

CLOSED-FORM SOLUTION + MAXIMUM LIKELIHOOD: A ROBUST APPROACH TO MOTION AND STRUCTURE ESTIMATION

JUYANG WENG, NARENDRA AHUJA, THOMAS S. HUANG

Coordinated Science Laboratory
University of Illinois Urbana, IL 61801

Abstract

This paper presents a robust approach to estimating motion and structure from image sequences. The approach consists of two steps. The first step is estimating the motion parameters using a robust linear algorithm that gives a closed-form solution for motion parameters and scene structure. The second step is improving the results from the linear algorithm using maximum likelihood estimation. In other words, the motion parameters and structure of the scene are such that the conditional probability for the observed data, given the motion parameters and structure, reaches the maximum. The maximum likelihood estimates are computed through an iterative algorithm. Since preliminary estimates are available, the algorithm reach the global minimum quickly and reliably.

An algorithm using point correspondences from monocular images is discussed in detail and experimented with. An algorithm using line correspondences is briefly discussed. The simulations show that maximum likelihood estimation achieves remarkable improvement over the preliminary estimates given by the linear algorithm. The algorithm is also tested on images of real scenes from automatically computed displacement field.

The proposed approach is independent of the exact tokens used to establish correspondences, e.g. displacement flow, optical flow or discrete features. Two or more types of tokens may be used, for monocular or binocular images.

1. INTRODUCTION

From correspondences (displacement field, optical flow, feature correspondences, etc) a variety of algorithms have been proposed to solve for the motion and structure parameters of the scene. They can be classified as nonlinear algorithms and linear algorithms.

A nonlinear algorithm solves a set of nonlinear equations directly using iterative procedures. A linear algorithm, on the other hand, gives closed form solutions. It mainly solves linear equations and some nonlinear equations in special forms. Typically intermediate parameters are solved based on a set of linear equations. The motion parameters are then solved from those intermediate parameters based on some nonlinear equations. One of the advantages of linear algorithms over nonlinear ones is that the structure of the solution of a set of linear equations is clear. Therefore a correct solution is guaranteed if the data are free of noise. Another important advantage is that the uniqueness of the solutions can be investigated. Also a linear algorithm is faster. However, in the presence of noise the solution from a linear algorithm may not be an optimal one.

We present here a two-step approach that employs both types of algorithms to make use of their advantages. The next section introduces the two-step approach. Section 3 discusses the maximum likelihood estimation of the motion and structure parameters. Details of implementation are discussed in Section 4. Simulation results are

presented in Section 5. Section 6 presents concluding remarks.

2. A TWO-STEP APPROACH

The two-step approach is as follows. First, a linear algorithm is applied. Then the solution of the linear algorithm is used in the second step as an initial guess for an iterative algorithm which improves on the initial guess based on an objective function. This two-step approach has the following advantages: 1) A solution is generally guaranteed by linear algorithm. 2) The approach allows flexible design of the objective function since a good initial guess is available. 3) The approach yields a reliable solution due to optimization. 4) Fast computation. A linear algorithm is fast, and generally a nonlinear algorithm is slow. Since the linear algorithm provides good initial guess, the time taken by the nonlinear algorithm to reach a solution is greatly reduced.

For the following discussion, we will consider monocular images with point correspondences. The use of line correspondences will be discussed briefly. Other type of features could be treated in a similar way. For the first step, we will use the linear algorithms for point [Weng87a] and line correspondences [Weng88a] developed by Weng et. al..

We have a sequence of images taken at different camera positions and orientations relative to the scene. The objective is to estimate the relative motion between the camera and the scene. To be specific, we consider two images. The geometrical setup with image coordinate systems and world coordinate systems is shown in Fig. 1. Without loss of generality, assume the focal length of the camera is unity. Let a feature point be located at $\mathbf{x}=(x, y, z)^t$ in world coordinate system. The image vector of \mathbf{x} in the first image is defined by

$$\mathbf{u} = (u, v, 1)^t = (x/z, y/z, 1)^t \quad (2.1)$$

in its image coordinate system. The image coordinates of the point are (u, v) which are the perspective projections of the point \mathbf{x} onto the image plane. After the first image is taken, the camera undergoes a motion and takes the second image as shown in Fig. 1. The motion of the camera is first a translation represented by a vector \mathbf{T} followed by a rotation represented by a rotation matrix R . Any point \mathbf{p} on the camera is moved to \mathbf{p}' . They are therefore related by

$$\mathbf{p}' = R(\mathbf{p} + \mathbf{T}) \quad (2.2)$$

The transformation from \mathbf{p} to \mathbf{p}' corresponds to the motion of the camera. In the coordinate system of the second image the feature point has a new position vector $\mathbf{x}'=(x', y', z')^t$ whose image vector is

$$\mathbf{u}' = (u', v', 1)^t = (x'/z', y'/z', 1)^t \quad (2.3)$$

(see Fig. 1). The relation between \mathbf{x} and \mathbf{x}' is related by

$$\mathbf{x}' = R^t \mathbf{x} - \mathbf{T} \quad (2.4)$$

Let $\mathbf{T} \neq \mathbf{0}$, $\alpha = \|\mathbf{T}\|^{-1}$, $\tilde{\mathbf{x}} = \alpha \mathbf{x}$ and $\tilde{\mathbf{x}}' = \alpha \mathbf{x}'$. α is a positive global scale factor that cannot be determined from monocular images

[Weng87a]. Given the correspondences for the image vectors at the two time instants, \mathbf{u} and \mathbf{u}' , the motion analysis algorithm [Weng87a] solves for the rotation matrix R , the unit translation vector (direction) $\hat{\mathbf{T}} = \alpha \mathbf{T}$. For each point, the algorithm solves for $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}'}$. α can be determined only if additional information is available. (For example, if the absolute distance between a pair of feature points is known, α can be determined).

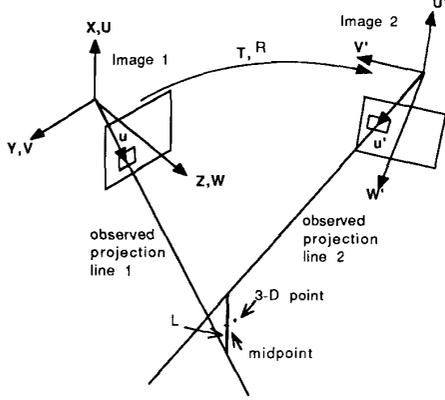


Fig. 1 A moving camera takes two images

Notice that $\tilde{\mathbf{x}} = \alpha \mathbf{z}$ and $\tilde{\mathbf{x}'} = \alpha \mathbf{z}'$. Letting $\tilde{\mathbf{z}} = \alpha \mathbf{z}$, $\tilde{\mathbf{z}'} = \alpha \mathbf{z}'$ denote relative depths, (2.4) yields

$$\tilde{\mathbf{z}'} \mathbf{u}' = R' \tilde{\mathbf{z}} \mathbf{u} - \hat{\mathbf{T}} \quad (2.5)$$

After solving R and $\hat{\mathbf{T}}$, for each image point the linear algorithm [Weng87a] solves for $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{z}'}$ using linear least squares based on (2.5). We get $\tilde{\mathbf{x}} = \tilde{\mathbf{z}} \mathbf{u}$ and $\tilde{\mathbf{x}'} = \tilde{\mathbf{z}'} \mathbf{u}'$. However, with noise such $\tilde{\mathbf{x}}$ in the first image coordinate system and $\tilde{\mathbf{x}'}$ in the second image coordinate system generally do not correspond to the same point. For the linear algorithm, we use a simple solution to this problem. Each point in the final estimated structure is the midpoint of the corresponding $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}'}$ as shown in Fig. 1. Therefore the image vectors are modified according to the estimated 3-D position of the feature points.

The solution given by the linear algorithm is used as an initial guess for the nonlinear algorithm for the second step which improves this initial guess based on the desired optimality. The next section discusses the optimization based on maximum likelihood estimation.

3. MAXIMUM LIKELIHOOD ESTIMATION

A more robust way for estimating the parameters is using error (or noise) distribution. In reality, the feature locations are the results of feature detectors whose accuracy is influenced by a variety of factors including lighting condition, scene structure, imaging device calibration and image resolution. So, the observed 2-D image vectors \mathbf{u}_i for image 1 and \mathbf{u}'_i for image 2 are noise corrupted versions of the true image vectors. Therefore $(\mathbf{u}_i, \mathbf{u}'_i)$ is the observed value of a pair of random vector (U_i, U'_i) . With n point correspondences over two time instants what we observed is the sequence of image vector pairs

$$\mathbf{u} \triangleq (\mathbf{u}_1, \mathbf{u}'_1, \mathbf{u}_2, \mathbf{u}'_2, \dots, \mathbf{u}_n, \mathbf{u}'_n) \quad (3.1)$$

of a sequence of random vector pairs

$$\mathbf{U} \triangleq (U_1, U'_1, U_2, U'_2, \dots, U_n, U'_n) \quad (3.2)$$

We need to estimate the motion parameters M and the 3-D positions of the feature points (scene structure)

$$\mathbf{X} \triangleq (\mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}_2, \mathbf{x}'_2, \dots, \mathbf{x}_n, \mathbf{x}'_n) \quad (3.3)$$

Let the conditional density of \mathbf{U} given that $M = m$ and $X = x$ be $f_{U|M,X}(\mathbf{u} | m, x)$. The maximum likelihood estimates of motion

parameters, m^* , and scene structure x^* are such that the conditional density $f_{U|M,X}(\mathbf{u} | m, x)$ reaches maximum. That is,

$$f_{U|M,X}(\mathbf{u} | m^*, x^*) \geq f_{U|M,X}(\mathbf{u} | m, x) \quad (3.4)$$

holds for all possible motion parameters m and scene structure x .

To find the maximum likelihood estimate, we need the conditional density $f_{U|M,X}(\mathbf{u} | m, x)$.

Model 1: Gaussian Distribution

The Gaussian distribution is a commonly used for modeling noise. Intuitively, if the errors come from many sources and are influenced by many factors, the distribution is roughly Gaussian by the central limit theorem. Let each image coordinate of the observed projection has an independent additive zero mean Gaussian noise. For simplicity we assume the conditional distributions, given motion parameters and scene structure, are independent between different points and the variances are the same. Such an assumption is reasonable when the image resolution is not very low. Intuitively, the error value of the image coordinates of a point does not closely related to the error value of other points if the sample grid is dense. In reality such a dependency is not influential and can be neglected. Then the conditional density of the observed projection given the motion parameters M and the structure of scene X may be written as

$$f_{U|M,X}(\mathbf{u} | m, x) = \prod_{i=1}^n f_i(\mathbf{u}_i) f'_i(\mathbf{u}'_i) \quad (3.5)$$

where \mathbf{u}_i is the observed projection of the 3-D feature point i in the first image, \mathbf{u}'_i is that of i in the second image, f_i is the Gaussian density with mean equal to the exact projected location of the given feature point, \mathbf{u}_{0i} in the first image. f'_i is analogous to f_i , for the second image. That is

$$f_i(\mathbf{u}_i) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\mathbf{u}_i - \mathbf{u}_{0i}\|^2}{\sigma^2}\right) \quad (3.6)$$

$$f'_i(\mathbf{u}'_i) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\mathbf{u}'_i - \mathbf{u}'_{0i}\|^2}{\sigma^2}\right) \quad (3.7)$$

For each image, the line passing through the observed image point and the focal point is called the observed projection line (Fig. 1). Fig. 2 shows the value of $f_i(\mathbf{u}_i) f'_i(\mathbf{u}'_i)$ in $X-Z$ plane as intensity image. It shows the distribution of the 3-D points given its images.

The maximum likelihood estimates of motion parameters and the structure of the scene are such that the conditional density is the maximum. Maximizing the left hand side of (3.5) is equivalent to maximizing the logarithm of it.

$$\ln f_{U|M}(\mathbf{u} | m) = - \sum_{i=1}^n (\|\mathbf{u}_i - \mathbf{u}_{0i}\|^2 + \|\mathbf{u}'_i - \mathbf{u}'_{0i}\|^2) / \sigma^2 - n \ln(2\pi\sigma^2) \quad (3.8)$$

Maximizing the left-hand side of (3.8) is equivalent to minimizing

$$\sum_{i=1}^n (\|\mathbf{u}_i - \mathbf{u}_{0i}\|^2 + \|\mathbf{u}'_i - \mathbf{u}'_{0i}\|^2) \quad (3.9)$$

which is just the sum of disparities between the observed projection and the true projections. It is clear that the maximum likelihood estimation results from minimization of (3.9) by adjusting both motion parameters and point configuration at the same time. This is an intuitive explanation of the estimates. Even without the assumption of the Gaussian noise distribution, the objective function of (3.9) is desirable. We define the (average) image error as the value of (3.9) divided by $2n$.

Model 2: Uncertainty Polyhedron

Let us consider another case where the noise is confined in a small range. Digitization noise is one example of such cases. We assume that the true image of a point is confined in a rectangle centered at the observed image position of the point. Such a rectangle is

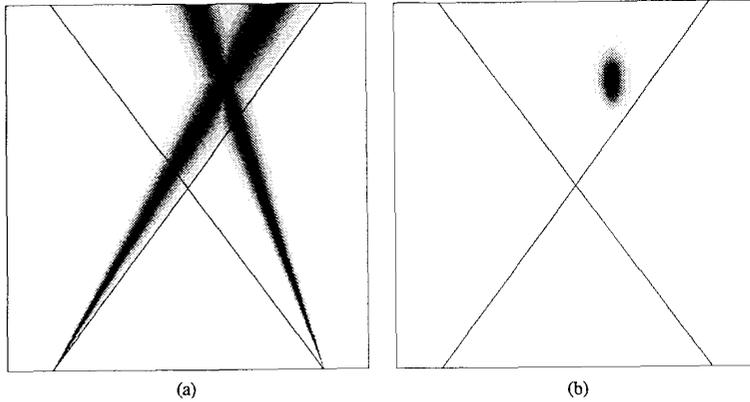


Fig. 2 The distribution of a 3-D points given its images. Darker areas have higher probability (the half tone images in this paper have 9 printable graytones). Two diagonal lines across the figure are optical axes of two cameras. (a) two observed projection lines, with error distribution, intersect. (b) two observed projection lines, with error distribution, determine the distribution of the point in 3-D.

called uncertainty rectangle. The true 3-D position of the point is confined in an infinite pyramid defined by the focal point as apex and the uncertainty rectangle as a cross-section as shown in Fig.3. The second view defines another pyramid for the point. The 3-D position of the point must be inside the volume of intersection of these two pyramids. Such an intersection is called uncertainty polyhedron.

Assume the image positions of a point in two images are exactly correct, the corresponding uncertainty polyhedron has a volume V . If those image positions are perturbed by noise, the volume of the uncertainty polyhedron will generally decrease to a value v . If the perturbation is so large that the two pyramids do not intersect, the volume of the uncertainty polyhedron v is zero. So we assume that the probability for a point to be confined in the uncertainty polyhedron with a volume v is equal to v/V .

Without knowing the true 3-D position of a point, the volume V can be estimated as the corresponding volume of uncertainty polyhedron of the midpoint of the line segment L connecting two observed projection lines (Fig. 1). This estimate is accurate enough since the value of V is not sensitive to a small perturbation to a 3-D point.

We assume, as before, the conditional probability for the point to lie in the intersection is independent between different points. Let the observed point i have intersection v_i and the volume of the corresponding largest possible intersection be V_i . The event considered here is that the 3-D feature points are all confined in the corresponding intersections. Then the conditional density in (3.4) should become discrete conditional probability distribution. Thus the conditional probability that every point lies in the corresponding uncertainty polyhedron can be written as

$$p_{UIM,X}(u|m,x) \approx \prod_{\text{points}} \frac{\text{volume of intersection}}{\text{volume of largest possible intersection}} = \prod_{i=1}^n \frac{v_i}{V_i} \quad (3.10)$$

Now, let us assume that the cameras are perturbed by a small amount away from the position at which the image was taken. Any perturbation may increase the volume of the intersection for some individual point but generally it decreases the volume of intersection for most points. If the perturbation is large enough, the intersection volume for a point becomes zero and so the conditional probability becomes zero. Thus perturbation of the camera position away from the correct position decreases the conditional density in (3.10) gen-

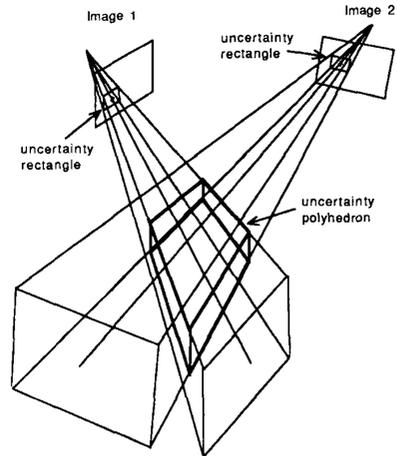


Fig. 3 The intersection of two pyramids defines uncertainty polyhedron.

erally. If the perturbation is large, the conditional density becomes zero.

Based on the maximum likelihood principle, the maximum likelihood estimates of the motion parameters are such that the conditional probability reaches the maximum.

If the estimated parameters are such that the corresponding conditional probability in (3.10) is equal to zero, the results must not be a maximum likelihood estimate since the conditional probability in (3.10) for the true parameters is not zero (for implementation, we make sure that the volume for a mere touch is positive). Therefore the maximum likelihood must correspond to a conditional probability that is nonzero.

The conditional density in (3.10) does not explicitly depend on the true 3D position of feature points. This is what we need since the optimization process would not have to iterate on all points which is otherwise computationally extremely expensive (iteration on n points needs $3n$ -dimensional parameter space!). After the motion parameters are estimated by the maximum likelihood estimates, the 3-D position of each feature point is naturally estimated by the mean of the corresponding intersection. Since the uncertainty polyhedron for the maximum likelihood estimate is always nonempty, we can always calculate the mean position of the intersection.

If the conditional density corresponding to the final estimates is positive (which is the case for all the simulations performed), the results are at least a valid solution from the images given. In other words, we obtain a structure of the scene and the corresponding motion such that the projection of each point is within the noise range. The maximum likelihood estimates makes sure that for the answer obtained, the observed data has the maximum probability.

It is easy to see that if the given motion parameter has large errors such that a point does not lead to any intersection, the conditional probability in (3.10) is zero. In such a case, the conventional iterative procedure for a maximum will be trapped at a point since the gradient vanishes. So the iterative procedure needs a good estimate of motion parameters to start with. The initial guess provided by the linear algorithm does not always guarantee nonzero intersection. Therefore we need a model for the distribution of the noise so that the point is not restricted to a finite area. The corresponding objective function of Gaussian distribution can be used as intermediate criterion for iteration before using the uncertainty polyhedron model.

Line Correspondences

When the images have few distinctive points but many lines, the lines (without known end points) can be used as features for motion analysis. With line correspondences, at least 13 line correspondences over three views are required for a linear algorithm to give a unique solution [Weng88a]. More lines are needed to provide overdetermination in the presence of noise.

The observables for lines are the normal of the plane passing through the focal point and the line in the image. Those observables play a role similar to points in the case of point correspondences.

4. IMPLEMENTATION

This section discusses some aspects of implementation of the algorithms discussed in the previous sections.

First, for Gaussian distribution, equation (3.9) involves both motion parameters and 3D position of every feature point. The maximum is over all the possible motion parameters and scene structures. The parameter space for iteration is huge and computation is very expensive. In implementation, we don't have to iterate on the scene structure. An exact solution of the optimal point position requires solving a fourth order polynomial equation. By a reasonable approximation, we can get a closed form solution. The exact solution is far from the midpoint of the line segment L when the distance to the object and the viewing angle are not drastically different from two images. In simulations, only small performance difference is observed between the midpoint and the analytical solution. For simplicity, we can just choose the mid point of the line L as an approximated optimal point.

For the model of uncertainty polyhedron we need to calculate the volume of the intersection as well as the largest intersections in the neighborhood of the 3-D points. The shape of the intersection of two arbitrary pyramids at arbitrary orientations is very irregular. Exact computation through exhaustive procedure will be very inefficient for numerical calculations. We use a simple and efficient way to calculate the volume of the intersection. First the region of intersection is circumscribed by a regular diamond-like polyhedron (we call it diamond here). The circumscribing diamond is divided into eight octants along its three axes. The division is similar to octree except the axes here is not orthogonal. The 8 octants are recursively subdivided into 8 suboctants, until the desired resolution is reached. Then intersection can be checked recursively for each block using octree techniques introduced in [Weng87b]. Simulation suggested the performance does not improve much for the tree level beyond 3.

For both Gaussian model and uncertainty model, we need to compute nonlinear least squares solution. For uncertainty polyhedron model, the maximization of a product of n factors corresponds to the maximization of the sum of their logarithms, which in turn corresponds to the minimization of a sum of squares as we did for the Gaussian distribution. Therefore the maximum likelihood problem is reduced to a nonlinear least squares problem for both models. The derivative free analogues of the Levenberg-Marquardt and Gauss algorithms by Brown [Brow72] is used for experiments. An implementation of the algorithm is available in IMSL with subroutine name ZXSSQ.

5. EXPERIMENTAL RESULTS

To show the performance of the approach, simulations are performed for the algorithm from point correspondences. For the results shown here the ZXSSQ subroutine in IMSL library is used for iteration in the second step.

In the simulations the focal point of the camera is located 11 units away from the center of the object. The focal length is one unit. The image is a two by two square. The field of view is determined by the size of the image and the focal length. The object

feature points are generated randomly according to a uniform distribution in a cube of 10 by 10 by 10 the center of which is the center of the object. The simulated camera undergoes a motion such that the whole object is still in the field of view for both images, before and after motion. The image coordinates of the points are digitized according to the resolution of the camera. If the resolution is k by k , horizontal and vertical coordinates each has uniformly spaced k levels. The positions of these levels correspond to the locations of the pixels. The image coordinates are rounded-off to the nearest levels before they are used by the motion estimation algorithm. These round off errors result in the errors in the motion parameters and the relative depths calculated by the algorithm. Though the noise is just simulated by digitization noise, it may represent other kinds of noise. For example, additive noise can be simulated by a reduced resolution. All errors shown in this section are relative. Relative error of a matrix, or vector, is defined by the Euclidean norm of the error matrix, or vector, divided by the Euclidean norm of the estimated matrix, or vector, respectively.

If the noise is moderate (resolution is 128 by 128 or higher), the initial guess provided by the linear algorithm [Weng87a] in the first step is generally good enough to ensure the convergence for the iterative procedure. When the noise is large, for example the resolution is 64 by 64 or lower, the initial guess given by the linear algorithm is severely corrupted by noise. To get reasonable results under such resolutions when convergence can not be obtained, the initial guess can be modified so that the algorithm skips to another iteration region. To do so, the initial guess for rotation angle is equal to zero (since the rotation is usually small) and for the direction of translation, the parameters ϕ and ψ in (4.1) are on a coarse grid of their space. When the corresponding motion parameters give all positive depths, the search on the grid terminates and the motion parameters are improved through iterations.

First, to investigate whether the initial guess provided by the linear algorithm indeed helps the second step, a fixed guess is used as the initial guess for the second step. The initial guess is a zero rotation angle and a translation vector of (1, 1, 1). The image resolution is 256 by 256 and sign reversal for the translation and rotation angle is performed when convergence is not achieved. Different motion parameters are chosen randomly. Among 36 examples with 12 point correspondences, 16 of them do not converge, or converge to a wrong answer. A wrong answer means that the error of translation is larger than 100% or error of rotation matrix is larger than 50%. Remaining cases give correct solutions with the translation error less than 10% and the rotation error less than 5%. This shows that a good initial guess is needed to ensure a correct solution.

To demonstrate the improvement of the maximum likelihood estimation (second step) over the results of linear algorithm (first step), the errors in the initial guess given by the linear algorithm are compared with the results of the maximum likelihood estimation. Since it is not true that the iteration will always improve the initial guess, the average behavior is what we are interested in. From Fig.5 to Fig. 10, the errors shown are average errors over 50 random trials, with randomly generated points. Image resolution is 256 by 256, and 12 point correspondences are used.

Fig.5 to Fig. 8 show the comparison for Gaussian model. The rotation axis is aligned with (1, 0.8, 0.9). The rotation angle is 8° . The translation vector has a length of 3 units. The direction of translation changes from (1, 0, 0) to (0, 0, 1) in X-Z plane at evenly spaced 21 steps, corresponding to the X-coordinates from 0 to 20, respectively. The corresponding errors of the solution of the linear algorithm are shown by the solid curve, and those of the final iteration results are shown by the dashed curve. Fig.5 shows the errors of R . Fig. 6 gives those of T . From these two figures it can be seen that the errors are large for the first step with small X index. When the index increases, the errors decrease. The relationships between the errors and the motion parameters and system parameters are dis-

cussed in [Weng88b]. The average errors of recovered depth of points are shown in Fig. 7. Fig. 8 compares the image errors of the first and second steps. It is clear the image errors of the second step are very consistent. This is what is expected. The average image error should be close to the average errors in the image coordinates.

Fig. 9 to Fig. 12 are the results for uncertainty polyhedron model using the same set of motion parameters. The uncertainty rectangle corresponds to a pixel for the simulation. Very similar performance is observed for Gaussian model and uncertainty polyhedron model. This means that the errors are not very sensitive to the moderate changes in the assumed noise distribution. It can be predicted that the assumption of Gaussian distribution generally yields good performance, though in reality the noise distribution is not exactly Gaussian.

The algorithm has been tested on real images. Fig. 13 shows a pair of a scene of our laboratory. A CCD video camera with roughly 512 by 512 resolution is used as imaging sensor. The focal length of the camera is simply calibrated but no nonlinearity correction is made for the camera. The camera takes two images at different positions. A two-view matcher computes image displacement field and occlusion on pixel grid [Weng88c]. The displacement field is shown, on a 13 by 14 grid, overlaid on the image in Fig. 14. Those $13 \times 14 = 182$ displacement vectors are used as point correspondences to compute motion parameters. The motion parameters computed are shown in the following table.

Data and Results for Images of Laboratory Scene			
Translation	-0.006601	-0.980438	-0.196718
Rotation axis	0.916341	0.315033	-0.247132
Rotation angle		-0.958501°	
Image error		0.000631	
Pixel width		0.001111	

Since no attempt is made to obtain the ground truth, we do not know the accuracy of those motion parameters. However, the image error is about half a pixel width, which is very satisfactory.

6. CONCLUSIONS

A robust approach is proposed for the problem of motion and structure estimation. The first step uses a robust linear algorithm to obtain preliminary estimates to the motion parameters and the scene structure. Then the second step modifies the motion parameters and structure using maximum likelihood criterion. Although the second

step does not always reduce the errors in motion parameters and structure for every individual case, the theoretical results on maximum likelihood estimates and the simulations show that on average the errors in the estimated motion parameters and the structure of the scene after the second step are remarkably reduced.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation under grants ECS-83-52408 and IRI-86-05400

REFERENCES

- [Brow72] K. M. Brown and J. E. Dennis, Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation, *Numerische Mathematik*, vol 18, 1972, pp 289-297.
- [Weng87a] J. Weng, N. Ahuja, and T. S. Huang, Error analysis of motion parameters estimation from image sequences, in *Proc. Inter. Conf. Computer Vision*, London, England, June, 1987.
- [Weng87b] J. Weng and N. Ahuja, Octree of objects in arbitrary motion: representation and efficiency, *Computer Vision, Graphics, and Image Processing*, 39, 1987, pp 167-185.

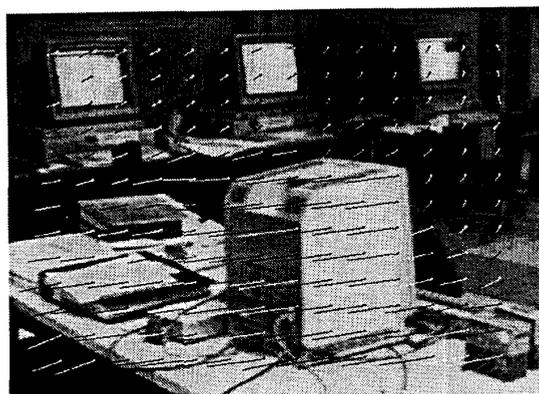


Fig. 14. Sample of the displacement field computed

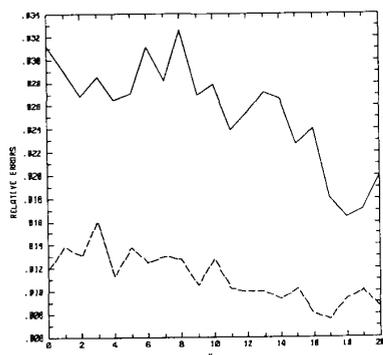


Fig. 5. Relative errors of R of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (Gaussian model)

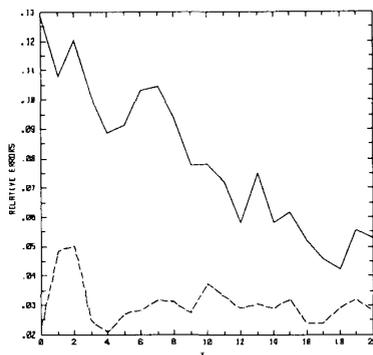


Fig. 6. Relative errors of T of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (Gaussian model)

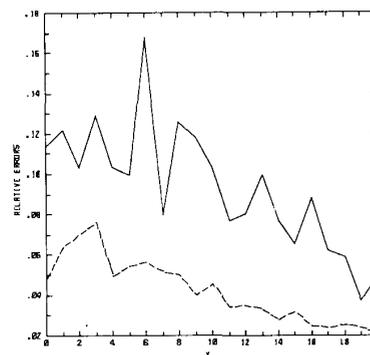


Fig. 7. Relative errors of recovered depth of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (Gaussian model)

[Weng88a] J. Weng, Y. C. Liu, T. S. Huang, and N. Ahuja, Estimating motion/structure from line correspondences: a robust linear algorithm and uniqueness theorems, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June 5-9, 1988.

[Weng88b] J. Weng, T. S. Huang, and N. Ahuja, Motion and structure from point correspondences: algorithm, error analysis and error estimation, to appear in *IEEE Trans. Pattern Anal. Machine Intell.*

[Weng88c] J. Weng, N. Ahuja and T. S. Huang, A two-view/stereo matcher, to be submitted to ICCV 88.

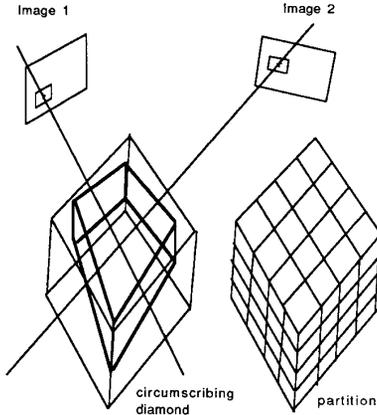


Fig. 4 Circumscribing diamond and partitions.

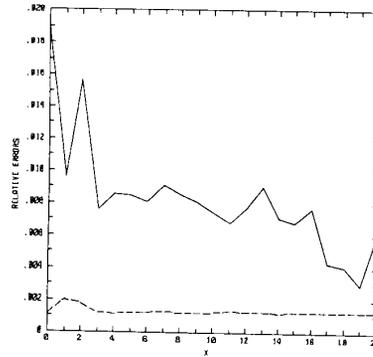


Fig. 8. Image errors of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (Gaussian model)

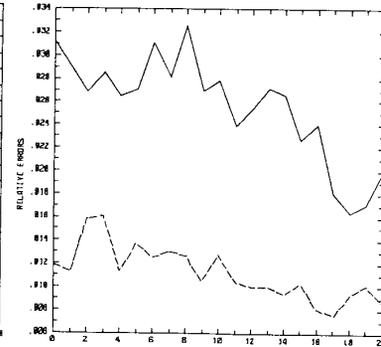


Fig. 9. Relative errors of R of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (uncertainty polyhedron model)

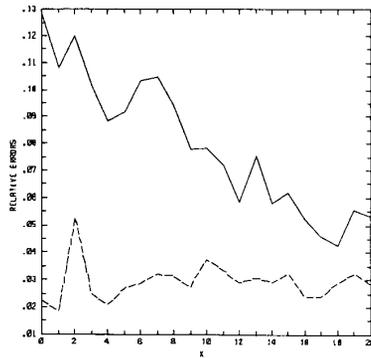


Fig. 10. Relative errors of T of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (uncertainty polyhedron model)

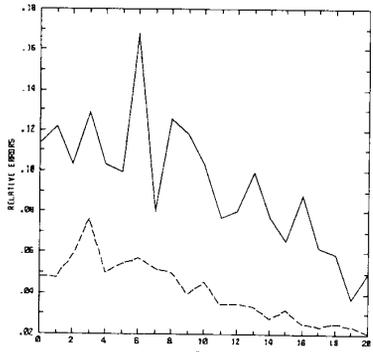


Fig. 11. Relative errors of recovered depth of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (uncertainty polyhedron model)

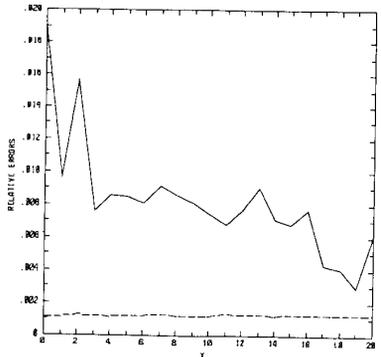


Fig. 12. Image errors of first step (solid curve) and those of the second step (dashed curve) versus different translation directions (uncertainty polyhedron model)

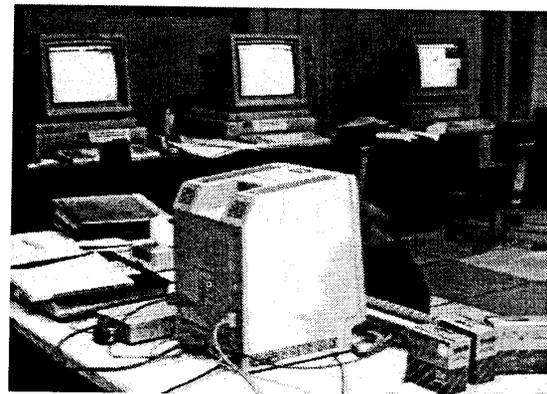
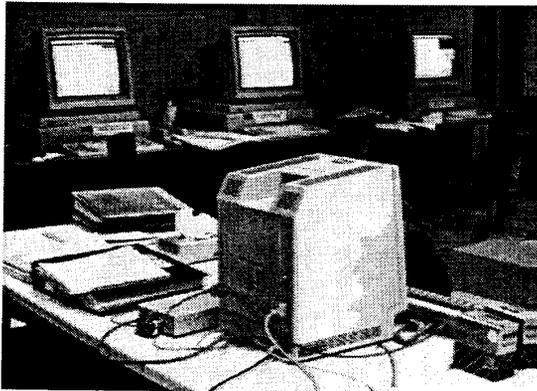


Fig. 13. Two views of a laboratory scene