

Purdue University

Purdue e-Pubs

Department of Computer Science Technical
Reports

Department of Computer Science

1991

Correspondence Problem in Image Sequence Analysis

Chia-Hoang Lee

Anupam Joshi

Report Number:

91-041

Lee, Chia-Hoang and Joshi, Anupam, "Correspondence Problem in Image Sequence Analysis" (1991).
Department of Computer Science Technical Reports. Paper 882.
<https://docs.lib.purdue.edu/cstech/882>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

**CORRESPONDENCE PROBLEM IN
IMAGE SEQUENCE ANALYSIS**

Chia-Hoang Lee
Anupam Joshi

CSD-TR-91-041

May 1991

Correspondence Problem in Image Sequence Analysis

Chia-Hoang Lee *
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907

Anupam Joshi
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907

April 24, 1991

Abstract

In this paper, we propose an algorithm to address the correspondence problem in image sequence analysis. The underlying philosophy of this algorithm is reductionist, and is based on the assertion that measurements pertaining to motion that are obtained from the whole image sequence must be very similar to the measurements that are obtained from corresponding portions of the image sequence. The algorithm has been extensively tested on synthesised data as well as on real image sequences. It appears that the assumption underlying our proposed algorithm is quite useful and applicable to a large class of images. The proposed algorithm has a high degree of inherent parallelism, and is thus suited for implementation on parallel machines.

*The support of NSF under grant IRI-8702053A1 is gratefully acknowledged

1 Introduction

Image sequence analysis is an important task in computer vision and has applications in many areas, for instance, mobile robots. Two different approaches are often distinguished in measuring visual motion, depending on whether the motion is small or large. One approach, called flow-based method, uses local changes in light intensity to compute image flow (or optical flow) at each image point and then computes 3D motion parameters based on the derived flow. The other one, called featured-based method, is based on image features. It first extracts such features as corners, high curvature points, lines from each frame of an image sequence. Next, it establishes the correspondence of these features or markings between two successive frames. Finally, it computes motion parameters and object structure from the positions of the markings in the sequence. The second step is known as the "*correspondence problem*" while the last step is referred to as the "*structure from motion problem*" (motion problem). Both the motion and correspondence problems have to be solved for a complete solution to the image sequence analysis problem.

In contrast to the vast amount of literature devoted to the study of the motion problem [1 - 14], very little literature could be found that addressed the correspondence problem. Almost all the work in dynamic image analysis assumes the correspondence problem to have been resolved. This has resulted in frustration with, and criticism and doubts about any method that uses the feature-based scheme[7]. In this paper, we focus on the correspondence problem that arises in feature-based schemes.

In general, finding corresponding primitives in successive frames relies on a fundamental principle: *the image appearances of an object point in different frames should be similar to each other in some respect*. For example, [19] uses sign change and orientation of the local zero-crossing contours as attributes of candidates, [15] uses photometric information as attributes of an area, [16] attempts to characterize local structure, [17,18] suggest keeping track of features through a small incremental movement.

The main purpose of this paper is to examine the correspondence problem from a purely geometric perspective as opposed to the existing attribute-based methods. Our task, therefore, is to not rely on defining a set of appropriate attributes from image appearance of an object that would be attached to primitives (tokens) and find ways to perform best matches.

Our scheme is also different from the type of schemes proposed in [20,21] . These schemes, for instance, try to minimise the total smoothness deviation for the entire set of trajectories by appropriately assigning the points in distinct frames to each trajectory. The continuity of motion, they argue, provides enough constraints to yield the correspondence.

This does not mean that we advocate ignoring attributes or local structures in the analysis of image sequence. In our view, to solve a real problem, any available information should all be part of the solution. Different approaches to the same problem reinforce each other and resolve any conflicts and ambiguities. The scope of this paper is limited to addressing one of the fundamental problems that must be resolved (and that has largely been left unexplored), if the real problem is to be solved. This technique is based on three assumptions:

1. a rigid motion that underlies the movement of the object.
2. the translation vector is small compared to the distance of the object.
3. a perspective projection model can be used to simulate a camera.

Extensive simulations and real images are used to support the applicability of the proposed method.

2 The Correspondence Problem

2.1 Real Issues in Image Sequence Analysis

Figure 1 shows a sequence of images that were taken outdoors. The field of view of the camera is $24^\circ \times 23^\circ$ and the image size is 484×512 . The goal of the task is to figure out the structure of the layout (buildings, trees) and the sequence of movements of the vehicle.

Figure 2 shows the results of applying sobel edge detector to the images. Although human visual system has already partitioned the images into meaningful parts, these are just a sequence of two dimensional arrays with regard to a machine and there are no meaningful partitions attached to the images yet. This process, often called segmentation, is an ill-defined but very important problem. Although it is an issue that has to be overcome in order

to have a complete solution to the dynamic image analysis problem, this report will not enter into discussions of this aspect.

Figure 3 shows the extraction of a set of distinct tokens. This step is an abstraction of the problem and makes such simplifying assumptions as no missing tokens in successive frames. Another simplification made is that the task of finding correspondence is assumed to have been resolved (mostly through manual operation).

Based on all these simplifications, most of the studies in this area concentrate on designing algorithms to solve for the motion parameters and structure of the objects. It should be of no surprise then that deep doubts of realistic applications of these techniques remain and get strengthened[23,24]. If the real problem is to be solved, then all these simplifications should be removed. In this paper, we propose a technique that removes the assumption of correspondence. Our subsequent work will address the task of getting rid of the other simplifying assumptions.

2.2 Problem Formulation

We assume that the image plane is stationary, and that two perspective views at time t_1 and t_2 , respectively, are taken of a n -point rigid object moving in the 3-D space. The task is to establish corresponding points in the two views. The focal length f is taken to be 1 without loss of generality. Let

$z_i A_i$ = Position vector of the i^{th} point on the rigid object at t_1

$z'_{\sigma(i)} B_{\sigma(i)}$ = Position vector of the i^{th} point at t_2

$A_i = (x_i, y_i, 1)^t$

$B_{\sigma(i)} = (x'_{\sigma(i)}, y'_{\sigma(i)}, 1)^t$

where (x_i, y_i) and $(x'_{\sigma(i)}, y'_{\sigma(i)})$ are, respectively, the image coordinates of the i^{th} point at t_1 and t_2 . Then

$$z'_{\sigma(i)} B_{\sigma(i)} = \mathbf{R} z_i A_i + \mathbf{T} \quad i = 1, \dots, N \quad (1)$$

where

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \text{ is a rotation matrix,} \quad (2)$$

$\mathbf{T} = [t_x \ t_y \ t_z]^t$ is a translation vector, and σ is a unknown permutation.

The problem we are trying to solve is: Given n image points $\{A_i\}$ and $\{B_i\}$ $i = 1, 2, \dots, N$ as depicted in Figure 4, establish the correspondences, i.e., determine the permutation σ .

3 Method

Since we have assumed that the translation vector is small compared to the distance of the object, we will approximate equation(1) above by

$$z_{\sigma(i)} B_{\sigma(i)} = \mathbf{R} z_i A_i \quad i = 1, \dots, N \quad (3)$$

Let \hat{A}_i and \hat{B}_i be a unit vector of A_i and B_i , respectively. The above can then be written as

$$\hat{B}_{\sigma(i)} = \mathbf{R} \hat{A}_i \quad i = 1, \dots, N \quad (4)$$

Thus,

$$\hat{B}_{\sigma(i)} \hat{B}_{\sigma(i)}^t = \mathbf{R} \hat{A}_i \hat{A}_i^t \mathbf{R}^t \quad i = 1, \dots, N \quad (5)$$

Summing the above over i , we get

$$\sum_{i=1}^N \hat{B}_{\sigma(i)} \hat{B}_{\sigma(i)}^t = \sum_{i=1}^N \mathbf{R} \hat{A}_i \hat{A}_i^t \mathbf{R}^t \quad (6)$$

Since the summation is over all points, the order in which they appear makes no difference. The σ is thus irrelevant and can be dropped in the formula. Consequently,

$$\mathbf{R} Q_1 \mathbf{R}^t = Q_2 \quad (7)$$

where $Q_2 = \sum_{i=1}^N \hat{B}_i \hat{B}_i^t$, and $Q_1 = \sum_{i=1}^N \hat{A}_i \hat{A}_i^t$. This formula implies that Q_1 and Q_2 should have the same eigenvalues. In practice, they would not have the same set of eigenvalues for the Translation is ignored. Let $Q_1 = V D_1 V^t$ and $Q_2 = U D_2 U^t$, then $\mathbf{R} = U V^t$.

¹The difference between D_1 and D_2 are ignored. These can be used to determine how accurate our method will be.

Consider A_i . Our goal is to find $\sigma(i)$, the permutation that will map this into the corresponding point in $\{B_i\}$. To do this, consider the following. Suppose we remove from both images a pair of corresponding points. Then, the rotation matrix as computed by equation (6) will not change. Conversely, if we remove a point from $\{A_i\}$, and remove some other point from $\{B_i\}$, then the computed rotation matrix would differ from the original one. This, then, gives us an elegant method of solving the correspondence problem. Using all the points, compute the rotation matrix \mathbf{R} . Let us suppose that we wish to find the point j in the second frame which corresponds to point l in the first frame. From the previous argument, it follows *ipso facto* that when l is dropped from the first frame and j from the second, the rotation matrix thus computed will be very close to the original one. Let \mathbf{R}'_k denote the rotation matrix when points l and k are dropped from the first and second frames respectively. The corresponding point j then satisfies the relation

$$diff(\mathbf{R} - \mathbf{R}'_j) = \min_k \{diff(\mathbf{R} - \mathbf{R}'_k)\}$$

The function *diff* mentioned above can be any geometrically meaningful measure of the difference in the rotation matrices. We chose to compute the rotation angle from the rotation matrix, and use the difference of rotation angles as our difference measure. This angle can be trivially computed from the matrix as

$$\theta = \frac{180}{\pi} \times (\arccos(0.5 \times trace(\mathbf{R})))$$

We may mention here that the algorithm returns four possible angles of rotation. While only one of the angles represents the actual rotation, all of them satisfy the aforementioned property with regard to removing point pairs. As such they are equally valid for the purpose of obtaining correspondences. One may however note that if some *a priori* estimate is available for the angle of rotation, the consistency property can then be checked with regards to this angle, and the results obtained are somewhat superior.

A naive way to obtain correspondences then is that for each point in the first image, compute the *diff* measure by successively dropping all N points in the second frame, and choose the one which minimises this measure. Since there are N points, and for each of them, N rotation matrices will be

computed, the method will end up doing the computations required to obtain \mathbf{R} , $O(N^2)$ times. However, the constant involved in this can be lowered by observing that once a point in the second frame is known to correspond in the first frame, it need not be used for subsequent comparisons. Thus when computing the i^{th} point's correspondence, only $N - i$ points from the second frame have to be considered. The computations for \mathbf{R} need be done only $O(\frac{1}{2}N^2)$ times.

4 Exploiting Parallelism

In order to obtain the correspondences, we have to, for each point, compute N different rotation matrices, which are obtained by dropping different points from the second frame. These computations do not interact with each other in any way, and can be done independently. This is a classic case of *Data Partitioning* or *Homogeneous Multitasking*. It involves identical tasks, operating on different parts of the data in parallel. Exploiting this parallelism can speed up this process considerably. In fact, given $O(\frac{N}{\log(N)})$ processors, the time required is $O(N \log(N))$. With $O(\frac{N^2}{\log(N)})$ processors, the time can be further reduced to $O(\log(N))$ [25].

5 Experiments on Synthetic Data

In order to verify the performance of the proposed algorithm and test the validity of our assumption, we have carried out extensive computer simulations. The first set of these were carried out using synthesised data. This enabled us to test the algorithm without bothering about the problem of which points to use from a real image. The following parameters were used in the simulations :

- Field of view was taken to be $120^\circ \times 120^\circ$
- Screen size was taken as 512×512
- The focal length was assumed to be 1 .

We started with a set of points expressed in terms of their x,y and z coordinates. These were then subjected to a rotation and translation. A

perspective projection model was then used to obtain the first image frame. The points of the first image frame were then subjected to another rotation, translation and perspective projection to obtain the second image frame. The image frames so obtained contain the three spatial coordinates of the various points. These image frames are then converted to the corresponding screens, by computing the screen coordinates of the points in question on a 512×512 screen.

The two sets of screen coordinates thus obtained are used to compute the rotation matrix, and from this, the angle of rotation is obtained. Then, points that do not correspond to each other are dropped, and the rotation angles are computed. This is followed by dropping corresponding points from the screens and once again computing the rotation angles. This process was done for differing angles of rotation as well as translations. The results obtained from these are presented in Figures 5 through 7.

Figure 5 illustrates the graph of the error in the computed angle compared to the actual angle. We note that for most cases, the computed angles are within acceptable error limits. The only exceptions to this are the results from data which combine small rotation angles with large translations. This condition is, of course, contrary to our assumption of small translation.

Figure 6 shows graphs of the maximum error when non corresponding points are removed superposed over graphs of the maximum error when corresponding points are removed. The values for non-corresponding points are obtained by removing the first point from the first frame, and second through last points from the second frame. Each graph is for a given translation, and has the angle of rotation as a dependent variable. The graphs clearly demonstrate that there is marked difference in the errors in the two cases, with the error in the former case being in general overwhelmingly large compared to the error in the later case. Again, it may be noted that the smaller the translation, the better the distinction. This clearly supports our contention that correspondence can be established by removing point pairs from the two image frames.

Tables 1 and 2 respectively show the number of points correctly corresponded both when estimate of rotation angle is not *a-priori* available and vice versa. In the both the above cases, while small translations (Less than 20% of the object - image plane distance) gave better results, we note that the performance of the algorithm deteriorated only for translations larger than 50% of the distance of the object from the image plane.

While under idealised conditions, exact points are made available to the algorithm, this cannot be expected in real image data. In order to evaluate our algorithms performance for noisy data, random noise was added to the points of the second image frame. This noise was obtained using the unix random number generator, and the values were transformed to lie between 0 and some specified maximum value. The noisy frame was then used for subsequent computations. As the results given in Figure 7 demonstrate, the proposed algorithm is robust under noisy conditions. We found that the algorithm was able to handle noise in the range 0 - 50 pixels, which is about 10% of 512 pixels, without being very adversely affected, and serious deterioration of results was obtained only when the noise became about 20% in terms of pixels.

6 Experiments on real images

While the simulation results from the synthesised data validate our claims, we also carried out experiments on real images. Since this work does not address the problem of extracting the tokens from the image, we assume that points from the two image frames are made available to the algorithm. Accordingly, points were extracted from several real image sequences, and our algorithm was run on these. The results were in keeping with the predictions. The algorithm was able to successfully establish correspondences between points in the first and second image frames.

We may add here that the images we used all had very low angles of rotation, all within 10° . For instance, consider the image sequence presented in Figure 8. This involves a camera mounted on a robotic arm. The computed angle of rotation here was 3.9° . We noticed that when we removed corresponding points, our errors were within 1° . When we removed non corresponding points, the errors were in the range $10^\circ - 30^\circ$, and generally tended to be around 20° . However, one or two points caused errors that were within a degree or two of the computed rotation angle, leading to their being incorrectly corresponded.

Similar results were obtained for the outdoors image sequence presented in Figure 3. It should be noted that for the robotic arm images, the focal length of the camera was not available to us, and we used an educated guess to estimate it.

7 Discussion

The feature based schemes for image analysis have been forced to assume an Oracle like subsystem to solve the correspondence problem. This had led to major criticism of such schemes. In this paper, we have identified an assumption, and based on it, presented a robust and efficient algorithm which will solve the correspondence problem from a purely geometric perspective. This will enable feature based schemes to get rid of their assumptions about the correspondence problem .

The argument that we have made about corresponding points can also be extended to groups of points. Since a portion of an image can be viewed as a group of points, it follows that if we remove corresponding portions of two image frames, we should get consistent rotations, and that if different portions are removed, the rotations obtained will not be consistent. We are currently working on an algorithm to implement this idea.

The proposed algorithm can also be extended to obtain approximations to the translation vector \mathbf{T} and the depths, z_i and z_i' . The extended algorithm will not only obtain the correspondences, but also estimate the complete motion parameters. This is illustrated in Figure 9, where the squares mark the original frame, the deltas mark the points in the second frame, and the stars denote predictions about the points in the second frame made by the extended algorithm. This extension shall be presented in a future publication.

8 References

1. S. Ullman, *The Interpretation of Visual Motion* , Cambridge, MA, MIT Press, 1979.
2. J.W. Roach and J.K. Aggarwal, "Determining the Movement of Objects from a Sequence of Images", *IEEE Trans. Pattern Anal. Machine Intelligence* , Vol. PAMI-2, Nov. 1979.
3. H.C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from Two Projections", *Nature*, 293, 1981.
4. R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Sur-

- faces", *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol PAMI-6, No. 1, Jan. 1984.
5. H.H. Nagel, "Image Sequences - Ten (octal) years - from phenomenology towards a theoretical foundation ", in *Proc. Eighth International Conference on Pattern Recognition*, Paris, France (1986) 1174-1185
 6. J.K. Aggarwal, "Motion and time-varying imagery-An overview. Proc. of IEEE Workshop on Motion: Representation and Analysis, South Carolina, 1986.
 7. J.K. Aggarwal and A. Mitiche "Structure and Motion from Images: facts and fiction." Proc. of IEEE Workshop on Motion: Representation and Analysis, South Carolina, 1986.
 8. Aloimonos J. "Perception of Structure from Motion", Conference of Computer Vision and Pattern Recognition, IEEE Computer Society, June 25-27, Miami, Florid, 1986.
 9. G.S. Young and Rama Chellappa, *3-D Motion Estimation Using a Sequence of Noisy Stereo Images*, CVPR, Michigan Ann Arbor, 1988.
 10. J. Weng et.al "3D motion estimation, understanding, and prediction from noisy image sequence". IEEE PAMI Vol-PAMI-9, pp. 370-389, May 1987.
 11. O.D. Faugeras and S. Maybank, "Motion from point matches: multiplicity of solutions" *Proceedings IEEE Workshop on Motion*, 1989.
 12. B.K.P. Horn, "Relative Orientation" *International Journal of Computer Vision*, Vol. 4 1990.
 13. B.K.P. Horn, "Recovering Baseline and Orientation from Essential Matrix" *MIT AI Memo*, 1990.
 14. W. Burger and B. Bhanu, "Estimating 3-D motion from perspective Image Sequences" *IEEE Trans. Pattern Recognition and Machine Intelligence*, Vol. PAMI-12, No. 11, 1990
 15. Baker (1981), "Depth from Edge and Intensity Based Stereo" ,Ph.D thesis, AIM-347, Computer Science Department, Stanford University.

16. Leclerc, Y.G. and S.W Zucker (1987), "The local Structure of Image Discontinuities in One Dimension", *IEEE Trans. PAMI* 9, 341-355.
17. Nevatia R. (1976), "Depth from Camera Motion in a Real World Scene", *Computer Vision, Graphics and Image Processing* 9, 203-214.
18. Williams , T.D. (1980), "Depth from Camera Motion in a Real World Scene", *IEEE Trans. PAMI* 2, 511-516.
19. Grimson, W.E.L , "From Image to surface" , MIT Press, Cambridge, MA.
20. Salari,V. and Sethi, I.K. , "Feature point correspondence in presence of occlusion" , *IEEE Trans. PAMI* Vol. 12, 87-91
21. Jain, R. and Sethi, I.K., " Finding trajectories of feature points in a monocular image sequence " , *IEEE Trans. PAMI*, Vol. 9, 56-73
22. C.H. Lee and T.S. Huang, "Finding Point Correspondences and Determining Motion of a Rigid Object from Two Weak Perspective Views" *Computer Vision, Graphics, and Image processing* 52, 309-327 (1990).
23. Jain, R and Binford, T.O , " Dialogue: Ignorance, Myopia and Naivette in Computer vision Systems", *Computer Vision, Graphics and Image Processing*, 53, 112-117 (1991).
24. Huang, T.S. "Reply: Computer Vision needs more experiments and applications" , *Ibid*, 125-126 .
25. Gibbons,A. and Rytter, W, "Efficient Parallel Algorithms", *Cambridge University Press* .

Translation	No. of Points	Correctly corresponded
0 0 0	7	6
4 4 4	7	6
12 12 12	7	3

Table 1: Rotation angle : 70 degrees

Translation	No. of Points	Correctly corresponded
0 0 0	7	7
4 4 4	7	6
12 12 12	7	4

Table 2: Rotation angle : 70 degrees

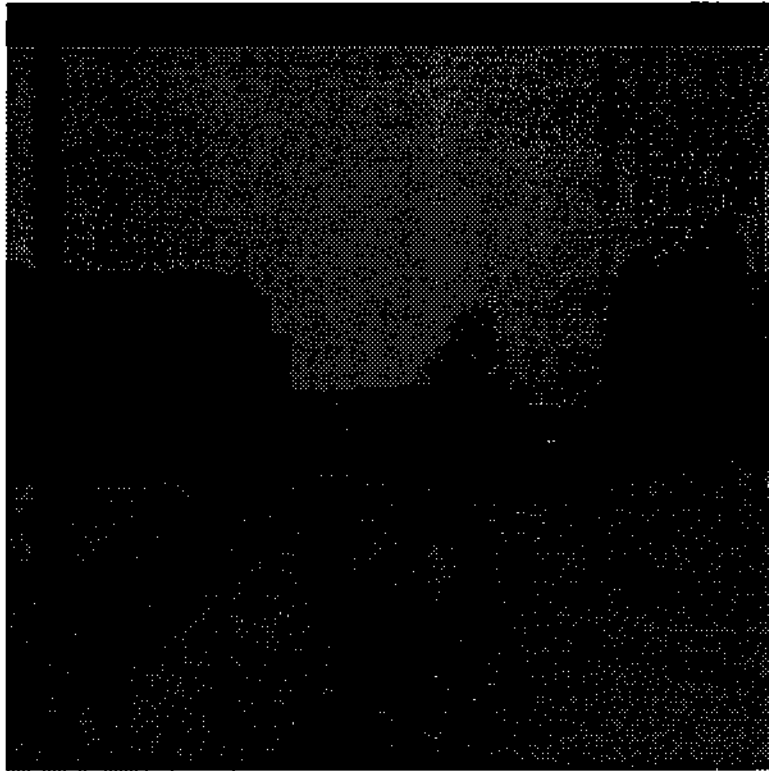


Figure 1a

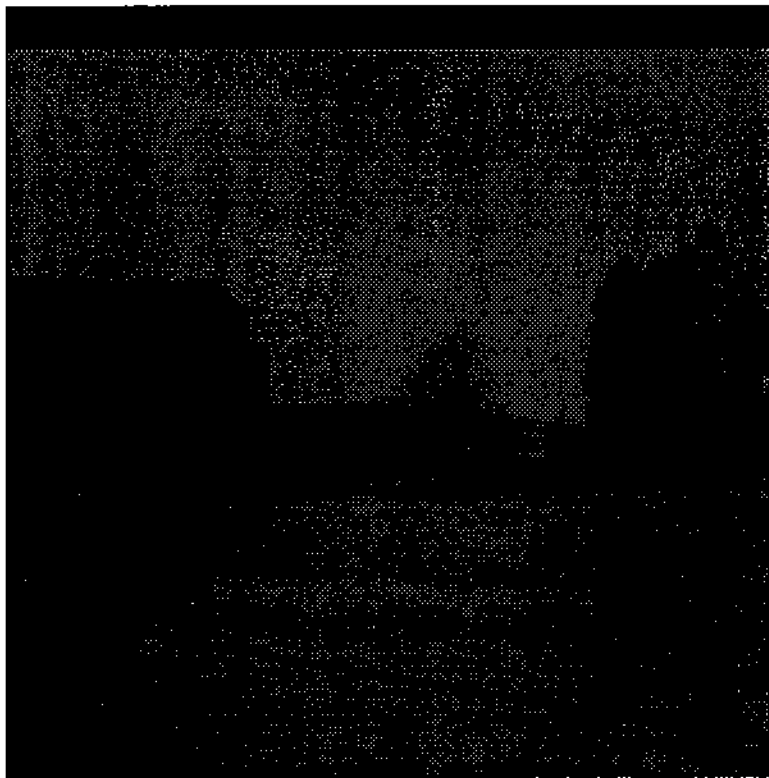


Figure 1b

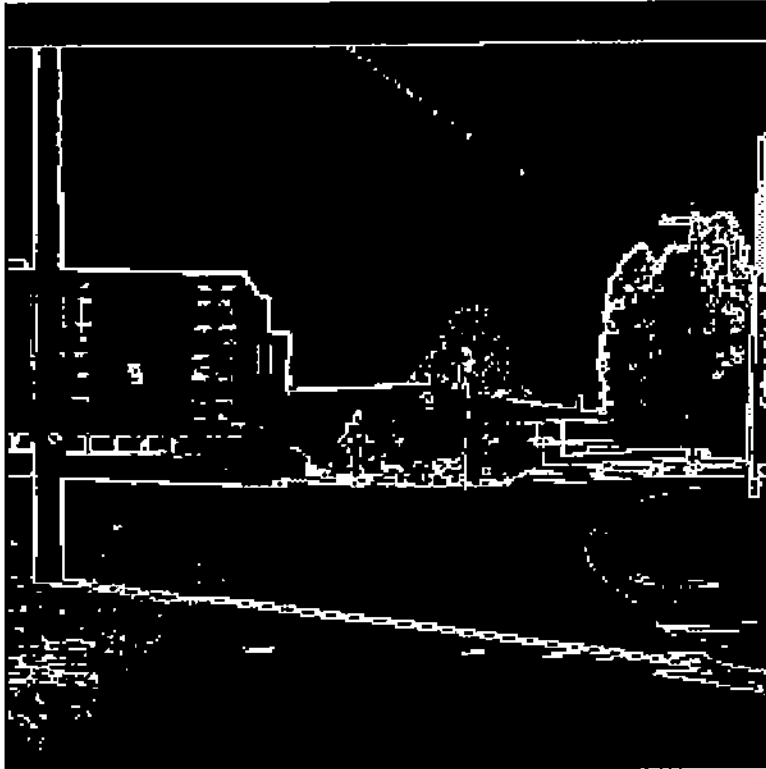


Figure 2a



Figure 2b



Figure 3a

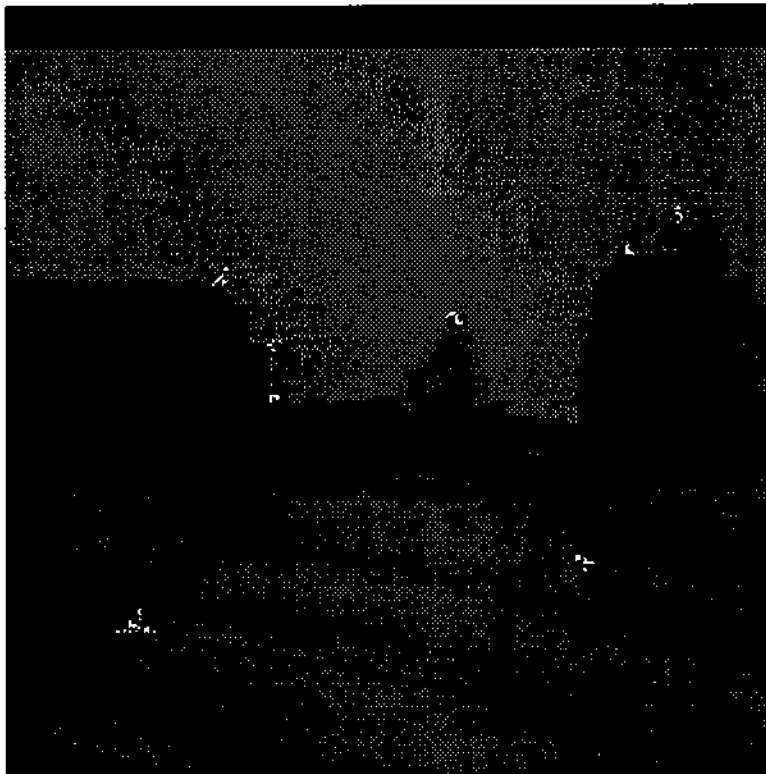
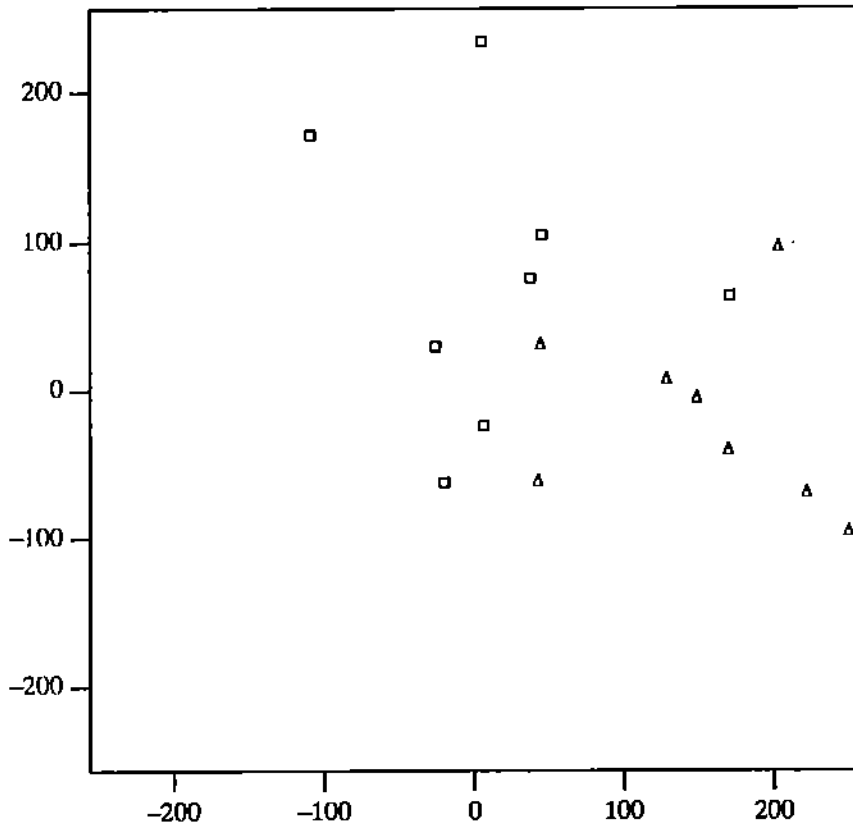
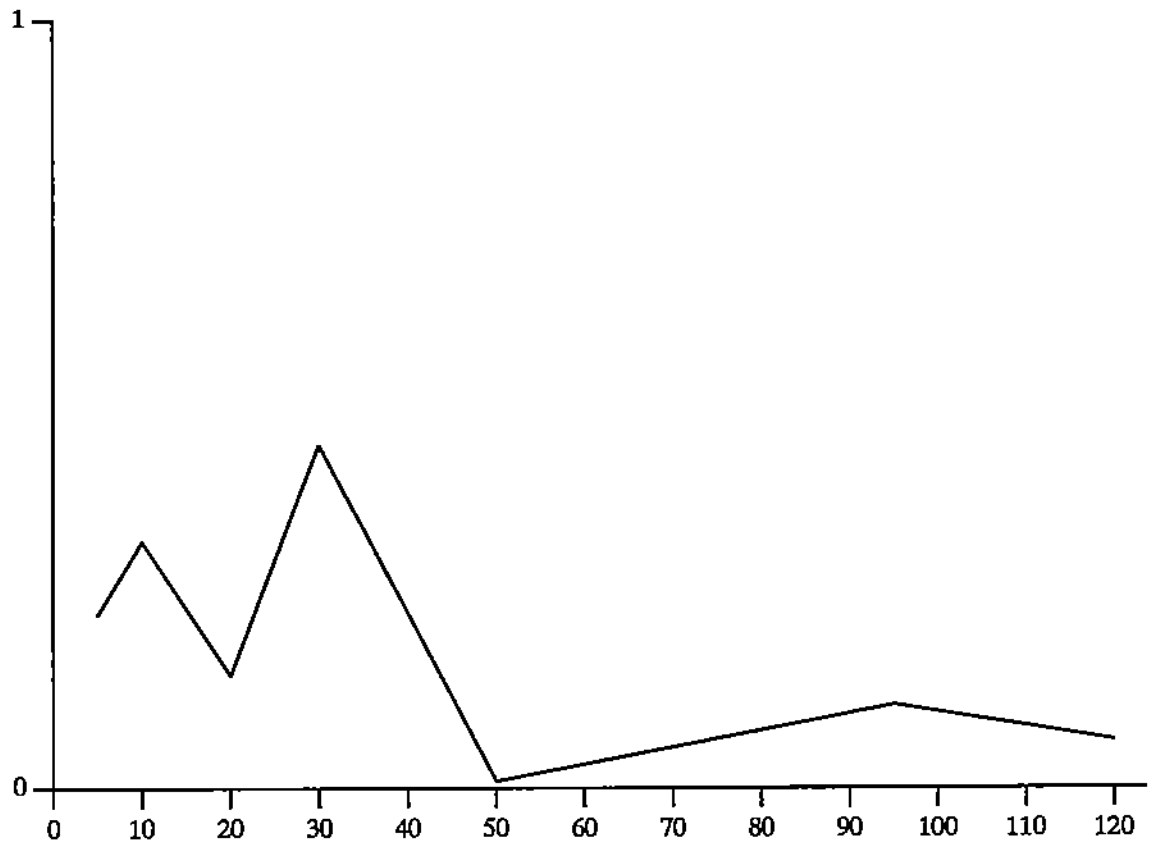


Figure 3b

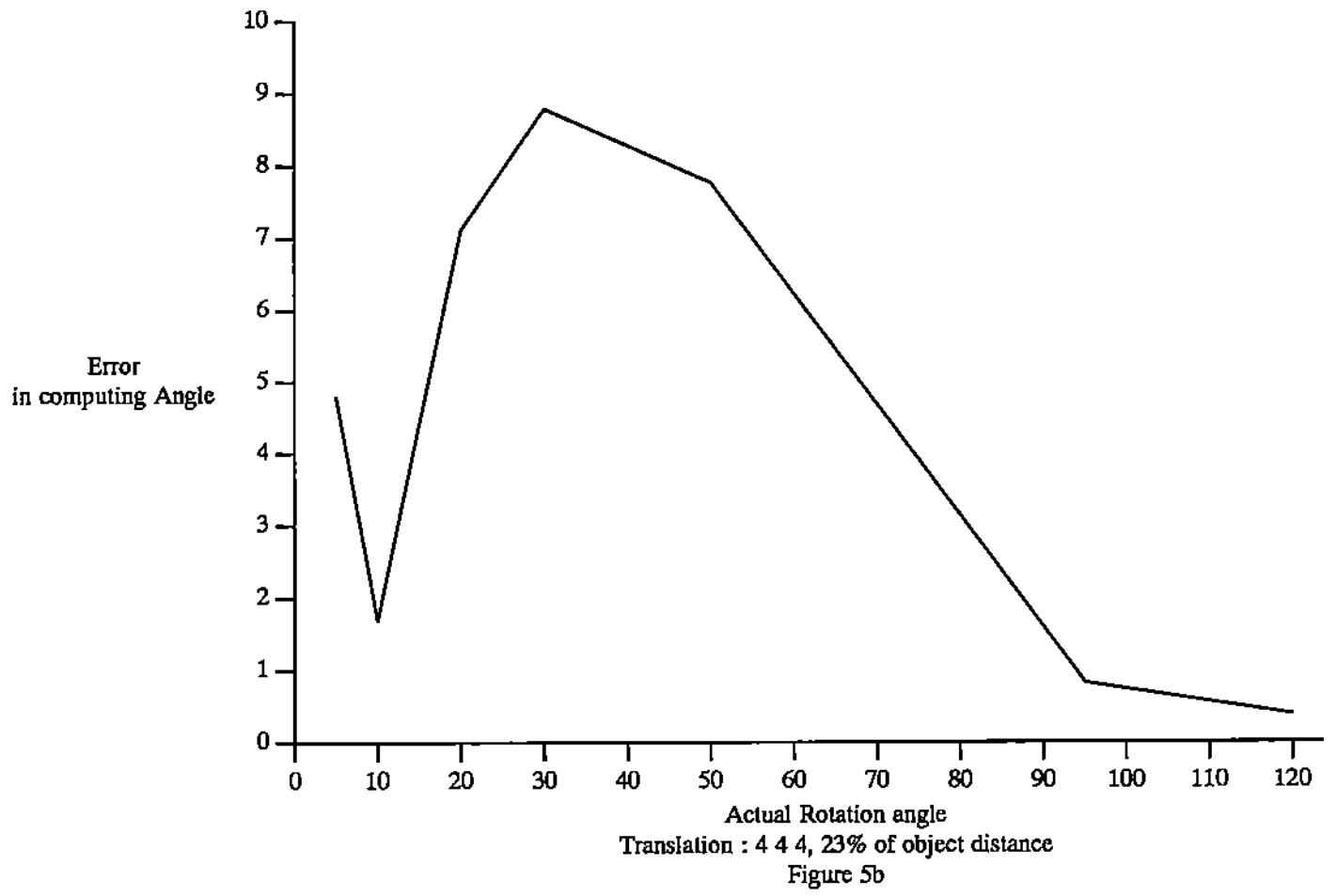


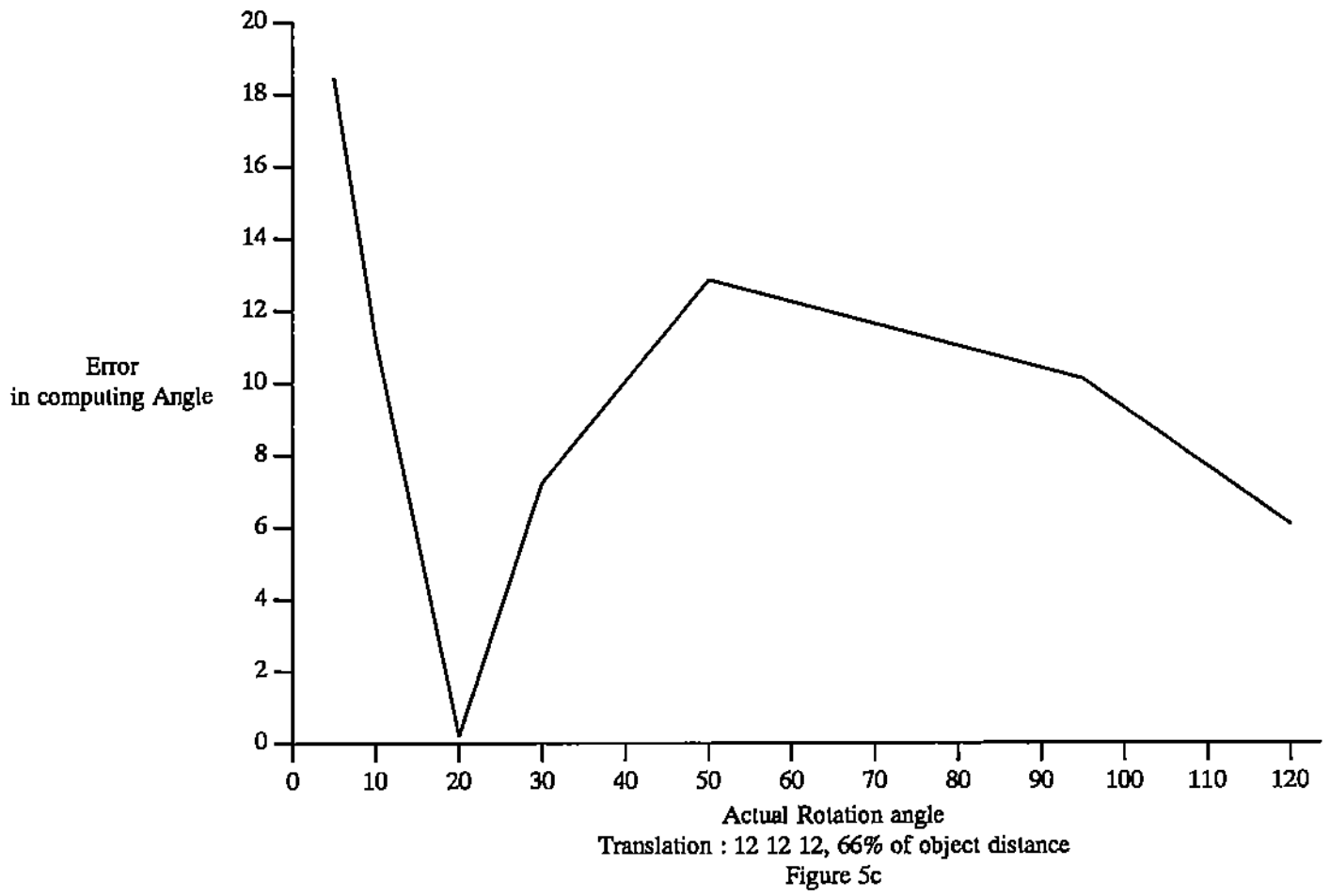
Squares represent points in the first frame,triangles in the second
Figure 4

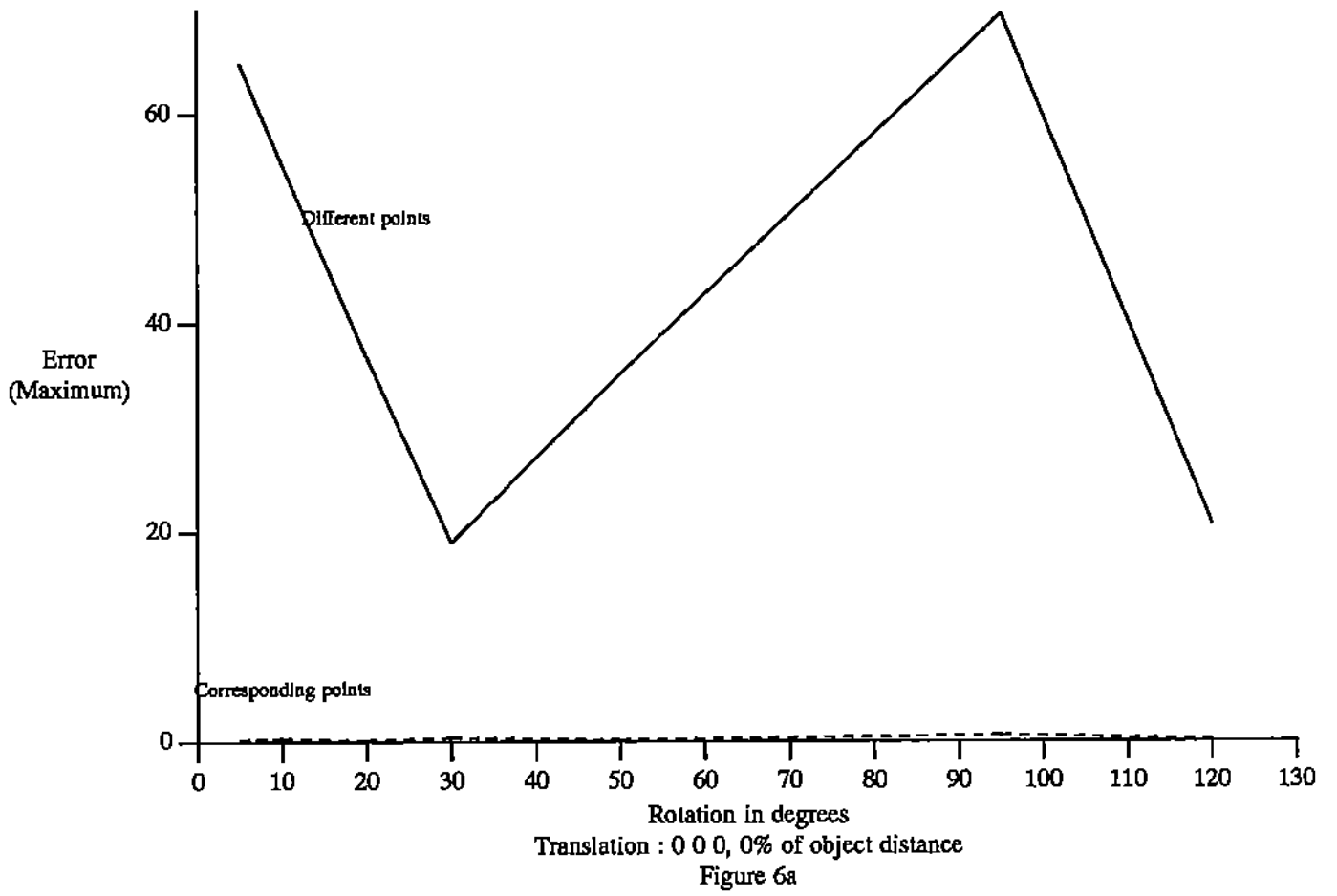
Error
in computing Angle

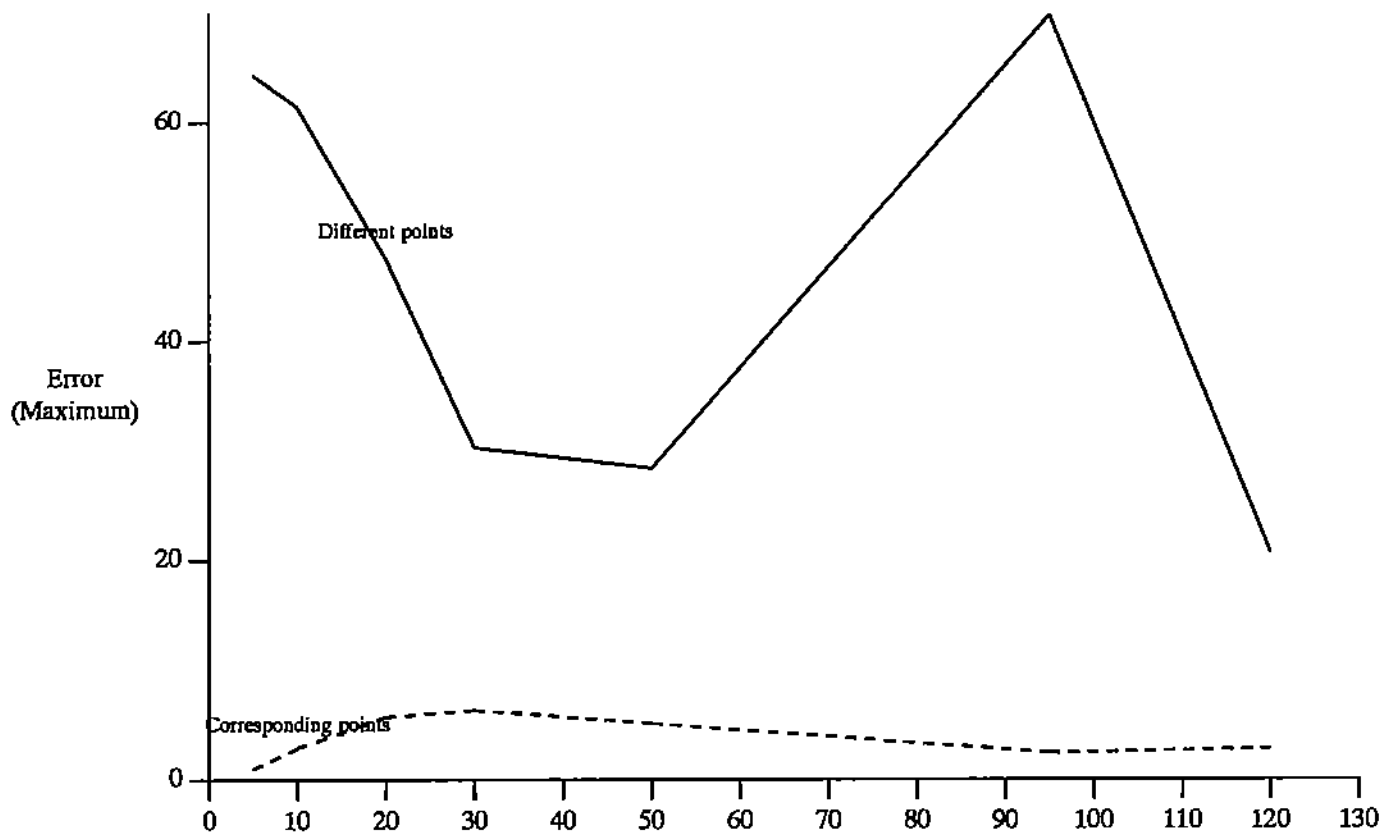


Actual Rotation angle
Translation : 0 0 0, 0% of object distance
Figure 5a

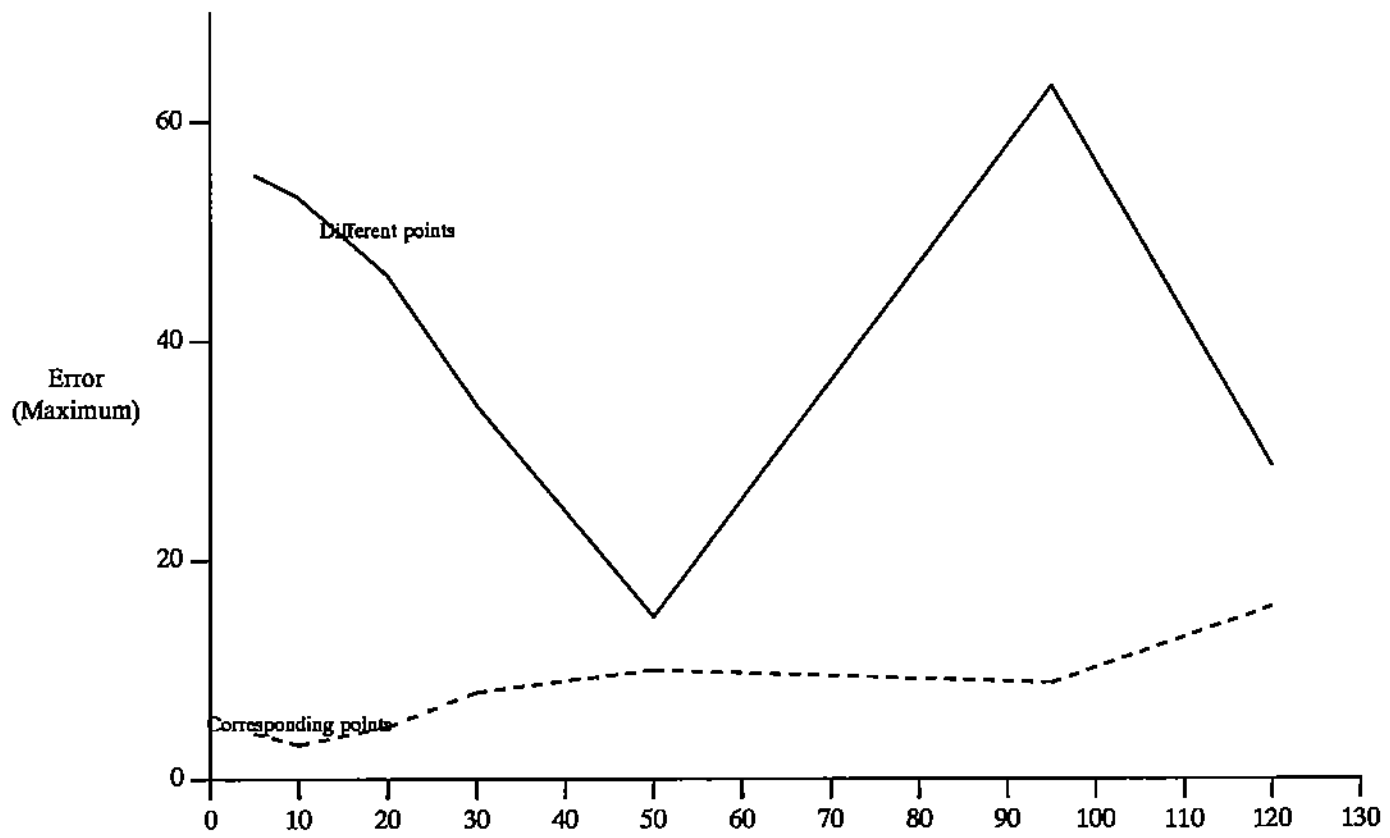




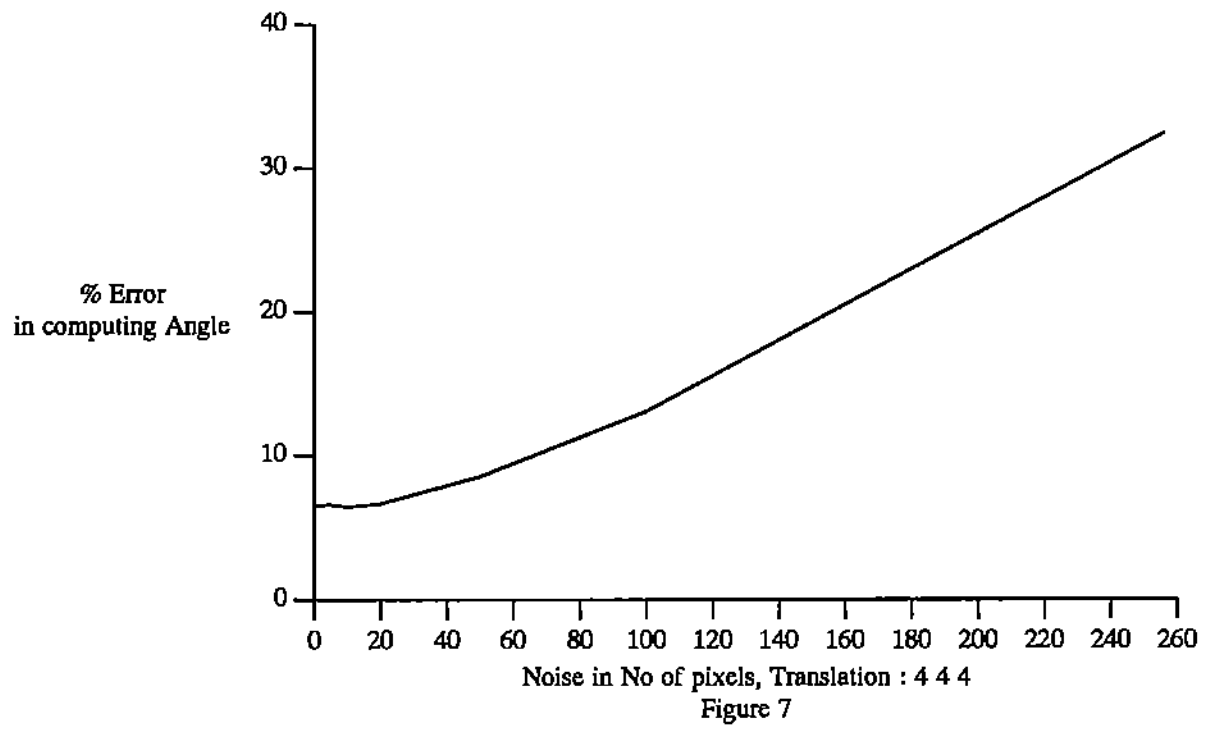




Rotation in degrees
Translation : 4 4 4, 23% of object distance
Figure 6b



Rotation in degrees
Translation : 12 12 12, 66% of object distance
Figure 6c



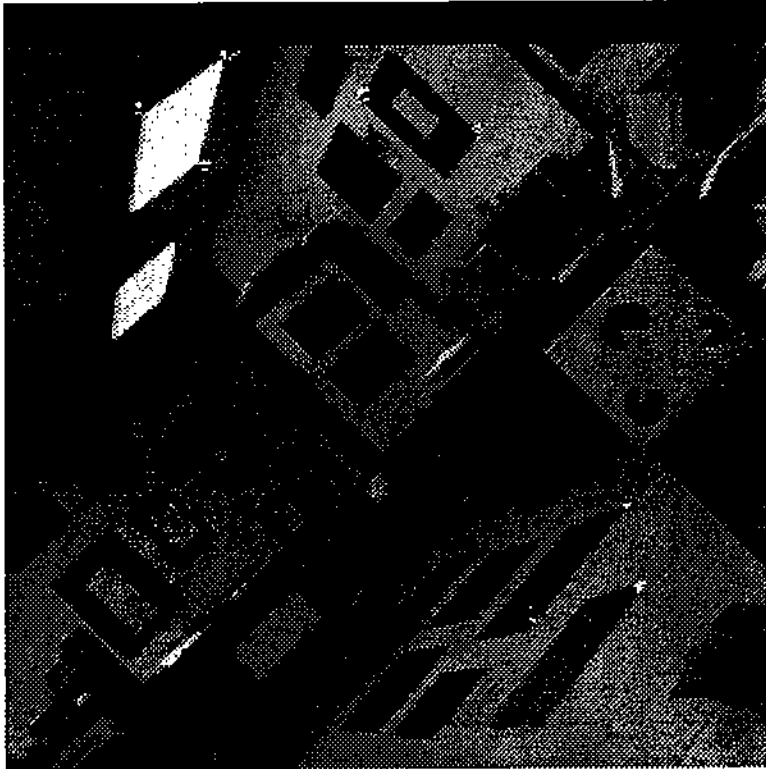
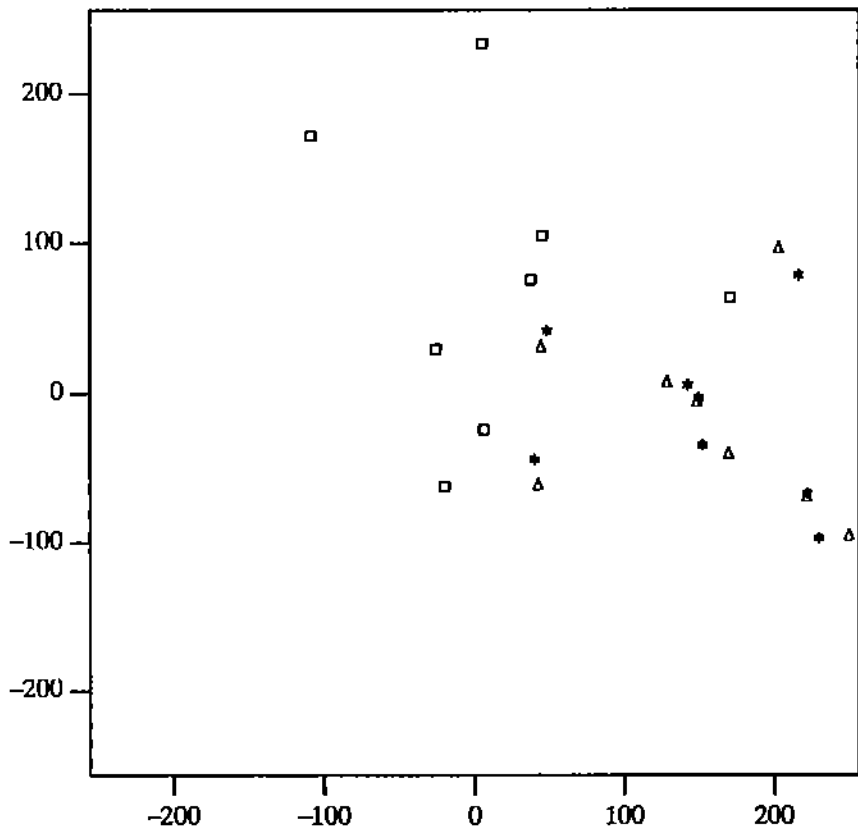


Figure 8a



Figure 8b



Squares represents points in the first frame, triangles in the second
 Stars are predicted positions of the points in the second frame

Figure 9