Learning in Computer Vision: What, Why, and How?, Raleigh, North Carolina, USA*, pages 40–44, October 1993.

[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 relationnal structures in an image. In *Proceedings of the 1st European Conference on Computer Vision, Antibes, - France*, 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.

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

[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.