MOTION FROM IMAGES: IMAGE MATCHING, PARAMETER ESTIMATION AND INTRINSIC STABILITY

JUYANG WENG, THOMAS S. HUANG, NARENDRA AHUJA

Coordinated Science Laboratory
University of Illinois, Urbana, IL 61801

Abstract

This paper first presents an image matching algorithm that uses multiple attributes associated with a pixel to yield a generally overdetermined system of constraints, taking into account possible structural discontinuities and occlusions. Both top-down and bottom-up data flows are used in multi-resolution computational structure. The matching algorithm computes dense displacement fields and the associated occlusion maps. The motion and structure parameters are estimated through optimal estimation (e.g., maximal likelihood) using the solution of a linear algorithm as an initial guess. To investigate the intrinsic stability of the problem in the presence of noise, a theoretical lower bound on error variance of the estimates, Cramér-Rao bound, is determined for motion parameters. Experiments showed that the performance of our algorithm has essentially reached the bound. In addition, the bounds show that, intrinsically, motion estimation from two perspective views is a fairly stable problem if the image disparities are relatively large, but is unstable if the disparities are very small (as required by optical flow approaches).

1 INTRODUCTION

Estimating motion and structure parameters from image sequences has been a very active research topic in computer vision. Difficulties have been persistent in two basic problems: (1) Reliable image matching — establishing correspondences between images, in the form of discrete feature (or feature set) matches, displacement fields or optical flows. The matching algorithms have to deal with a wide range of real world scenes with both textured and uniform surfaces, depth discontinuities and occlusions. (2) Reliable computation of motion and structure parameters from the correspondences. (No approaches have been proposed so far that truly do not need any correspondences.) Since the early linear algorithms ([Long81], [Tsai84] from point correspondences and [Zhua84] from optical flow), the stability of estimating motion from image sequences has been controversial.

This paper reports our recent advances in attacking these two problems. We will first present our approach to image matching. To be applicable to complex real world scenes, the matching algorithm is designed to deal with: (1) both textured and uniform surfaces, (2) both small and large image disparities, (3) discontinuities in depth and the displacement fields and (4) occlusions. Bottom-up data flow is used in addition to top-down data flow in multi-resolution structure to improve the accuracy of the results. Then, we briefly discuss an algorithm that computes the motion parameters from the computed point correspondences, assuming the scene is a rigid. Finally, we determine the theoretical error bounds for the motion parameters, which enable us to evaluate the performance of the algorithm and the intrinsic stability of the problem. The bounds also show the intrinsic limitation of optical flow based approaches.

The next section presents our matching algorithm. Section 3 deals with estimating motion and structure from the computed point correspondences. Section 4 discusses the theoretical performance bounds. The experimental results are presented in Section 5. Section 6 presents concluding remarks.

2 IMAGE MATCHING

2.1 A Framework of Image Matching

A monochrome digital image consists of a two-dimensional array of intensity values. What we perceive from such an image is not just individual intensity values, but more importantly, the spatial arrangement or patterns of those values. Intensity, edgeness and cornerness are examples of the attributes that describe the local intensity pattern around a pixel. As we will see, the criteria that the matched points should have the similar attributes generally provide overdetermination in matching process. Such an overdetermination is very important for reliably matching complicated real world scenes.

Regions with uniform intensity often result from the same continuous surface. This suggests that a uniform region will have a uniform displacement field. We call this *intra-regional smoothness criterion*. The objective of this criterion is to fill in displacement information in those areas where no significant intensity variation occurs. We cannot generally assume smoothness across different regions.

Occlusion occurs when a part of scene visible in one image is occluded in the other by the scene itself, or a part of the scene near the image boundary moves out of field of view in the other image. If occlusion regions are not detected, they may be incorrectly matched to nearby regions, interfering with the correct matching of these regions. To identify occlusion regions, we define two occlusion maps, occlusion map 1 showing parts of image 1 not visible in image 2, and similarly occlusion map 2 for image 2 (in Figure

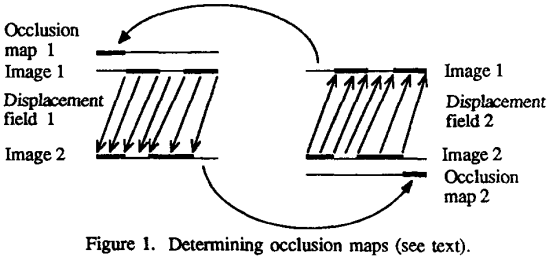

Figure 1. Determining occlusion maps (see text).

1, black areas denote occlusion regions). We first determine the displacement field from image 2 to image 1, without occlusion information. The objective of this matching process is to compute occlusion map 1. This matching may "jam" the occluded parts of image 2 (e.g., the right-most section) into parts of image 1 (e.g., the right-most section). This generally will not affect the computation of occlusion map 1. Those areas in image 1 that have not been matched (in Figure 1, no arrows pointing to them) are occluded in image 2 and are marked in occlusion map 1. Once occlusion map 1 is obtained, we then compute the displacement field from image 1 to image 2 except for the occluded regions of image 1. The results of this step determine occlusion map 2.

Large disparities are crucial for stability of motion and structure estimation. However, to find matches with large disparities requires that we know approximate locations of the matches, since otherwise multiple matches may be found. One solution to this problem is image blurring to filter out high spatial frequency components. Because blurred intensity image has very few features left, and their locations are unreliable, we blur the original edgeness and cornerness images (called attribute images here) Since the cornerness measure has a sign, nearby positive and negative corners may be blurred to give almost zero values, which is the same as the result of blurring an area without corners. We therefore separate positive and negative corners into two attribute images. Blurring is done for positive and negative images separately. Such blurred edgeness and cornerness images are not directly related to the blurred intensity images. They are related to the strength and frequency of occurrence of the corresponding features, or to the texture content of the original images. While texture is lost in intensity images at coarse levels, the blurred edgeness and cornerness images retain a representation of texture, which is used for coarse matching. The intraregional smoothness constraint at coarse levels applys to blurred uniform texture regions (with averaged intensity). When the computation proceeds to finer levels, the sharper edgeness and cornerness measures lead to more accurate matching. Therefore, in general the algorithm applys to both textured or non-textured surfaces.

At a coarse resolution, the displacement field only needs to be computed along a coarse grid, since the displacement computed at a coarse resolution is not accurate, a low sampling rate suffices. In the approach described in this paper, the coarse displacement field is projected to the next finer level (copied to the four corresponding grid points) where it is refined. Such a refinement continues down to finer levels successively until we get the final results at the original resolution. The computational structure and data flow used in this process are shown in Figure 2.

2.2 Matching Algorithm

Now, we present the matching algorithm we have developed to implement the approach outlined in the previous section. Let the position of a point in an image be denoted by $\mathbf{u}=(u, v)$. Let the intensity of the first image be denoted by $i(\mathbf{u})$ and that of the second image by $i'(\mathbf{u})$. The objective of the algorithm is to compute displacement field $\mathbf{d}(\mathbf{u})$ such that $i(\mathbf{u})$ and $i'(\mathbf{u}+\mathbf{d})$ are the projections of the same scene point in the two images.

In our implementation, *edgeness* at a point is defined by magnitude of the gradient at the point. For cornerness, we consider positive and negative cornerness separately. Roughly speaking, the cornerness at a point $\mathbf{u}$ measures the changes of the direction of gradient at two nearby points, weighted by the gradient at the point. The closer the change of angle is to $\pi/2$, the higher the positive cornerness measure. For more detailed discussion of the cornerness, see [Weng88b].

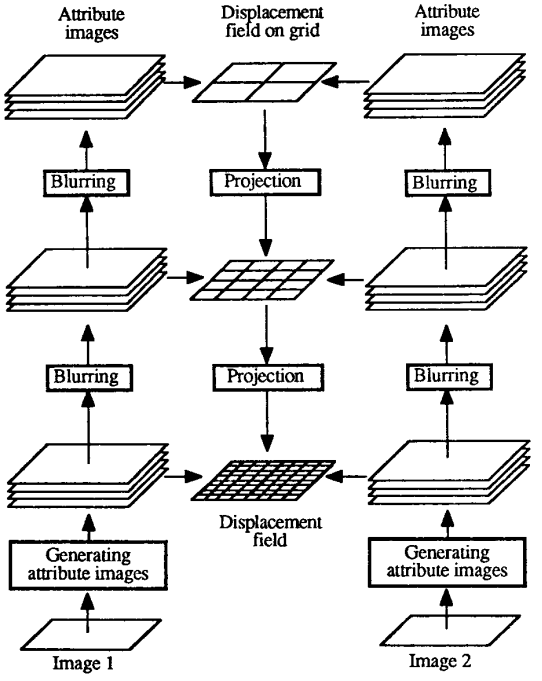

Figure 2. Computational structure and data flow

We separately consider the smoothness of the orientation of displacement vectors to emphasize its role in matching. The reason is that (1) the orientation of the displacement vectors projected from a coarse level is generally more reliable than their magnitude, and (2) at a fine level, the local attribute gradient perpendicular to the displacement vector can easily lead the displacement vector in a wrong direction if orientational smoothness is not emphasized.

Clearly, smoothness constraint should be enforced only over points whose displacements are related, e.g., over adjacent points from the same surface. To selectively apply the smoothness constraint to two points, we use the similarity of intensities and the similarity of available displacement vector estimates at the two points. We represent the displacement vector filed in the vicinity of a point $\bar{\mathbf{d}}(\mathbf{u}_0)$ by a vector $\bar{\mathbf{d}}(\mathbf{u}_0)$. It is intended to approximate the displacement filed within the region that $\mathbf{u}_0$ belongs to. In the implementation, $\mathbf{u}_0$ is computed as:

$$\bar{\mathbf{d}}(\mathbf{u}_0)= \sum_{0<\|\mathbf{u}-\mathbf{u}_0\|<r} w(i(\mathbf{u})-i(\mathbf{u}_0), \mathbf{d}(\mathbf{u})-\mathbf{d}(\mathbf{u}_0))\, \mathbf{d}(\mathbf{u})$$

where $0<\|\mathbf{u}-\mathbf{u}_0\|<r$ denotes a region around $\mathbf{u}_0$, and $w(\cdot,\cdot)$ denotes the weight assigned to the displacement vector at a neighboring point $\mathbf{u}$. In digital implementation, $\{\mathbf{u}\}$ are adjacent grid points (8-connectivity). The weight is a function of intensity difference $i(\mathbf{u})-i(\mathbf{u}_0)$, and displacement vector difference $\|\mathbf{u}-\mathbf{u}_0\|$. Let $\eta_i=|i(\mathbf{u})-i(\mathbf{u}_0)|$ and $\eta_d=\mathbf{d}(\mathbf{u})-\mathbf{d}(\mathbf{u}_0)$. A definition of weight is as follows:

$$w(\eta_i,\eta_d) = \frac{c}{\varepsilon+|\eta_i|(1+\|\eta_d\|^2)} \qquad (2.1)$$

where ε is a small positive number to reduce the effects of noise in intensity and prevent denominator from becoming 0, and c is a normalization constant which makes the sum of weights equal to 1:

$$\sum_{0<\|\mathbf{u}-\mathbf{u}_0\|<\sigma} w(i(\mathbf{u})-i(\mathbf{u}_0),\ \mathbf{d}(\mathbf{u})-\mathbf{d}(\mathbf{u}_0))=1$$

In (2.1), the term $\|\eta_d\|^2$ should be replaced by zero for the first half number of iterations at each level, since the displacement vectors are not reliable when they are projected from a coarser level. Another definition of the weighs is

$$w(\eta_i,\eta_d) = \frac{c}{\varepsilon+|\eta_i|}$$

If η_i and $\|\eta_d\|$ are large (determined dynamically at different levels), $w(\eta_i,\eta_d)$ is equal to zero. The goal is to reduce the weight if the intensity difference is large (across different regions) and displacement vectors are quite different (field discontinuity occurs).

Thus, the weight is automatically determined based on intensity difference and displacement difference. The smoothness constraint imposes similarity of $\mathbf{d}(\mathbf{u}_0)$ and $\bar{\mathbf{d}}(\mathbf{u}_0)$. The larger the difference in intensity, the more easily the fields for two adjacent regions can differ. If two regions get different displacements after some iterations, the quadratic term $\|\eta_d\|^2$ results in very small weight to reduce their interactions. On the other hand, the displacement vectors in the same region will be similar since the corresponding weight is large. Since intensity difference is usually much larger than the magnitude of displacement difference, $|\eta_i|$ is not squared in (2.1) (unlike η_d), otherwise the weight will be too sensitive to small changes in intensity. The weights, thus, implicitly take into account discontinuities. The registered value $\bar{\mathbf{d}}(\mathbf{u}_0)$ allows us to perform matching using uniform numerical optimization despite the presence of discontinuities. This is discused below.

Any given displacement vector field leads to measures of similarity, or residual errors, between the attributes of estimated corresponding points. The residuals for various attributes are:

(1). Residual of intensity:

$$r_i(\mathbf{u},\mathbf{d})=i'(\mathbf{u}+\mathbf{d})-i(\mathbf{u})$$

(2). Residual of edgeness:

$$r_e(\mathbf{u},\mathbf{d})=e'(\mathbf{u}+\mathbf{d})-e(\mathbf{u})$$

(3) Residual of positive cornerness:

$$r_p(\mathbf{u},\mathbf{d})=p'(\mathbf{u}+\mathbf{d})-p(\mathbf{u})$$

(4) Residual of negative cornerness:

$$r_n(\mathbf{u},\mathbf{d})=n'(\mathbf{u}+\mathbf{d})-n(\mathbf{u})$$

(5) Residual of orientation smoothness:

$$r_o(\mathbf{u},\mathbf{d})=\|\mathbf{d}(\mathbf{u})\times\bar{\mathbf{d}}(\mathbf{u})\|/\|\bar{\mathbf{d}}(\mathbf{u})\|$$

(6) Residual of displacement smoothness:

$$r_d(\mathbf{u},\mathbf{d})=\|\mathbf{d}(\mathbf{u})-\bar{\mathbf{d}}(\mathbf{u})\|$$

We want to minimize the weighted some of squares of residuals:

$$\sum_{\mathbf{d}}\{r_i^2(\mathbf{u},\mathbf{d})+\lambda_e r_e^2(\mathbf{u},\mathbf{d})+\lambda_p r_p^2(\mathbf{u},\mathbf{d})$$

$$+\lambda_n r_n^2(\mathbf{u},\mathbf{d})+\lambda_o r_o^2(\mathbf{u},\mathbf{d})+\lambda_d r_d^2(\mathbf{u},\mathbf{d})\} = \min \qquad (2.2)$$

where λ_e, λ_p, λ_n, λ_o and λ_d are weighting parameters that are dynamically adjusted at different resolutions. Let

$$\mathbf{r} \triangleq (r_i, r_e, r_p, r_n, r_o, r_d)^T$$

With previous estimate of the displacement vector $\mathbf{d}$ (initially $\mathbf{d}$ is a zero vector at the highest level), we need to find increment $\delta_\mathbf{d}$. Expanding $\mathbf{r}(\mathbf{u}, \mathbf{d}+\delta_\mathbf{d})$ at $\delta_\mathbf{d}=\mathbf{0}$, we have (suppressing variable $\mathbf{u}$ for conciseness):

$$\mathbf{r}(\mathbf{d}+\delta_\mathbf{d}) = \mathbf{r}(\mathbf{d})+\frac{\partial \mathbf{r}(\mathbf{d})}{\partial \mathbf{d}}\delta_\mathbf{d}+o(\|\delta_\mathbf{d}\|) \triangleq \mathbf{r}+A\,\delta_\mathbf{d}+o(\|\delta_\mathbf{d}\|) \quad (2.3)$$

where

$$A=\frac{\partial \mathbf{r}(\mathbf{d})}{\partial \mathbf{d}}=\begin{bmatrix} \frac{\partial i'}{\partial u} & \frac{\partial e'}{\partial u} & \frac{\partial p'}{\partial u} & \frac{\partial n'}{\partial u} & -\bar{d}_v/\|\bar{\mathbf{d}}\| & 1 & 0 \\ \frac{\partial i'}{\partial v} & \frac{\partial e'}{\partial v} & \frac{\partial p'}{\partial v} & \frac{\partial n'}{\partial v} & \bar{d}_u/\|\bar{\mathbf{d}}\| & 0 & 1 \end{bmatrix}^T \quad (2.4)$$

where $(\bar{d}_u, \bar{d}_v)^T=\bar{\mathbf{d}}$, the partial derivative $\frac{\partial i'}{\partial u}$ denotes the partial derivative of $i'(u, v)$ with respect to u at point $\mathbf{u}+\mathbf{d}$, and so on. Let

$$\Lambda = diag\{1, \lambda_e, \lambda_p, \lambda_n, \lambda_o, \lambda_d\}$$

We want to find $\delta_\mathbf{d}$ such that the sum of squared residuals in (2.2) at the point is minimized. Neglecting high order terms and minimizing $\|\Lambda(\mathbf{r}+A\,\delta_\mathbf{d})\|$, from (2.3) we get the formula for updating $\mathbf{d}$:

$$\delta_\mathbf{d} = -(A^T\Lambda^2 A)^{-1}A^T\Lambda^2\,\mathbf{r}(\mathbf{u})$$

The partial derivatives in the entries of A are computed by a finite difference method. Let s denote the distance between two adjacent points on a grid, along which finite deference of the attributes is be computed, assuming a unit spacing between adjacent pixels. Then s should vary with the resolution. In addition, s should also also vary with successive iterations within a resolution level. A large spacing is necessary for a rough displacement estimate when iterations start at a level. As iterations progress, the accuracy of the displacement field increases and s should be reduced to measure local structure more accurately. The spacing s of a 3 by 3 difference mask at level l is equal to 2^l for the first one-half number of iterations at level l, and is reduced by a factor of 2 for the second half, except for l=0. At the original resolution (l=0), the spacing is always equal to 1, since no smaller spacing is available on pixel grid.

The blurring of level l+1 is done using the corresponding attribute image at level l: For each pixel at level l+1, its value is equal to the sum of the value of four pixels at level l divided by m (m=4 for intensity, m=3 for edgeness and m=2 for cornerness). The locations of these four pixels are such that each is centered at a quadrant of a square of $s\times s$. s is equal to 2^l at level l. Therefore, the blurred intensity image at level l is equal to the average over all pixels in a square of size $s\times s$. To enhance sparse edges and corners, m is smaller than 4 for edgeness and cornerness. So, the results can be larger than 255. If this occurs, the resulting value is limited to 255. This multilevel recursive normalization is useful for the algorithm to adapt to different scenes. Some details of the matching algorithm that are not covered here were presented in [Weng88b].

2.3 Bottom-Up and Top-Down Refinements

The computational structure illustrated in Figure 2 can be extended further to improve the displacement field. The data flow characterized by the projections shown in Figure 2 is of a top-down fashion in the sense that the displacement fields are computed from high levels (coarser resolution) down to low levels. At high levels, coarse estimates of the field are computed to scale down the possible matching area. At low levels, details of the displacement fields are computed. However, at a coarse level, different initial estimates may result in different results. The more accurate the initial estimate is, generally the more accurate the final result will be. Since the result of a lower level is generally more accurate than that of a higher level, the result of a lower level can

be used as an initial estimate for a higher level. We may also observe the problem in a slightly different way: The result of a coarse level needs to be verified and refined at low level when more detailed image is available. Such refined local field needs to be propagated to wider areas. One computationally efficient way to do this is to go up to the higher levels where a coarse grid is available. These considerations motivate the bottom-up scheme — the result of a lower level is projected up to a higher level as an initial estimate. The upward projection is done as follows: The initial value of the grid point at a higher level is the average of the values of the corresponding grid points at the lower level. Then another pass of computation is performed from the higher level to the lower level, at which a refined field is obtained. A multiple of such a bottom-up and top-down structure can be embedded in the entire algorithm. Figure 3 shows an example of the computational structure which we implemented. Upward projections cross two levels to make significant refinement. The initial estimate of each level (levels 5, 4, 3, 2 in Figure 3) is refined by the next two lower levels until a level (level 2) is reached whose refinement needs a level below the lowest level (0 in Figure 3). Such a refinement of initial estimate results in visible improvements for most scenes.

3 Optimal Motion and Structure Estimation from Point Correspondences

In reality, given the computed point correspondences (or displacement field) between two images, the observed 2-D image coordinate vectors $\mathbf{u}_i$ of image 1 and $\mathbf{u}'_i$ of 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 vectors $(\mathbf{U}_i, \mathbf{U}'_i)$. With n point correspondences over two time instants, we add subscripts i to denote the i-th point and the subscript-free letters denote the general collection of vectors:

$$\mathbf{u} \triangleq (\mathbf{u}_1^T, (\mathbf{u}'_1)^T, \mathbf{u}_2^T, (\mathbf{u}'_2)^T, \cdots, \mathbf{u}_n^T, (\mathbf{u}'_n)^T)^T$$

Let the conditional density of $\mathbf{U}$ conditioned on $\mathbf{M}=\mathbf{m}$ and $\mathbf{X}=\mathbf{x}$ be $p_{\mathbf{U}|\mathbf{M},\mathbf{X}}(\mathbf{u}|\mathbf{m},\mathbf{x})$. The maximum likelihood estimates of motion parameters, $\mathbf{m}^*$, and scene structure $\mathbf{x}^*$ are such that the conditional density $p_{\mathbf{U}|\mathbf{M},\mathbf{X}}(\mathbf{u}|\mathbf{m},\mathbf{x})$ reaches the maximum. To find a maximum likelihood estimate, we need to know the conditional density $p_{\mathbf{U}|\mathbf{M},\mathbf{X}}(\mathbf{u}|\mathbf{m},\mathbf{x})$. Gaussian distribution is commonly used for modeling noise. We assume that the conditional distributions, given motion parameters and scene structure, are independent between different points:

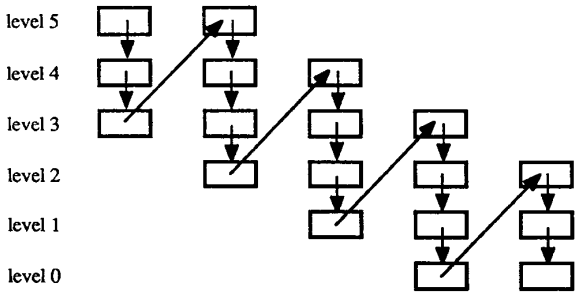

Figure 3 Bottom-up and top-down refinement

$$p_{\mathbf{U}|\mathbf{M},\mathbf{X}}(\mathbf{u}|\mathbf{m},\mathbf{x}) = \prod_{i=1}^{n} p_i(\mathbf{u}_i|\mathbf{m},\mathbf{x})\, p'_i(\mathbf{u}'_i|\mathbf{m},\mathbf{x}) \qquad (3.1)$$

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, p_i is the Gaussian density with mean equal to the exact projected location of the given feature point, $\mathbf{h}_i(\mathbf{m},\mathbf{x})$, in the first image. p'_i is the Gaussian density for the second images. The maximum likelihood estimator leads to minimizing

$$\sum_{i=1}^{n} (\|\mathbf{u}_i - \mathbf{h}_i(\mathbf{m},\mathbf{x})\|^2 + \|\mathbf{u}'_i - \mathbf{h}'_i(\mathbf{m},\mathbf{x})\|^2) \qquad (3.2)$$

We define the *standard image plane error*, or simply *Image error* as

$$[\sum_{i=1}^{n} (\|\mathbf{u}_i - \mathbf{h}_i(\mathbf{m},\mathbf{x})\|^2 + \|\mathbf{u}'_i - \mathbf{h}'_i(\mathbf{m},\mathbf{x})\|^2)/2n\ ]^{1/2} \qquad (3.3)$$

Or, let the computed projections be denoted by a vector $\mathbf{h}$:

$$\mathbf{h}(\mathbf{m}) \triangleq (\mathbf{h}(\mathbf{m})_1^T, (\mathbf{h}'(\mathbf{m})_1)^T, \mathbf{h}(\mathbf{m})_2^T, (\mathbf{h}'(\mathbf{m})_2)^T, \cdots, \mathbf{h}(\mathbf{m})_n^T, (\mathbf{h}'(\mathbf{m})_n)^T)^T$$

where we omitted $\mathbf{x}$ in $\mathbf{h}(\mathbf{m},\mathbf{x})$ since projection $\mathbf{h}$ can be determined from $\mathbf{m}$ only, with $\mathbf{x}$ computed from $\mathbf{m}$ ([Weng88a]). Then, for white Gaussian noise, we want to find $\mathbf{m}$ such that $\|\mathbf{h}(\mathbf{m}) - \mathbf{u}\|$ is minimized.

In practice, the distribution of noise is usually unknown. However, it can be proved that minimizing (3.2) leads to a linear minimum variance estimator for a locally linearized system. In general, the objective function of (3.2) gives very good performance for general noise (a series of related results will appear in a forthcoming paper).

The optimization equation (3.2) is nonlinear in terms of unknown parameters. To deal with local minima problem and improve computational efficiency, the result of a linear algorithm is used as an initial guess for nonlinear optimization. The computational aspects of optimization was presented in ([Weng88a]).

4 Performance Bounds

We need to estimate the parameter vector $\mathbf{m}$ from the noise corrupted version of observation vector $\mathbf{x}$. Due to the random noise in the observation, we cannot determine the parameter vector $\mathbf{m}$ exactly. In other words, we do not have the information to determine the exact solution. The estimates we obtained always have errors. A fundamental question to ask is that what kind the best accuracy one can possibly achieve from the noise corrupted data, independent of the methods used. Since the errors are random, such accuracy is meaningful only in the sense of statistics. Two commonly used measurements of the quality of an estimator are bias and error variance. For error vectors, the corresponding measurements are the bias vector and the covariance matrix of the error vectors. There exist theoretical bounds for the covariance matrix of any estimator. Cramér-Rao bound is one of them.

Cramér-Rao bound [Cram46] [Rao73]. Suppose $\mathbf{m}$ is a parameter of probability density $p(\mathbf{z},\mathbf{m})$. $\hat{\mathbf{m}}$ is an estimator of $\mathbf{m}$ based on measurement $\mathbf{z}$ with $\mathbf{E}\hat{\mathbf{m}}=\mathbf{b}(\mathbf{m})$ ($\mathbf{E}$ denotes expectation). Let $\mathbf{y}^T = \frac{\partial \ln p(\mathbf{z},\mathbf{m})}{\partial \mathbf{m}}$. Define

$$F = \mathbf{E}\mathbf{y}\mathbf{y}^T \qquad (4.1)$$

matrix F is called Fisher information matrix. Denote

$$B = \frac{\partial \mathbf{b}(\mathbf{m})}{\partial \mathbf{m}} \qquad (4.2)$$

Then

$$\mathbf{E}(\hat{\mathbf{m}}-\mathbf{b}(\mathbf{m}))(\hat{\mathbf{m}}-\mathbf{b}(\mathbf{m}))^{\mathrm{T}} \geq BF^{\dagger}B^{\mathrm{T}} \tag{4.3}$$

where the inequality for matrices means that the difference of two sides is positive semi-definite, and $F^{\dagger}$ is the pseudo-inverse of F. The equality holds if and only if

$$\hat{\mathbf{m}}-\mathbf{b}(\mathbf{m}) = BF^{\dagger}\left[\frac{\partial \ln p(\mathbf{z}, \mathbf{m})}{\partial \mathbf{m}}\right]^{\mathrm{T}}$$

almost everywhere.

The proof of the Cramér-Rao bound can be found in [Rao73] and [Zack71]. □

The Cramér-Rao bound provides a lower bound for the variance of the estimator. However, we are often interested in the expected errors instead of variance. We know

$$\mathbf{E}(\hat{\mathbf{m}}-\mathbf{m})(\hat{\mathbf{m}}-\mathbf{m})^{\mathrm{T}} \geq \mathbf{E}(\hat{\mathbf{m}}-\mathbf{E}\hat{\mathbf{m}})(\hat{\mathbf{m}}-\mathbf{E}\hat{\mathbf{m}})^{\mathrm{T}} \tag{4.4}$$

Therefore, Cramér-Rao bound also provides a lower bound for the expected errors. However, the bias vector $\mathbf{b}(\mathbf{m})$ is often unknown for many real world problems. The same is true for the nonlinear problem invested here. Letting $\mathbf{b}(\mathbf{m})=\mathbf{0}$ ($B=I$), Cramér-Rao bound provides a lower bound for the expected errors of any unbiased estimator. Therefore, we can compare the expected errors with that of a "best possible" unbiased estimator using Cramér-Rao bound.

The evaluation of Cramér-Rao bound requires noise distribution. Suppose that the noise is Gaussian (i.e., $\mathbf{f}=\mathbf{h}(\mathbf{m})-\mathbf{u}$ is zero mean white Gaussian vector). With parameters $\mathbf{m}$ (the best $\mathbf{x}$ can be computed from $\mathbf{m}$ [Weng88a]) and observation $\mathbf{u}$, equation (3.1) gives

$$\ln p_{\mathbf{U}|\mathbf{M}}(\mathbf{u}\,|\,\mathbf{m}) = \frac{\mathbf{f}(\mathbf{m})^{\mathrm{T}}\mathbf{f}(\mathbf{m})}{2\sigma^2} - 2n\ln(2\pi\sigma^2)$$

$$\frac{\partial \ln p_{\mathbf{U}|\mathbf{M}}(\mathbf{u}\,|\,\mathbf{m})}{\partial \mathbf{m}} = \sigma^{-2}\mathbf{f}(\mathbf{m})^{\mathrm{T}}\frac{\partial \mathbf{f}(\mathbf{m})}{\partial \mathbf{m}} \tag{4.5}$$

Let

$$J = \frac{\partial \mathbf{f}(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial \mathbf{h}(\mathbf{m})}{\partial \mathbf{m}} \tag{4.6}$$

We get the expression of the Fisher information matrix:

$$F = \mathbf{E}\left[\frac{\partial \ln p_{\mathbf{U}|\mathbf{M}}(\mathbf{u}\,|\,\mathbf{m})}{\partial \mathbf{m}}\right]^{\mathrm{T}}\left[\frac{\partial \ln p_{\mathbf{U}|\mathbf{M}}(\mathbf{u}\,|\,\mathbf{m})}{\partial \mathbf{m}}\right]$$

$$= \sigma^{-4}J^{\mathrm{T}}(\mathbf{E}\mathbf{f}(\mathbf{m})\mathbf{f}(\mathbf{m})^{\mathrm{T}})J = \sigma^{-4}J^{\mathrm{T}}(\sigma^2 I)J = \sigma^{-2}J^{\mathrm{T}}J \tag{4.7}$$

Then, for the unbiased estimator $\hat{\mathbf{m}}$ with independent, uniform variance Gaussian noise, the Cramér-Rao bound gives

$$\Gamma_{\hat{\mathbf{m}}} \leq F^{-1} = \sigma^2(J^{\mathrm{T}}J)^{-1} \tag{4.8}$$

assuming $J^{\mathrm{T}}J$ has a full rank. Notice that J in (4.8) is evaluated with the true $\mathbf{m}$. More generally, if the zero mean Gaussian vector $\mathbf{f}(\mathbf{m})$ are not independent and has a covariance matrix $C_{\mathbf{f}}=\mathbf{E}\mathbf{f}(\mathbf{m})\mathbf{f}(\mathbf{m})^{\mathrm{T}}$, then it is easy to show that the Fisher information matrix is given by $F=J^{\mathrm{T}}C_{\mathbf{f}}^{-1}J$.

When the minimum attainable variance is larger than the Cramér-Rao bound, other tighter bounds can be derived. For example, Bhattacharyya bound gives another bound on covariance [Zack71]. In fact, the Cramér-Rao bound is a special case of the Bhattacharyya bound. Since Bhattacharyya bound involves higher order derivatives of probability density, the computation is more involved. If the actual errors are close to the Cramér-Rao bound (this is true in the experiments we performed), the more general Bhattacharyya bound is obviously very close to Cramér-Rao bound.

In Section 5, simulations show that for the optimized solution, the actual bias is small and the actual errors are very close to the Cramér-Rao bound for unbiased estimators. In other words, the errors are very close to those that would result from the "best possible" unbiased estimator.

5 EXPERIMENTAL RESULTS

First we show the results of simulations where we can control the noise and assess the performance of the algorithm quantitatively. For the simulations, the focal length is one unit. The image is a $s\times s$ square. The field of view is then determined by the image size s and the unit focal length. Unless stated otherwise, s=0.70 (the corresponding field of view is roughly equivalent to a 50mm normal lens of a 35mm camera) and 12 point correspondences are used. The object points are generated randomly between depth 6 and 11. Only those points that are visible before and after motion are used. Random noise is added to the image coordinates of the points. 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 correct matrix, or vector, respectively.

5.1 Essential Reach of Cramér-Rao Bound

The results of the linear algorithm presented in [Weng87] is employed as an initial guess and the nonlinear optimization in Section 3 (see [Weng88a] for details) is performed to improve the initial guess. Figure 4. shows the comparison between the actual relative errors of the final results and the corresponding Cramér-Rao bound for the Gaussian noise. Two types of noise are simulated, Gaussian and uniform (with same variance). The variance of the uniform noise is equivalent to that of digitization noise of a 256×256 image. (Experiments on real images show that the error variance of the points given by the matching algorithm are generally not larger than the quantization noise of a 256×256 image). As shown in Figure 4, the actual relative errors are very close to the Cramér-Rao bound for the Gaussian noise. In other words, the errors of the algorithm are very close to that of a best possible unbiased estimator with Gaussian noise. Figure 4(b) and Figure 4(c) show that the errors are similar for the two types of noise: Gaussian and uniform. This implies that the distribution of noise does not significantly influence the actual errors as long as the variance is kept the same. Figure 4 also shows the relative absolute bias of the estimates (the norm of the bias matrix, or vector, divided by the norm of the true matrix, or vector). The bias is small relative to the actual errors.

Since the performance of the algorithm virtually reach the theoretical lower bounds, there exists no algorithms that can give considerably more accurate estimates from the given data. On the other hand, these results give our fundamental insight into the effects of the amount of errors that may seem negligible at first sight, such as digitization errors. From Figure 4 we know that to give an estimated translation direction of 2.0% error or lower, on average, by the given setup with 12 points, the variance of errors in the points locations cannot be larger than those of digitization noise of a 256×256 image.

5.2 Intrinsic Limitation of Motion from Optical Flow

The discrete approaches are applicable to both small or large inter-frame motions, while the continuous approaches are applicable to only small motions. The restriction of small motion for continuous approaches arises primarily from two facts: (1) Optical flow, by definition, is the projection of 3-D velocity onto image

plane. Therefore, the formulation of computing optical flow is in terms of velocity. (2) The mathematical formulation of computing motion parameters from optical flow is in terms of motion velocity. However, what actually observed is the displacements between images. Only in the case of small motion, can velocity be estimated by displacement.

Although the restriction of small inter-frame motion simplifies both computing image matching (optical flow as a result) and computing motion parameters (motion velocity as a result) from optical flow, the reliability of computed motion parameters in the presence of noise is intrinsically limited. The small amount of motion is easily overridden by the errors in the estimated optical flow, even if the optical flow can be estimated in subpixel accuracy. In other words, the signal to noise ratio in the estimated optical flow is low. Many researchers have been trying to compute motion parameters from optical flow. But so far, no satisfactory numerical results have been reported under realistic setups. There may be a lot one can do to improve the existing algorithms to compute motion parameters and structure from optical flow. However, the theoretical lower error bound using optical flow may be large. We need to investigate the intrinsic limit of motion from optical flow.

As shown in Figure 4, the algorithm (discrete approach) has essentially reached the theoretical error bound. With a relatively large inter-frame motion, the error bound is small (e.g., about 2% in the direction of translation in Figure 4(b)). What about the small motion typically used by optical flow? We consider a setup: The image has 512×512 pixels in a unit square (the field of view is roughly equivalent to that of a f=35mm wide angle lens of a 35mm camera). We assume that the image positions of the points are corrupted by additive white Gaussian noise with a variance equal to that of the uniform distribution in the range of ±1 pixel. The configuration of the random points is the same as mentioned in the beginning of the Chapter. The magnitudes of translation are such that the maximum disparities caused by translation in the image plane are 2, 4, 8 and 16 pixels, respectively. The Cramér-Rao bounds of the relative error in the estimated translation, averaging over 10 random point sets, are shown in Figure 5. It can be seen that, under a small motion with 2-pixel maximum disparity (average disparity is roughly equal to 1 pixel), the errors in translation are bounded below by 60% even using a large number of points (70). A small motion with 4-pixel maximum disparity still causes a large error bound (about 38% with 70 points). Recall that the bound here is for exact algorithms (discrete approaches) and does not include any approximations that a continuous approach may use. In other words, these bounds apply to any algorithms. Therefore, it is intrinsically very unreliable trying to recover motion parameters from small motion with a disparity of a few pixels. The data shown here quantitatively predict the intrinsic unstability for estimating motion and structure from small motions (e.g., by optical flow approaches).

5.3 Real World Images

Experiments have been performed for a variety of real world scenes. A CCD monochrome video camera with roughly 500×480 pixels is used as image sensor. The focal length of the camera is calibrated but no corrections are made for camera nonlinearity. The camera takes two images at different positions for each scene. The number of resolution levels used is equal to 7. 20 iterations are performed at each level.

We present the results for the pair of images shown in Figure 6, which is called Office scene. Significant depth discontinuities occur in the scene. A sample of dense displacement field at level 1 is shown in Figure 7. Examing by flickering between two images on a Sun workstation, 95 percent of the vectors shown in Figure 7 appear to have no visible errors.

The parameters of the motion of the scene relative to the camera are shown in Table 1. The translation direction and rotation axis are represented by three components, (up, right, forward). The 3-D surface is plotted as the value of $1/(z)$, where z is the depth, in Figure 8. The occlusion map 1 is shown in Figure 9. The occlusion maps are to detect relatively large occluded regions (more than one pixel wide) and not to show occlusion boundaries which can be easily detected by analyzing discontinuities in the constructed depth maps. Since no attempt is made to obtain ground truth, we do not know the accuracy of those motion parameters. However, we can measure the discrepancies between the projection of the recovered 3-D position of the points and the observed projection (with rigid motion for the inferred structure). As shown in Table 1, the maximum image error,

$$\max_i \{ \| \mathbf{u}_i - \mathbf{h}_i(\mathbf{m},\mathbf{x}) \|, \ \| \mathbf{u}'_i - \mathbf{h}'_i(\mathbf{m},\mathbf{x}) \| \}$$

which indicates the maximum discrepancy between the observed projection and inferred projections, is about a third of the width of a pixel. Thus, the performance of the algorithm for motion and structure estimation is very good, and the image matching algorithm at least does not make large errors that violate rigidity constraint of underlying 3-D motion.

6 SUMMARY

An approach is presented for computing displacement fields between two images of a scene taken from different view points. The approach employs multiple attributes of the images to yield an overdetermined system of matching constraints. The algorithm does not require extensively textured images. It allows discontinuities and occlusions in the scene. From the matches obtained, dense 3-D depths and occlusion maps are computed for real world scenes, assuming the scene is rigid. The maximum discrepancy between the projection of the computed 3-D points and the matched image plane points (maximum image error) is about one third of the pixel width.

The investigation of theoretical bounds enables us to evaluate the intrinsic stability of motion estimation and the intrinsic limitation of optical flow based approaches. The conclusion is that with a relatively large disparity, the motion estimation problem is intrinsically fairly stable using the current popular video cameras. With Gaussian noise, our motion estimation algorithm has essentially reached the theoretical bound of the performance of any unbiased estimator. Simulations showed that the type of noise distribution does not significantly affect the performance of the algorithm and therefore, the objective of minimizing image errors can be used generally.

ACKNOWLEDGEMENTS

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

REFERENCES

[Cram46] H. Cramér, *Mathematical Methods of Statistics*. Princeton Univ. Princeton, New Jersey, 1946.

[Long81] H. C. Longuet-Higgins, A computer program for reconstructing a scene from two projections, *Nature*, vol. 293, Sept. 1981, pp 133-135.

[Rao73] C. R. Rao, *Linear Statistical Inference and Its Applications*, 2nd Ed., Wiley, New York, 1973.

[Tsai84] R. Y. Tsai and T. S. Huang, Uniqueness and estimation of 3-D motion parameters of rigid bodies with curved surfaces, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, No., 1, pp. 13-27, 1984.

[Weng87] J. Weng, T. S. Huang, and N. Ahuja, Error analysis of motion parameter determination from image sequences, in *Proc. the First International Conference on Computer Vision*, London, England, June 8-11, 1987.

[Weng88a] J. Weng, N. Ahuja, and T. S. Huang, Closed-form solution + maximum likelihood: a robust approach to motion and structure estimation, in Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June 5-9, 1988, pp. 381-386.

[Weng88b] J. Weng, N. Ahuja, and T. S. Huang, Two-view matching. in *Proc. 2nd International Conference on Computer Vision*, Florida, Dec. 1988.

[Zack71] S. Zacks, *The Theory of Statistical Inference*, Wiley, New York, 1971.

[Zhua84] X. Zhuang and R. M. Haralick, Rigid body motion and the optical flow image, in Proc *1st Conf. Artificial Intelligence Applications*, Denver, Colorado, Dec. 1984, pp. 366-375.

Table 1

Data and Results for the Office Scene			
Translation	-0.061160	0.981081	0.183682
Rotation axis	0.942139	-0.137486	0.305733
Rotation angle		1.112160°	
Maximum image error		0.000303	
Pixel width		0.000938	

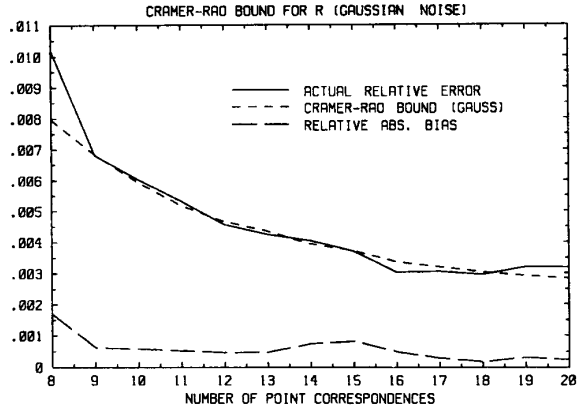

Figure 4(a)

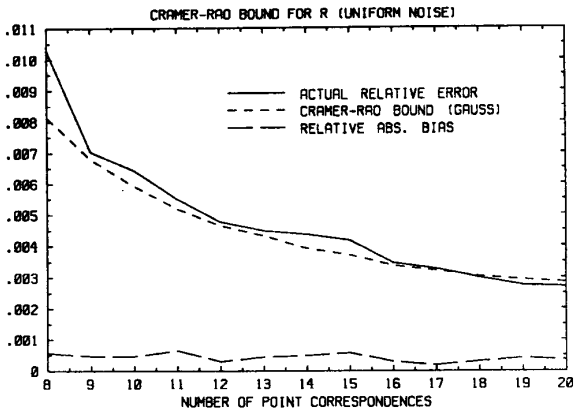

Figure 4(b)

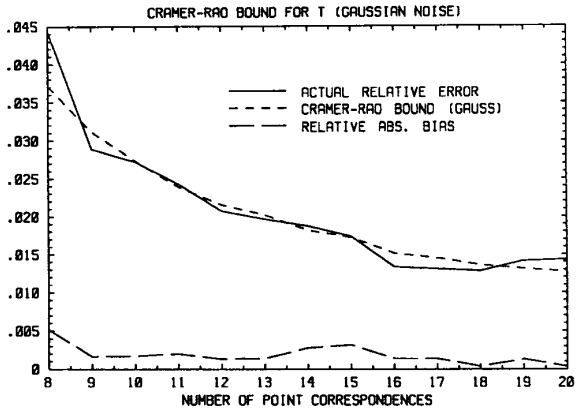

Figure 4(c)

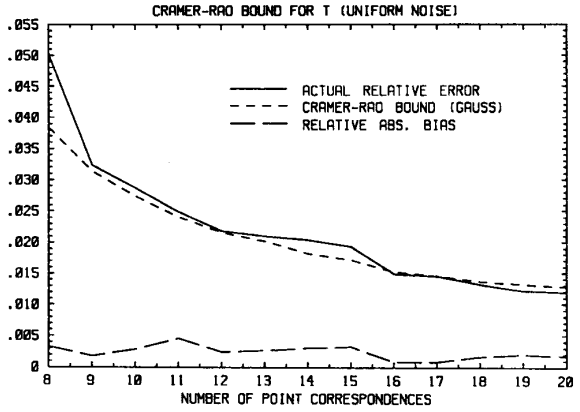

Figure 4(d)

Figure 4. Actual errors, Cramér-Rao bound for Gaussian noise, and the absolute bias of the estimator vs. number of point correspondences. Comparison for R: (a) Gaussian noise added; (b) uniform noise added. Comparison for T: (c) Gaussian noise added; (d) uniform noise added. Rotation axis: (1, 0.9, 0.8). Rotation angle: 5°. Translation: (0.5, –0.5, –3.0). 40 random trials.

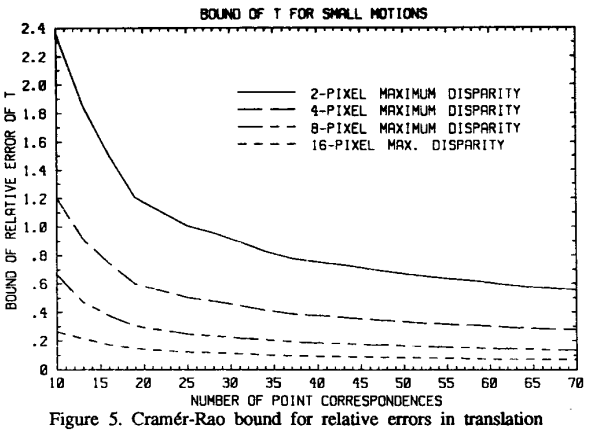

Figure 5. Cramér-Rao bound for relative errors in translation under small motions. 10 random trials. Translation: $(k, k, 0)$. The value of k is such that the maximum disparity caused by translation is d-pixels, d=2,4,8,16. Rotation axis: (1, 0.9, 0.8). Rotation angle: 5°. (Other simulations showed that the amount of rotation virtually does not affect the bounds presented here).

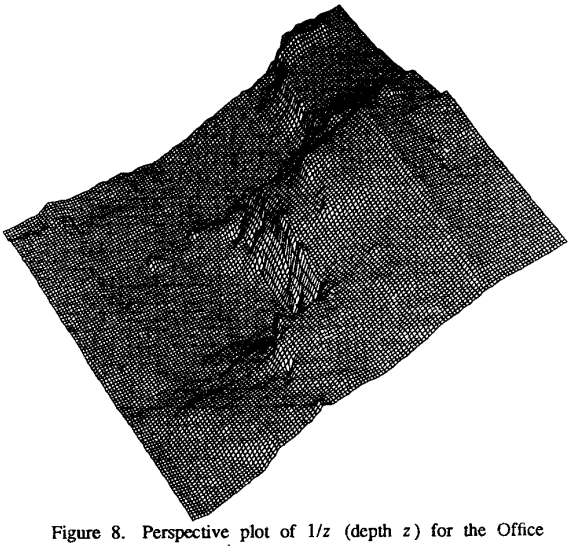

Figure 8. Perspective plot of $1/z$ (depth z) for the Office scene (from the viewpoint used for image 1).

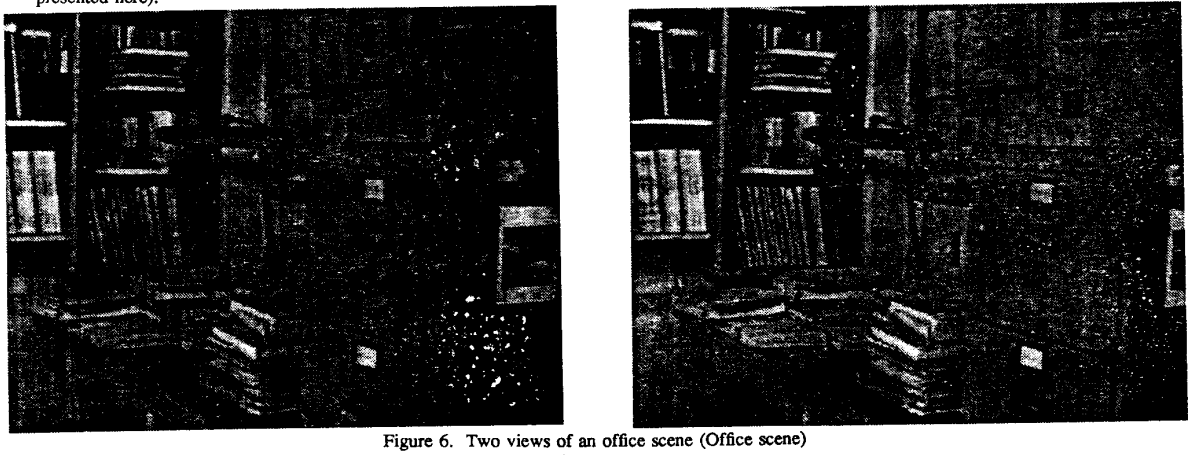

Figure 6. Two views of an office scene (Office scene)

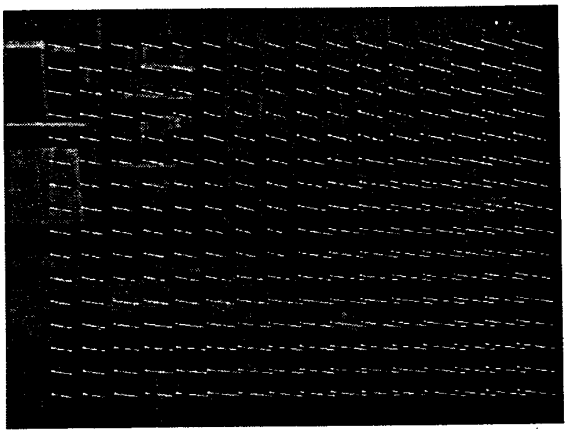

Figure 7. Samples of the computed displacement field at level 1 for the Office scene, superimposed on the blurred extended intensity image.

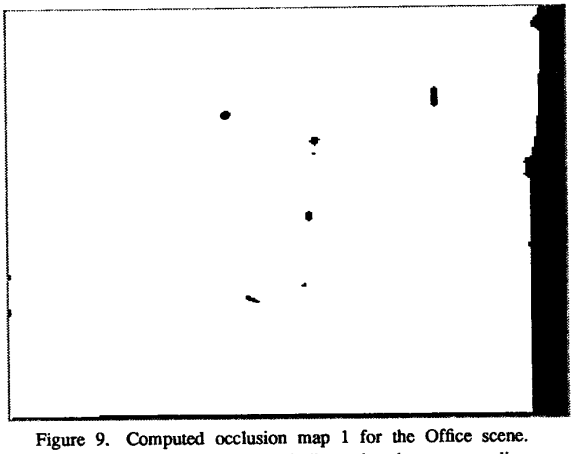

Figure 9. Computed occlusion map 1 for the Office scene. Black areas in occlusion map 1 indicate that the corresponding areas in image 1 (the first image in Figure 6) are not visible in image 2 (the second image in Figure 6).