N90-29802

Trinocular Stereovision using Figural Continuity, Dealing with Curved Objects

R. Vaillant and O.D. Faugeras
INRIA Domaine de Voluceau
BP 105, 78153 Le Chesnay Cedex France

Abstract

In this paper, we present a method to build a dense and reliable 3D description of a scene from three digital images by means of passive stereovision. Our method uses figural continuity to improve the results of a previously developed algorithm [AL87]. In particular, it copes much better with curved objects. It produces results which are organized as three-dimensional chains of segments.

1 Introduction

We want to build a three dimensional description of a scene observed from several viewpoints. We need a description which is dense enough to allow an analysis of the scene and, in particular, makes it possible to identify some of the objects present.

We start with the results of a fast and accurate algorithm of trinocular stereovision [AL87]. We improve on this result by using figural continuity and local support. We also compute thredimensional results which are a good representation of the outline of the objects which are visible.

Our method has the following attractive features : flexibility (an automatic calibration technique is used to calculate precisely the intrinsic parameters and the position and the orientation of the cameras), density (in our experiments more than 50% of the image segments are matched), accuracy (less than 2% of the matches found are false), organized (the reconstructed 3D segments are structured as 3D-chains which represent a part of the outline of an object).

2 Edge Detection

The features of the image which are used by the stereovision algorithm are line segments. The basic idea of stereovision is to use some features which can be detected reliably in the image and match them between images taken from several viewpoints. Therefore we search, in the image, for some characteristic points : the edge points. They are usually defined as zero crossings of the laplacian [Gri84] or local maxima of gradient. Such points are generally the images of classes of significant physical attributes of the scene : surface discontinuities, shadows, markings on objects. These points can be computed efficiently and accurately and organized in edge chains [Der87],[Gir87]. An edge chain is a list of pixels corresponding to edge points. They do not have any triple point. From them, we build up a lists of line segments by using a polygonal approximation algorithm [Ber86]. These line segments are the features which are used by the stereo algorithms developed at INRIA [AL87].

Using line segments to represent image contours has the great advantage to reduce the amount of data to be processed and yields fast matching algorithms and very good results for scene containing mostly polyhedral objects. When curved objects are present, two problems occur which are caused by the unstability of the polygonal approximation.

Indeed, stereo algorithms can work only if the features used can be extracted from the images in a stable fashion : a real physical element, as a part of an object, must have similar attributes when viewpoint varies slightly. This is generally true for the edge points. Unfortunately the polygonal approximation algorithm is not always stable. This is especially true if the observed contour is curved. The matching of line segment is more difficult in this case.

When a match can be made, we want to reconstruct the 3D line segment which projects on the matched segments : we must select in each of them the parts which are common in terms of the epipolar strip. This part may be only a small component of the initial segment. In this case, we are unable to reconstruct a large part of the contour. As a further consequence the reconstructed 3D segments are not generally connected. This is an important problem, for later analysis of the 3D scene. To solve these two problems, we exploit the connectivity information which is present in the polygonal approximations of the edge chains. This is equivalent to the use of figural continuity.

3 Propagation using figural continuity

As a start, we use the matches found by an accurate and fast stereo algorithm [AL87]. This algorithm uses heavily the epipolar constraints and some compatibility constraints for the length and orientation of the matched segments. We propagate these matches by using figural continuity, i. e. the information provided by the edge chains.

This propagation can be easily achieved if we consider only segments in two images. We are not therefore obliged to restrict ourselves to the areas which are seen by the three cameras. However for reconstruction, we use once again the three images : since this decreases the uncertainty of the reconstructed segments.

To achieve this propagation, we apply rules and decide to match some segments or a part of a segment by scanning the chain to which their belongs.

Let us introduce some notations : $C^{i,a}$ is the i-th chain in the plane image of camera a. A chain is characterized by its number n of segments. We enumerate these segments from 1 to n. Segment p of chain $C^{i,a}$ is denoted $C_p^{i,a}$. A match, between two segments is noted $[C_p^{i,a}, C_q^{j,b}]$. For each chain, we say that the first segment is the head and the n-th segment is the tail.

As all epipolar lines intersect at the epipole, we can define a direction of propagation on the chain. For each segment on the chain, we call A its endpoint which is toward the head and B its endpoint which is toward the tail. The match $[C_p^{i,a}, C_q^{j,b}]$, defines an interval $[A_a, B_a]$ on segment $C_p^{i,a}$ and an interval $[A_b, B_b]$ on segment $C_q^{j,b}$. These intervals are computed using the epipolar constraint. Two solutions are possible A_a matches A_b or A_a matches B_b. In the first case, the directions of propagation for the match $[C_p^{i,a}, C_q^{j,b}]$ are the same for the chains $C^{i,a}$ and $C^{j,b}$. In the second case, the directions of propagation are inverted : if we walk on chain $C^{i,a}$ from the head to the tail, then we walk on chain $C^{j,b}$ from the tail to the head.

We can now define the following rules used in the propagation :

- Rule 1 (figure 1)

  Rule 1 deals with the case where we have two pairs of matched segments.

  If the following conditions are true. Segment $C_p^{i,a}$ matches segment $C_q^{j,b}$. Segment $C_{p'}^{i,a}$ matches segment $C_{q'}^{j,b}$. For each integer r between p and p', segment $C_r^{i,a}$ has not yet found any match

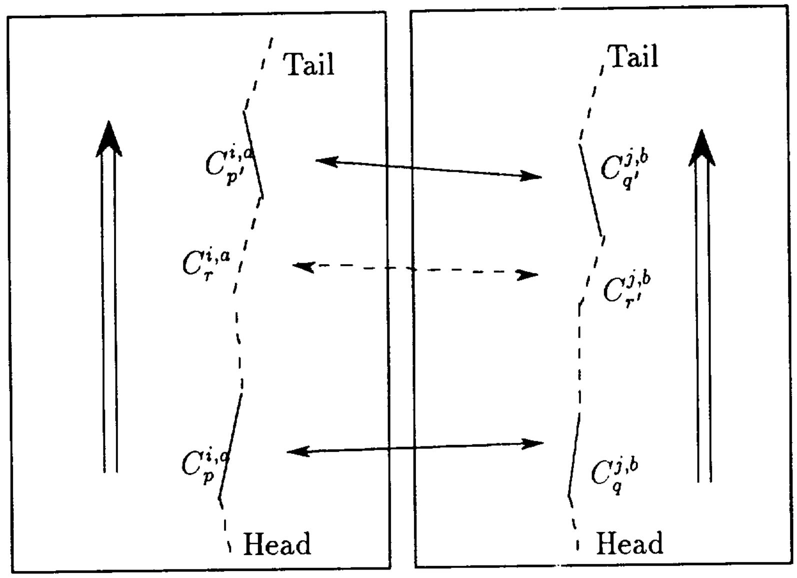


Figure 1: Propagation rule 1

in image b. For each integer s between q and q', segment $C_s^{j,b}$ has not yet found any match in image a.

Then, for each integer r between p and p', segment $C_r^{i,a}$ matches one of the segments $C_s^{j,b}$ of image b with s between q and q'.

- Rule 2 (figure 2)

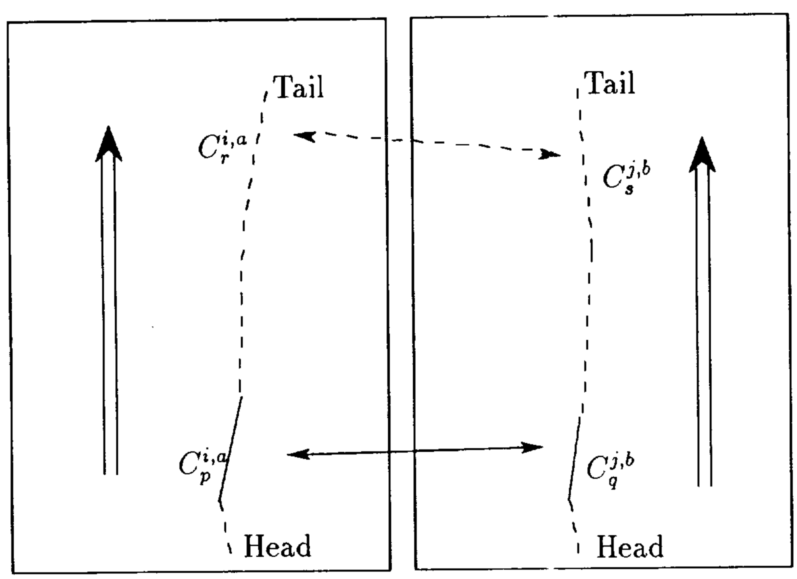


Figure 2: Propagation rule 2

This rules deals with the case where two segments match and all the segments from them to the end of the chain have not yet found any match.

If the following conditions are true. Segment $C_p^{i,a}$ matches segment $C_q^{j,b}$. For each integer r such that $r > p$, segment $C_r^{i,a}$ has not yet found any match in image b. The match $[C_p^{i,a}, C_q^{j,b}]$ defines a direction of propagation on the chain $C^{j,b}$. Two cases are possible but they are analogous. With respect to the match $[C_p^{i,a}, C_q^{j,b}]$, and the direction choose on the chain $C^{i,a}$, we consider that the segments following the segment $C_q^{j,b}$ are the segments $C_s^{j,b}$ with $s > q$ and the condition is : for each integer s, with $s > q$, the segment $C_s^{j,b}$ has not yet found any match in image a.

Then, for each integer r such that $r > p$, segment $C_r^{i,a}$ matches one of the segments $C_s^{j,b}$ with $s > q$.

This rule allows us to match the small segments, which sometimes are near the endpoints of chains.

- Rule 3 (figure 3)

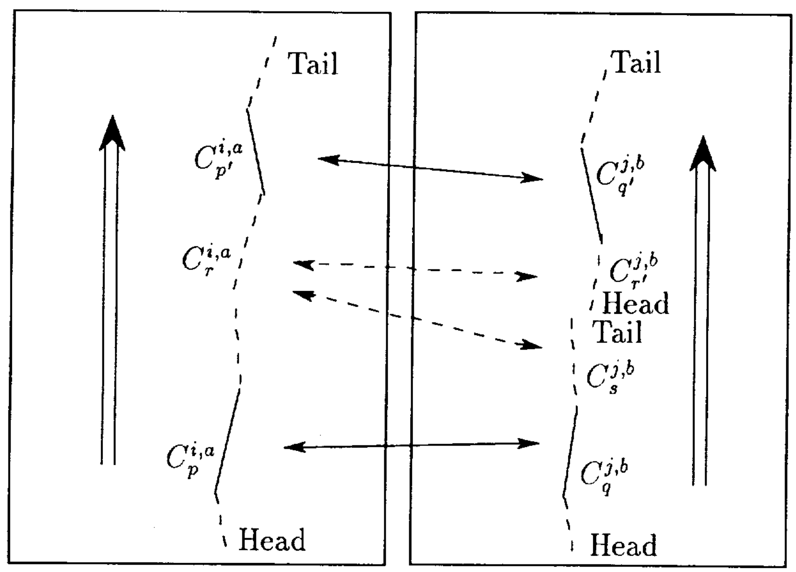


Figure 3: Propagation rule 3

This rule deals with the case where the matching chain of a chain has been cut in two chains in the opposite image by the segmentation process.

If the following conditions are true. Segment $C_p^{i,a}$ matches segment $C_q^{j,b}$. Segment $C_{p'}^{i,a}$ matches segment $C_{q'}^{j',b}$. To simplify, we suppose that $p < p'$ and that the epipolar constraint implies that the segment following the match $[C_p^{i,a}, C_q^{j,b}]$ are the segments $C_s^{j,b}$ with $s > q$, and the segment before the match $[C_{p'}^{i,a}, C_{q'}^{j',b}]$ are the segments $C_s^{j',b}$ with $s < q'$. All the possible cases are analogous. For each integer r between p and p', segment $C_r^{i,a}$ has not yet found any match in image b. For each integer s such that $s > q$, segment $C_s^{j,b}$ has not yet found any match in image a. For each integer s such that $s < q'$, segment $C_s^{j',b}$ has not yet found any match in image a.

Then for each integer r, between p and p', segment $C_r^{i,a}$ matches one of the segments $C_s^{j,b}$ with $s > q$ or one of the segments $C_s^{j',b}$ with $s < q'$.

With this rule, we have a way to correct some errors in the chaining process.

These rules define how line segments can be broken up into subsegments which are matched in the other image. We recursively compute them.

We begin with the match $[C_p^{i,a}, C_q^{j,b}]$ used to activate the rule. We compute the first interval which follows this match on the chain $C^{i,a}$. We calculate the epipolar strip of this part in the image b and take the intersection with the chain $C^{j,b}$. This part is not generally a complete segment. Therefore we must divide the segment in several parts which do not overlap. We implement the unicity constraint of the image of a 3D feature . We have thus created a new match which we use to continue this process.

This technique has the disadvantage of sometimes creating small partitionings on the segments. It therefore gives us some problems when we reconstruct the 3D segment.

These rules are applied to all pairs of cameras x, y. Some contradictory situations are possible with the third camera. This is uncommon because the epipolar constraint gives a very precise information. The contradictions are then solved during the reconstruction process.

A further unusual case is provided by the closed chains. We solve it by giving them an infinite length : the segment $(n + 1)$ is the segment 1 if the chain has n segment. The rules, that we have defined, can then be applied without difficulty.

This propagation is computed very rapidly and the rules give very few false matches because edge chains are nearly always similar created in the different images and the matches used to begin the propagation are very accurate.

4 Matching using neighbourhood

Unfortunately, with the previous technique, we cannot define matches for the chains for which no match was initially found.

Therefore, we have developed a technique to describe the local environment of a segment. We first find the segments which are nearest to the segment $C_p^{i,a}$. We organize them into several classes. They are close to parallel to segment $C_p^{i,a}$. They form a T-shape with segment $C_p^{i,a}$. They have an endpoint near to one of $C_p^{i,a}$.

For a segment $C_p^{i,a}$, which has not yet found any match, we consider especially the segments which already match one segment in image b. Let us imagine that for example, segment $C_{p'}^{i',a}$ is a neighbour of segment $C_p^{i,a}$, and that segment $C_{p'}^{i',a}$ matches segment $C_{q'}^{j',b}$.

We define the set of segments of image b, which are candidates to be matched with the segment $C_p^{i,a}$, as all the segments $C_q^{j,b}$ which intersect the epipolar strip of the segment $C_p^{i,a}$ of image b. Among these, we look for all the segments $C_{q'}^{j',b}$ which are neighbours of segment $C_q^{j,b}$ and have the same position with respect to $C_q^{j,b}$ as segment $C_{p'}^{i',a}$ with respect to segment $C_p^{i,a}$.

When we explore all the segments which are neighbours of the segment $C_p^{i,a}$, we find, where possible, one or several segments, which are likely matches for segment $C_p^{i,a}$. We test if we have found enough neighours of segment $C_p^{i,a}$ and we test if the different candidates are compatible : the intersection of their epipolar strip with segment $C_p^{i,a}$ do not overlap. If these tests are false, we reject the match.

This matching technique requires a lot of computation to find the neighbours of a given segment. To decrease the search area, we organize the data. The image is divided in buckets and for each of these buckets, we create the list of all the segments which intersect it. When we search for neighbours, we need only to explore the lists of the buckets near the segment currently being processed.

5 3D Reconstruction

3D reconstruction is one of the most important problems for stereovision algorithms. Our aim is to create spatial segment chains which are connected like the edge chains in the images.

Let us suppose that we match a set of m points M_i in m images. These points are the endpoints of part of the segments defined using the epipolar constraint. The coordinates of M_i in the image are : (u_i, v_i), for $i = 1, \ldots, m$. The coordinate $X = [x, y, z]^t$ of the spatial point M, whose images in the cameras are the points M_i are found by solving the equations :

$$\begin{cases} l_1^i X - u_i l_3^i X + l_{14}^i - u_i l_{34}^i = 0 \\ l_2^i X - v_i l_3^i X + l_{24}^i - v_i l_{34}^i = 0 \end{cases}$$

for $i = 1, \ldots, m$, ([FT86]). We have therefore a system of $2m$ linear equations in the three unknowns x, y, z which, in the exact case, has a unique solution. We can solve it in the real case by Kalman filtering which allows us to take into account both the noise on the images coordinates (u_i, v_i) and the result of calibration ([AF87]). We assume that the noise is gaussian. This noise appears with the digitization process, the edge detection and the polygonal approximation. The Kalman filtering yields a best estimation of X and its covariance matrix Λ.

Let us consider two spatial points M and P. We have computed them using Kalman filtering. We need to decide if they correspond to the same physical points. We call their covariance matrices Λ_M and Λ_P.

We assume that M and P are two independent gaussian points. Therefore the covariance matrix of the gaussian vector MP, is the sum $\Lambda_M + \Lambda_P$. If M and P are two instances of the same gaussian point MP, then the expected value of MP is 0 and the quantity

$$d^2(M, P) = MP^T (\Lambda_M + \Lambda_P)^{-1} MP$$

has a χ^2 distribution with 3 degrees of freedom (supposing that all points are gaussian). This quantity is the Mahalanobis distance. If $d^2(M, P)$ is less than some threshold s, the points M and P have a probability p (computed from χ^2 tables) of being two instances of the same physical point and we can fuse them. It is easy to build the fused point N by Kalman filtering. The uncertainty about the point N is less than that of M and P.

Notice that the Mahalanobis distance can be computed easily even though we must invert a 3×3 matrix. In fact the covariance matrix is symmetric and it is so possible to explicitly compute the expression of $d^2(M, P)$.

Thus we have a powerful tool for deciding to fuse the endpoints of a reconstructed segment. This allows us to build connected chains.

Indeed, let us consider a chain $C^{i,a}$ of image a. We reconstruct successively all the segments for this chain, if they are matched. If we find a match $[C_p^{i,a}, C_q^{j,b}]$, we also try to find the match with the third image c. It is the match $[C_p^{i,a}, C_r^{k,c}]$ and $[C_q^{j,b}, C_r^{k,c}]$. These last two matches do not always exist because they can be the image of a 3D segment which is not seen by the three cameras. We compute the endpoints of the 3D segment as described previously. For two segments $C_p^{i,a}$ and $C_{p+1}^{i,a}$, which follow each other on the chain $C^{i,a}$, we can decide using the Mahalanobis distance, to fuse the endpoints which are neighbours.

This allows us to build connected spatial segment chains which represent parts of an object. These chains can also be used for surface fitting algorithms or for object recognition and localization ([SS87]).

		doll	satellite
Segment Number	Cam 1	736	241
	Cam 2	836	217
	Cam 3	552	287
Segment Length	Cam 1	9600	6746
	Cam 2	9315	5278
	Cam 3	6822	7033
Initial Matches	Number	17.7%	27%
Initial Matches length	Cam 1	42.6%	31.1%
	Cam 2	41.1%	39.0%
	Cam 3	45.7%	29.2%
All Matches found	Number	63.3%	75%
All Matches found length	Cam 1	68.1%	48.4%
	Cam 2	66.3%	61.86%
	Cam 3	58.8%	45.72%

6 Experimental Results

Our algorithm is implemented in C.

We have tested it on a variety of images. Here we show the doll image (figure 5) and the satellite image (figure 7). For each scene, we show the polygonal approximations of the edge chains in the three images and the matches found, by projecting them back, on planes.

In the result of our algorithm, we notice a considerable improvement as for example on the head of the doll and the digits 5 in figure 5.

For each figure we give the number of segments for each image, the length, in pixels , of the segments in each image, the number of initial matches (it is a rate for the number of segments in the image of the first camera), the length of the parts of segments which are reconstructed in each image when using only the initial matches (it is a rate for the length, in pixels , of the segments in each image), the number of all matches found, (it is a rate for the number of segments in the image of the first camera), the length of the parts of segments which are reconstructed for each cameras when using all the matches. (it is a rate for the length, in pixels , of the segments in each image).

First we can note that the number of segments which are matched is considerably increased by our algorithm. In fact all the small segments are not matched by the algorithm used to find the first match because it is to difficult to find an accurate match for them among the three camera. With our algorithm it is now possible to take into account the local information and so we can match them using the information provided by the match of a longer segment of the same edge chain.

Since we subdivide the segments, the most characteristic information for the comparaison between the two algorithm is the length of the parts of segments which are reconstructed.

We also show the 3D reconstructions. We project them to a plane, which is not the same as those of the camera image plane.

For the propagation phase, the computation is approximately 30 seconds on a SUN-3 Workstation. The matching technique using an analysis of neigbourhood takes between 2 and 3 minutes. The spatial reconstruction of the segments also takes several minutes and depends on the number of matches found.

We can now present a very simple example of the use of the spatial chain for finding some structure in the 3D data. The key idea is to scan a spatial chain and to apply a least square on the segments.

For example we search a plane by applying least square algorithm for each endpoint of the segments of the chain. It calculates the best plane. After we look for the point which is the more longer from this plane and we can do a threeshold on this distance to decide that some chains are in a plane. For example, in the case of figure 5, all the points in the grid, which formed in the background of the scene, we determine this plane. All these points are less than 3 mm away from this plane.

7 Conclusion

We have described a stereovision algorithm which is a mixture of binocular and trinocular stereo. The main points of our approach are the following :

- It gives three dimensional maps which are much denser than in the trinocular stereo case by exploiting figural and neighbouring continuity.
- The maps are represented with connected segment chains.
- The uncertainty of the data has been reduced.

References

[AF87] N. Ayache and O.D. Faugeras. Building a Consistent 3d Representation of a Mobile Robot Environment by Combining Multiple Stereo Views. In *International Joint Conference on Artificial Intelligence*, August 1987.

[AL87] N. Ayache and F. Lustman. Fast and Reliable Passive Trinocular Stereovision. In *First International Conference on Computer Vision*, June 1987.

[Ber86] M. Berthod. Approximation polygonale de chaînes de contours. Programmes C, 1986. INRIA.

[Der87] R. Deriche. Using Canny's Criteria to Derive an Optimal Edge Detector Recursively Implemented. In *The International Journal of Computer Vision*, pages 15–20, April 1987.

[FT86] O.D. Faugeras and G. Toscani. The Calibration Problem for Stereo. In *Proceedings of CVPR'86,Miami Beach,Florida*, pages 15–20, 1986.

[Gir87] G. Giraudon. *Chaînage efficace de contour.* Rapport de Recherche 605, INRIA, Février 1987.

[Gri84] W. Eric L. Grimson. *Computational Experiments with a Feature Based Stereo Algorithms.* A.I. Memo 762, MIT, January 1984.

[II86] M. Ito and A. Ishii. Three-view stereo analysis. *IEEE Transactions on PAMI*, 8:524–531, july 1986.

[SS87] J.T. Schwartz and M. Sharir. *Identification of Partially Obscured Objects in Two and Three Dimensions by Matching of Noisy 'Characteristic Curves'.* Technical Report, Current Institute of Mathematical Sciences, New York University, 1987.

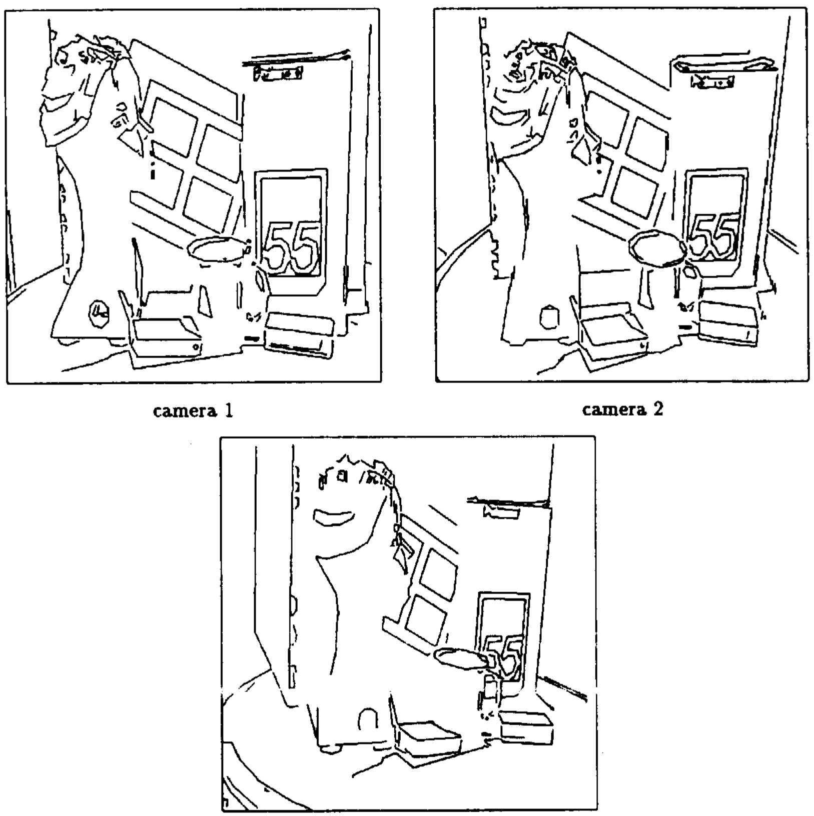


Figure 4: Scene with a doll : polygonal approximation

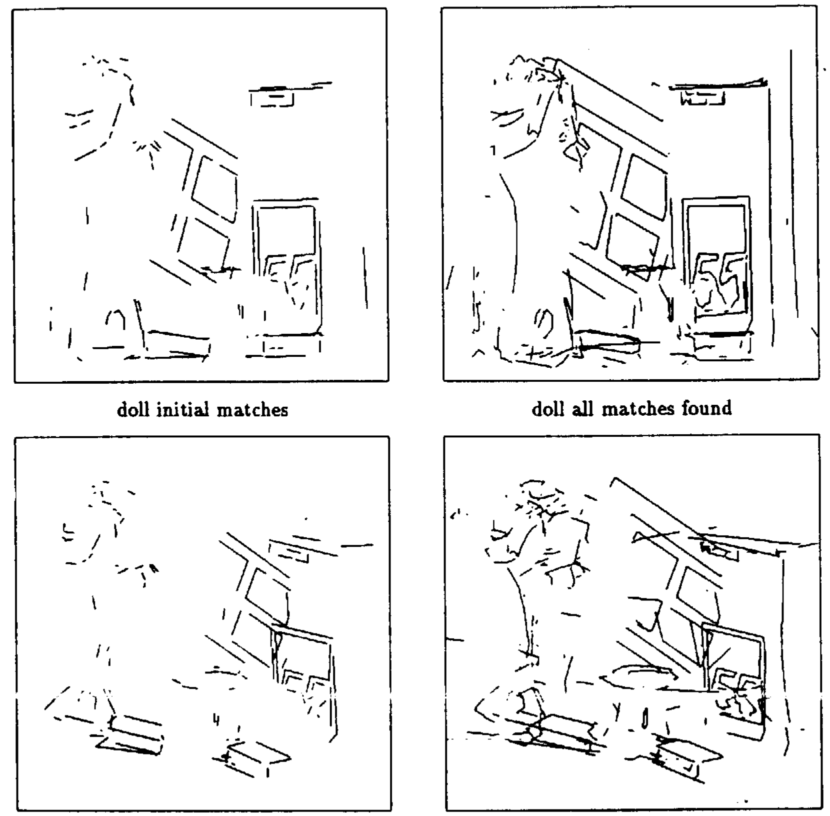
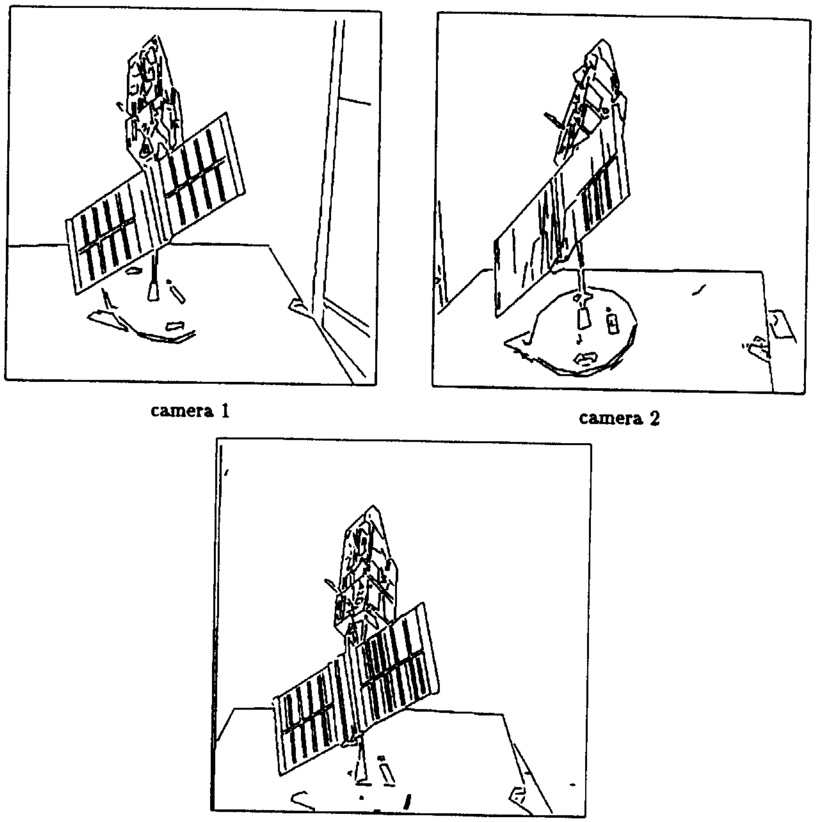


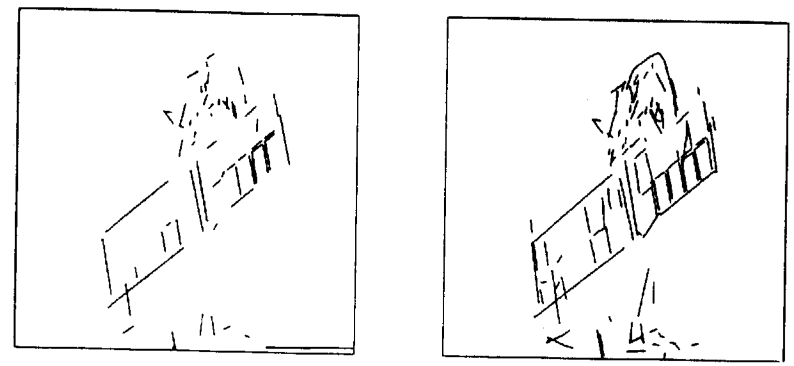
Figure 5: Reconstruction for the scene with a doll :Two projections planes are presented.



camera 1 camera 2

camera 3

Figure 6: Satellite : Spot Polygonal Approximation



initial matches all matches found

Figure 7: Reconstruction : satellite spot