TWO-VIEW MATCHING

JUYANG WENG, NARENDRA AHUJA, THOMAS S. HUANG

Coordinated Science Laboratory
University of Illinois, Urbana, IL 61801

Abstract

Establishing correspondences between images of the same scene is one of the most challenging and critical steps in motion and scene analysis. Part of the difficulty is due to a wide variety of three-dimension structural discontinuities and occlusions that occur in real world scenes. This paper describes a computational approach to image matching that uses multiple attributes associated with a pixel to yield a generally overdetermined system of constraints, taking into account possible structural discontinuities and occlusions. In the algorithm implemented, intensity, edgeness, and cornerness attributes are used in conjunction with the constraints arising from intraregional smoothness, field continuity and discontinuity, and occlusions to compute dense displacement fields and occlusion maps at pixel grids. A multiresolution multigrid structure is employed to deal with large disparities. Coarser level attributes are obtained by blurring the finer level attributes. The algorithms are tested on real world scenes containing depth discontinuities and occlusions. A special case of two-view matching is stereo matching where the motion between two images is known. The general algorithm given here can be easily specialized to perform stereo matching using epipolar line constraint.

1. INTRODUCTION

To estimate the 3-D motion and structure of objects from an image sequence, it is often necessary to establish correspondences between images. This paper presents an approach to matching two images of a scene that enforces similarity of matched multiple low level features as well as structural smoothness of the displacement field, while allowing for occlusions and discontinuities. This matching enables the analysis of motion parameters and structure of the scene from two or more images [Weng87].

Previous techniques for general two-view matching roughly fall into two categories: continuous and discrete.

(1) Continuous approaches. Though the approaches in this category compute image plane velocity field instead of performing explicit matching between features, the computed velocity field amounts to image matching. Optical flow is computed based on gradient of intensity function (e.g., [Horn81], [Nage86]) or spatiotemporal variation (e.g., [Heeg87]). The intensity constraint used by gradient based approaches is a linear equation in the two components of the velocity vector $(\alpha, \beta) \triangleq (\frac{du}{dt}, \frac{dv}{dt})$:

$$\frac{\partial i(u, v, t)}{\partial u}\alpha + \frac{\partial i(u, v, t)}{\partial v}\beta + \frac{\partial i(u, v, t)}{\partial t} = 0 \qquad (1.1)$$

This equation is insufficient to determine the two components of vector (α, β). A variety of smoothness constraints are proposed to solve this underdetermined problem. A typical one is minimizing $\iint \| \nabla\alpha \|^2 + \| \nabla\beta \|^2 du dv$ proposed by Horn and Schunck. Since this isotropic smoothness constraint is invalid across the image of occluding edges, Nagel and his colleagues introduce controlled smoothness constraint with the goal of smoothing along the edge direction at edge points, and smoothing isotropically at points having small gradient [Nage86]. This type of methods is commonly called (intensity) gradient based methods.

(2) Discrete approaches. The techniques of this category employ discrete features as tokens that are to be matched. Features used for matching include points, edges, lines, and other aspects of the scene structure.

Continuous approaches usually compute optical flow field along a pixel grid. There is no need for explicit feature extraction and matching. These approaches can potentially derive dense depth maps, however, they face the following problems: (1) The existing approaches resort to smoothness constraint to make the underdetermined problem solvable. When discontinuities occur in the velocity field, severe errors occur. (2) Since the motion is small, the magnitude of displacement vectors is also small. The results can be easily contaminated by pixel level perturbations. (3) The assumption that the intensity is constant for the same object patch in different images is not strictly true. (4) The approach needs well behaved and well textured intensity surfaces.

Discrete approaches allow either small motion or large motion corresponding to short or long range process. Therefore, accurate estimation of motion parameters and structure of the scene is possible under a relatively large motion. Discrete approaches do not suffer from the problem of varying image intensity for continuous approaches, since the existence of the discrete features is relatively more stable than intensity values. The intensity surfaces need not be smooth. However, these approaches also have problems: (1) It is very difficult to reliably match discrete features between two images. (2) Since the features are generaly sparse, only sparse depth data can be obtained. This makes it harder to estimate surfaces. (3) Features may be detected in one image but not in the other, e.g., due to occlusion. This creates more problems of mismatches. (4) To make matching possible, usually various smoothness constraints are used which may be invalid at occlusion and motion boundaries.

In this paper, we present a new approach to image matching that uses multiple attributes to yield an overdetermined system of matching constraints. This not only helps to combat noise in the images, but, more importantly, it also accommodates, to certain degree, possible changes in image intensity due to changes in viewing position, lighting, shading and reflection. A multi-resolution multi-grid computational structure is employed to deal with relatively large motions. We also address the problems of discontinuities in displacement field, and occlusion. The algorithms presented

CH2664-1/88/0000/0064$01.00 © 1988 IEEE

in this paper compute the displacement field along dense pixel grid as well as occlusion maps. To test the performance of matching, the motion parameters and the structure of the scene are computed [Weng88a] [Weng88b] from the matches obtained.

The next section discusses our approach. Section 3 discusses the algorithms. The experimental results are presented in Section 4. Section 5 presents concluding remarks.

2. AN APPROACH TO IMAGE MATCHING

It is desirable to have a dense distribution of matches to avoid artifacts in the estimates of scene structure. Since there may not be a large number of distinguishable feature points in the image, displacements across images, or matches, may have to be computed at pixels on a dense grid. We will use the term displacement field to refer to the result of image matching.

2.1 Image Attributes

Figure 10 shows a pair of monochrome images of a laboratory scene taken at two different positions. In general, the intensities of the corresponding points are not exactly the same in the two images, even under the same lighting conditions. They are generally close, except in some special cases, e.g., large reflection from a glossy surface. We call this *intensity similarity criterion.* However, this criterion does not uniquely determine the matching, even with the constraint that the displacement field is smooth in a region of the same intensity (see Figure. 1).

A much more reliable structural information for matching is sharp transitions of intensity — edges. As shown in Figure 1(b), the uncertainty is reduced if we match edges. The criterion that a given edge should be matched to another edge with similar edgeness measure is called *edgeness similarity criterion.*

However, intensity similarity and edgeness similarity are not sufficient to obtain matches. For example, when a closed contour is rotated as shown in Figure 2, the intensity similarity, edgeness similarity, and smoothness are not sufficient to determine the correct matches. Even if the variation of displacement field is minimized along the contour, the resulting displacement field is not correct [Hild83]. The problem here is that the similarity of contour shape is neglected. In general, the shape of the local intensity surface is a useful feature for matching. However, this shape varies significantly from image to image. Matching corner points unambiguously determines the displacement vector. In general, a right corner (the point at the apex of a right angle along an edge contour) should result in a high absolute measure of cornerness relative to corners of other angles. The sign of a corner should be such that it can distinguish a corner of a white rectangle on a black background from that of a black rectangle on a white background. Since the shape of iso-intensity contour is very unstable in a flat region where intensity gradient is small, the cornerness measurement at a flat point should be low. In other words, we should assign high cornerness measure only to those corners that are on edges. The criterion that a point should be matched to a point with similar cornerness measure is called *cornerness similarity criterion.*

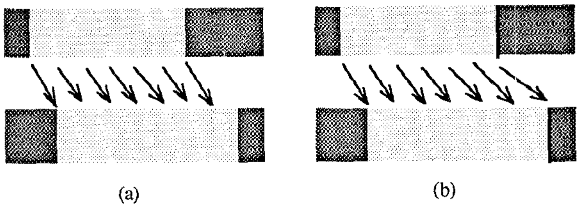

Figure 1. (a): A point can be incorrectly matched to any point with similar intensity. (b): The use of edges reduces the uncertainty in matching.

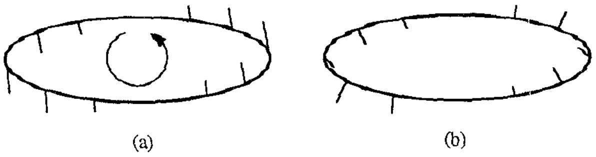

Figure 2. Intensity and edges are not sufficient to yield a correct match. (a): A closed contour is rotated. The thin line segments show true displacement. (b): The displacement vectors determined from local edge flow.

The algorithm described in this paper uses the intensity, edgeness and cornerness attributes for matching. The framework of our approach is such that additional attributes (e.g., color) could be easily included.

2.2 Relationships of Attributes

A corner point is isolated — it constitutes a zero dimensional point set. Matched corners constrain the displacement vector completely and without uncertainty. Edges usually form a contour — a one-dimensional point set. Locally, if a section of edge is matched with another edge, the displacement vector, starting from an edge point, can terminate at any point on the matched edge. This uncertainty is commonly referred to as the *aperture problem.* Similarly, a point can be matched to any point in a region having the same intensity (see Figure 1). This is a *two-dimensional aperture problem.*

Although matched corners completely determine the displacement vector, corners alone are not sufficient to determine the entire displacement field. First, we cannot guarantee that corners are available everywhere in images. Second, clusters of corners are difficult to match without additional support from other attributes. For similar reasons, corners and edges may not suffice without the intensity information.

Together, intensity, edgeness and cornerness attributes constrain the matching process and generally provide overdetermination to obtain matches. The overdetermination also provides a mechanism to accommodate small differences in attributes between two images.

2.3 Intra-regional Smoothness and Occlusion

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.

To correctly match two images, those scene regions which are occluded in one or the other image must be identified. 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 3,

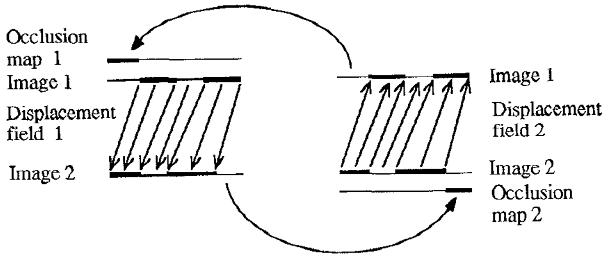

Figure 3. Determining occlusion maps (see text).

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 3, 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.

2.4 Multi-Resolution Multi-Grid Structure

Large disparities are crucial for general image matching, since matches may be spatially well separated. However, to find such matches 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. However, blurred intensity image has very few features left, and their locations are unreliable. Therefore, instead of blurring the image first and then measuring edgeness and cornerness, 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 4.

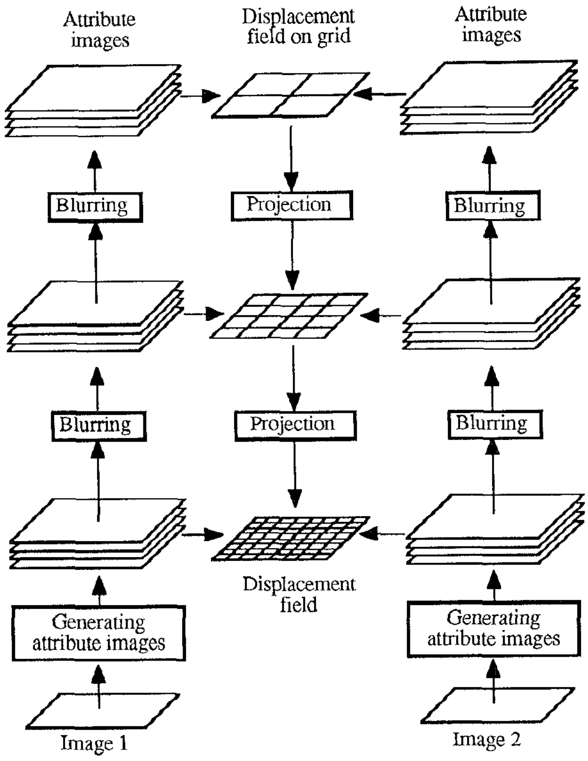

Figure 4. Computational structure and data flow

2.5 Limitations

It should be noted that our approach is not intended for situations where matching criteria involve image interpretation. For example, the approach will not always correctly match to satisfy high-level, context sensitive criteria, such as illustrated in Figure 5.

Another limitation of our approach is that corners may not always correspond to physical points in the scene. One reason is that two spatial lines that do not intersect in space may intersect in images. Since at coarse levels corners are blurred to contribute to texture measure, a small portion of non-physical corners or edges will not cause severe problems at coarse levels. The influence of the non-physical corners is expected to be overcome by other attributes and intra-regional smoothness. At finer levels, the weight for cornerness should be reduced since cornerness is not as reliable as edgeness and intensity and the strength of cornerness at finer levels begins to dominate the influence of intensities. A similar but slower reduction is performed for edgeness weights relative to the intensity weights.

3. ALGORITHM

In this section, 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.

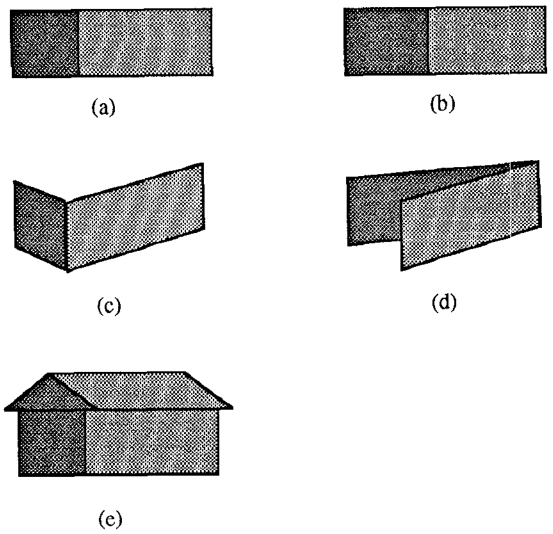

Figure 5. Ambiguity in finding displacement field for textureless surfaces. (a) and (b): Two views of two textureless surfaces. (c) and (d): Two different 3-D interpretations. (e): With more context, understanding of the scene helps to select the correct interpretation. The results of our algorithm will generally agree with the interpretation of (c). However, if the surfaces are textured, the cases (c) and (d) can be distinguished by the algorithm.

The intensity images are first filtered with a small (3 by 3) low pass filer to suppress gray level noise. The intensity is scaled linearly such that the minimum and maximum intensity values are equal to 0 and 255, respectively. The values of edgeness and cornerness discussed below are also normalized.

3.1 Edgeness and Cornerness

To get a continuous measure of edgeness, we use the magnitude of gradient. Some points need to be considered here. First, the magnitude of the gradient of the same scene point is not generally the same in the two images. Such differences may cause errors in the computed displacement field. Second, small gradients are more susceptible to intensity noise and are not reliable. Third, different scenes have different ranges of gradient magnitude. Therefore we need to normalize and transform the magnitude of gradient properly to edgeness. We define the edgeness measure range from 0 to 255. The magnitude of gradient is transformed by a normalization function f of the kind shown in Figure 6. It has two transition points x_0 and x_1. From x=0 to x=x_0, $f(x)$≈0 to suppress noise. From x=x_0 to x=x_1, $f(x)$ increases from ≈0 to ≈255 gradually and smoothly. The smooth transition interval $[x_0,x_1]$ allows continuous variation of edgeness for gradients of moderate magnitudes. For $x>x_1$, $f(x)\approx 255$, to limit strong edges and relatively enhance the moderate edges. The values of x_0 and x_1 are determined automatically through an analysis of the histogram of gradient magnitudes such that the fractions of the pixels in edgeness images that have values below $f(x_0)$ and above $f(x_1)$ are maintained at predetermined levels. This normalization function is important for the algorithm to adapt to different images. The *edgeness* is thus defined by

$$e(\mathbf{u}) = f(\nabla i(\mathbf{u})) \qquad (3.1)$$

where $\nabla i(\mathbf{u})$ is the gradient of intensity $i(\mathbf{u})$ at point **u** and f is one of the normalization functions shown in Figure 6.

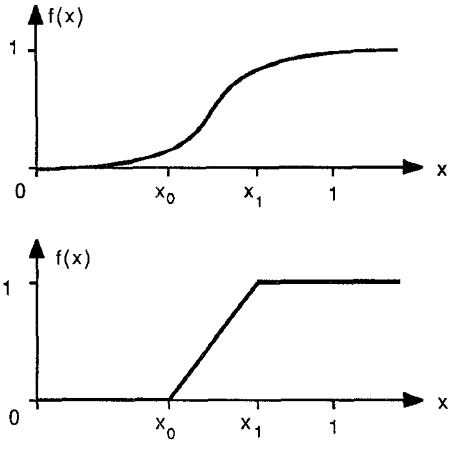

Figure 6. Two normalization functions for edgeness

We now define the cornerness measure at a point. We consider positive and negative cornerness separately. Roughly speaking, the cornerness at a point **u** measures the changes of the direction of gradient at two nearby points, weighted by the gradient at the point. These two points, $\mathbf{u}+\mathbf{r}_a$ and $\mathbf{u}+\mathbf{r}_b$ (see Figure 7) are located on a circle centered at **u**. The radius of the circle is determined by the level of resolution. We choose $\mathbf{r}_a$ and $\mathbf{r}_b$ such that the directional derivative along the circle reaches minimum and maximum values (see Figure 7). Let $\mathbf{a}=\nabla i(\mathbf{u}+\mathbf{r}_a)$, $\mathbf{b}=\nabla i(\mathbf{u}+\mathbf{r}_b)$, and $angle(\mathbf{a},\mathbf{b})$ be the angle from **a** to **b** measured in radians counter-clockwise, ranging from $-\pi$ to π. The closer the angle is to $\pi/2$, the higher the positive cornerness measure. In addition, the measure should be weighted by the magnitude of gradient at the point **u**. That is, the positive cornerness at **u** is defined by

$$p(\mathbf{u})=\begin{cases} e(\mathbf{u})(1-|angle(\mathbf{a},\mathbf{b})(2/\pi)-1|) & 0\le angle(\mathbf{a},\mathbf{b})\le\pi \\ 0 & otherwise \end{cases} \qquad (3.2)$$

Thus, the positive cornerness is normalized to range from 0 to 255. Similarly, if $angle(\mathbf{a},\mathbf{b})$ is negative, we have negative cornerness measure $n(\mathbf{u})$:

$$n(\mathbf{u})=\begin{cases} e(\mathbf{u})(1-|angle(\mathbf{a},\mathbf{b})(2/\pi)+1|) & -\pi\le angle(\mathbf{a},\mathbf{b})\le 0 \\ 0 & otherwise \end{cases} \qquad (3.3)$$

3.2 Orientation and Displacement Smoothness

Smoothness constraints impose similarity of the displacement vectors over a neighborhood. In addition to considering the smoothness of the overall displacement vectors, we separately consider the smoothness of the orientation of these vectors. The reason for emphasizing orientation smoothness 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:

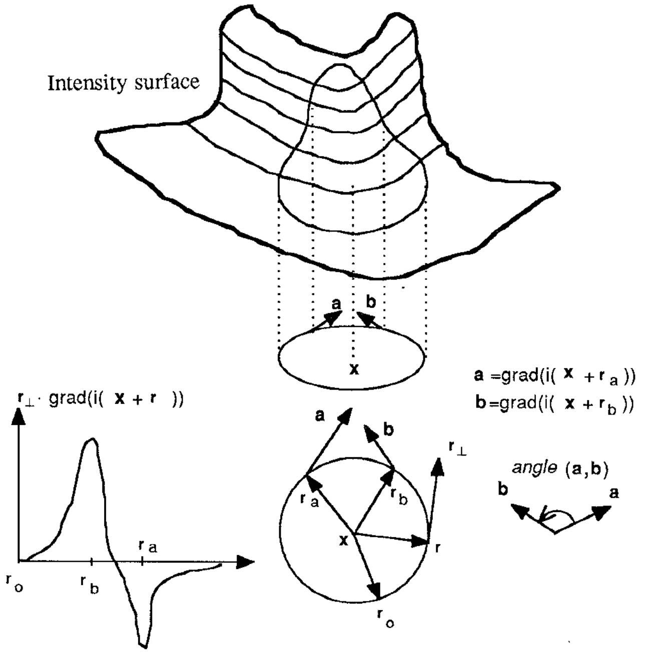

Figure 7. Definition of cornerness (see text).

$$\overline{\mathbf{d}}(\mathbf{u}_0)=\sum_{0<\|\mathbf{u}-\mathbf{u}_0\|<r} w(i(\mathbf{u})-i(\mathbf{u}_0),\ \mathbf{d}(\mathbf{u})-\mathbf{d}(\mathbf{u}_0))\ \mathbf{d}(\mathbf{u}) \qquad (3.4)$$

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 $\|\mathbf{u}-\mathbf{u}_0\|$. The objective that $\overline{\mathbf{d}}(\mathbf{u}_0)$ represent the neighboring displacement vectors of the region of $\mathbf{u}_0$ suggests the following requirements on the weight.

(1) The weight is large if intensity difference is small. We assume that small intensity difference is observed when two neighboring points $\mathbf{u}$ and $\mathbf{u}_0$ belong to the same region, and therefore, their displacement vectors should be similar.

(2) If $\mathbf{u}$ and $\mathbf{u}_0$ have similar intensity but the corresponding displacement vectors are different, the weight should remain small. This case occurs when the displacement field is projected from a coarse level to the finer level. Two adjacent points with the same intensity may take quite different initial displacement vectors if they belong to different grid points at the coarse level.

(3) If $\mathbf{u}$ and $\mathbf{u}_0$ have different intensities their displacement vectors are very different, the weight should be extremely small to suppress the influence of $\mathbf{u}$ on $\overline{\mathbf{d}}(\mathbf{u}_0)$.

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 that satisfies the above criteria is as follows:

$$w(\eta_i,\eta_d)=\frac{c}{\varepsilon+|\eta_i|(1+\|\eta_d\|^2)} \qquad (3.5)$$

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:

$$\sum_{0<\|\mathbf{u}-\mathbf{u}_0\|<r} w(i(\mathbf{u})-i(\mathbf{u}_0),\ \mathbf{d}(\mathbf{u})-\mathbf{d}(\mathbf{u}_0))=1 \qquad (3.6)$$

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 $\overline{\mathbf{d}}(\mathbf{u}_0)$. 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 (3.5) (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 $\overline{\mathbf{d}}(\mathbf{u}_0)$ allows us to perform matching using uniform numerical optimization despite the presence of discontinuities. This is discused below.

3.3 Minimizing Residuals

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:

$$r_i(\mathbf{u},\ \mathbf{d})=i'(\mathbf{u}+\mathbf{d})-i(\mathbf{u}) \qquad (3.7)$$

(2). Residual of edgeness:

$$r_e(\mathbf{u},\ \mathbf{d})=e'(\mathbf{u}+\mathbf{d})-e(\mathbf{u}) \qquad (3.8)$$

(3) Residual of positive cornerness:

$$r_p(\mathbf{u},\ \mathbf{d})=p'(\mathbf{u}+\mathbf{d})-p(\mathbf{u}) \qquad (3.9)$$

(4) Residual of negative cornerness:

$$r_n(\mathbf{u},\ \mathbf{d})=n'(\mathbf{u}+\mathbf{d})-n(\mathbf{u}) \qquad (3.10)$$

(4) Residual of orientation smoothness:

$$r_o(\mathbf{u},\ \mathbf{d})=\mathbf{d}(\mathbf{u})\times\overline{\mathbf{d}}(\mathbf{u})/\|\overline{\mathbf{d}}(\mathbf{u})\| \qquad (3.11)$$

(5) Residual of displacement smoothness:

$$r_d(\mathbf{u},\ \mathbf{d})=\mathbf{d}(\mathbf{u})-\overline{\mathbf{d}}(\mathbf{u}) \qquad (3.12)$$

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

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

$$+\lambda_n r_n^2(\mathbf{u},\ \mathbf{d})+\lambda_o r_o^2(\mathbf{u},\ \mathbf{d})+\lambda_d r_d^2(\mathbf{u},\ \mathbf{d})\} = \min \qquad (3.13)$$

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

$$\mathbf{r} \triangleq (r_i,\ r_e,\ r_p,\ r_n,\ r_o,\ r_d)^{\mathrm{T}} \qquad (3.14)$$

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 $\boldsymbol{\delta}_\mathbf{d}$. Expanding $\mathbf{r}(\mathbf{u},\ \mathbf{d}+\boldsymbol{\delta}_\mathbf{d})$ at $\boldsymbol{\delta}_\mathbf{d}=\mathbf{0}$, we have (suppressing variable $\mathbf{u}$ for conciseness):

$$\mathbf{r}(\mathbf{d}+\boldsymbol{\delta}_\mathbf{d}) = \mathbf{r}(\mathbf{d})+\frac{\partial\mathbf{r}(\mathbf{d})}{\partial\mathbf{d}}\boldsymbol{\delta}_\mathbf{d}+o(\|\boldsymbol{\delta}_\mathbf{d}\|) \triangleq \mathbf{r}+J\boldsymbol{\delta}_\mathbf{d}+o(\|\boldsymbol{\delta}_\mathbf{d}\|) \quad (3.15)$$

where

$$J=\frac{\partial \mathbf{r}(\mathbf{d})}{\partial \mathbf{d}}=\begin{bmatrix} \frac{\partial i'}{\partial u} & \frac{\partial i'}{\partial v} \\ \frac{\partial e'}{\partial u} & \frac{\partial e'}{\partial v} \\ \frac{\partial p'}{\partial u} & \frac{\partial p'}{\partial v} \\ \frac{\partial n'}{\partial u} & \frac{\partial n'}{\partial v} \\ -\bar{d}_u / \| \bar{\mathbf{d}} \| & \bar{d}_v / \| \bar{\mathbf{d}} \| \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.16}$$

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

$$\Lambda = diag\{1, \lambda_e, \lambda_p, \lambda_n, \lambda_o, \lambda_d\} \tag{3.17}$$

We want to find $\delta_{\mathbf{d}}$ such that the sum of squared residuals in (3.13) at the point is minimized. Neglecting high order terms and minimizing $\| \Lambda(\mathbf{r}+J\delta_{\mathbf{d}}) \|$, from (3.15) we get the formula for updating **d**:

$$\delta_{\mathbf{d}} = -(J^{\mathrm{T}}\Lambda^2 J)^{-1} J^{\mathrm{T}}\Lambda\ \mathbf{r}(\mathbf{u}) \tag{3.18}$$

The partial derivatives in the entries of J 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 mask to compute finite differences is shown in Figure 8, where spacing s 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.

3.4 Recursive Blurring

The structure of the computation and data flow is shown in Figure 4. The original images are first filtered by a 3×3 low pass filter to remove gray level noise. Then four attribute images pairs are generated (intensity, edgeness, positive cornerness and negative cornerness). The attribute images are extended in four directions to provide context for the points that are near the image border. The extension is made by repeating the border row or column. We use recursive blurring (to be specified below) to speed up computation. Only integer summations and a few integer divisions are needed to perform such a simple blurring. 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 $s{\times}s$ (See Figure 9). 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.

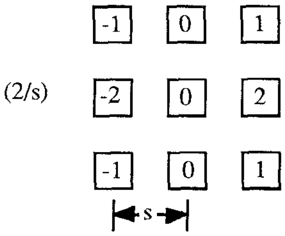

Figure 8. Mask for computing derivatives.

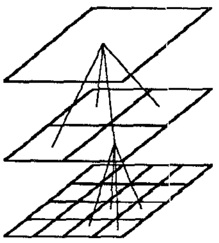

Figure 9. Recursive blurring and limiting (see text).

3.5 Motion and Depth from Displacement Fields

To test the performance of the matching algorithm, motion and structure parameters are estimated from the matches obtained, assuming the scene is rigid. First, the matching algorithm is applied to compute displacement from image 2 to image 1, from which occlusion map 1 is computed. The occlusion maps are filtered by 3×3 median filters to remove single-pixel occlusion regions and noise due to pixel grid. Then, the displacement from image 1 to image 2 is computed using occlusion map 1. The occluded points $\mathbf{u}_0$ are assigned the vector $\bar{\mathbf{d}}(\mathbf{u}_0)$ in (3.4) as the displacement vector. Algorithms to compute motion parameters and the 3-D position of the points ([Weng88a], [Weng88b]) can be used to compute the motion between two images and the depth of the scene (since we have very dense displacement field!). The algorithm first solve for motion parameters and 3-D positions of the points using a closed-form solution [Weng88a], assuming the scene is rigid. The results are then optimized [Weng88b] so that the discrepancies between the projection of the computed 3-D structure and the observed projection is minimized.

The matching algorithm can be easily modified to solve the problem of stereo matching. For example, for horizontal epipolar line matching, v=0. The minimization problem in (3.13) then becomes a one variable problem.

4. EXPERIMENTAL RESULTS

Experiments have been conducted on a variety of real world scenes. A CCD monochrome video camera with roughly 512 by 512 resolution 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.

First, we present the results for the pair of images shown in Figure 10, which is called Mac scene. Significant depth discontinuities occur in the scene. Books and Manuals lie irregularly on the table. Such a surface is very difficult to estimate accurately from sparse depth data. The largest disparity is about 80 pixels for this pair of 512 by 512 images. The two edgeness images are shown in Figure 11. The two positive cornerness images are shown in Figure 12. Blurred attribute images are shown in Figure 13. A sample of dense displacement field at level 1 is shown in Figure 14. Examing by flickering between two images on a Sun workstation, 95 percent of the vectors shown in Figure 13 appear to have no visible errors. The occlusion map 1 is shown in Figure 15. 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.

The algorithms presented in [Weng88a] and [Weng88b] are employed to compute the motion parameters and the 3-D structure of the scene from the computed displacement field. The 3-D surface is shown as the value of $1/(z)$, where z is the depth, as intensity image in Figure 16 and is plotted in Figure 17. Those surfaces agree fairly well with those observed in the real scene. It is worth mentioning that the complicated surfaces on the manuals and books on the front table are well recovered. 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). 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. Let us define the (standard) *image error* as

$$image\ error = \sqrt{\sum_{i=1}^{n}(d_i^2+d_i'^2)/2n} \qquad (4.1)$$

where n is the number of points (number of displacement vectors) in an image, d_i is the distance between the projection of the computed 3-D point i and its observed projection on image 1, and d'_i is analogous distance for image 2. If the displacement field is not correct, i.e., it does not correspond to the motion of a rigid scene, the image error will be large no matter how good the performance of the algorithm for motion and structure estimation is. On the other hand, if the errors in the displacement field do not violate the rigidity constraint (rigidity constraint means that the field should correspond to the motion of a rigid scene), the image error can still be small provided the performance of the motion and structure estimation algorithm is good. As shown in Table 1, the image error is within half of the pixel width. Thus, the performance of the algorithm for motion and structure estimation is good, and the matching algorithm at least does not make large errors that violate rigidity constraint. The displacement field could still conceivably make systematic errors so as to depict a rigid scene, although different than the real one. But such systematic errors are unlikely except those caused by the existence of multiple interpretations shown in Figure 5 (all those interpretations satisfy the rigidity constraint).

To compare our algorithm with one that uses only intensity gradients, we set the the weights for edgeness and cornerness, λ_e, λ_p and λ_n to zero. The resulting depth from the matches is shown in Figure 18, which can be seen to contain many errors. The result without using occlusion map is shown in Figure 19, from which we can see severe errors occur around the occluded regions. Figure 20 shows two images of another scene (called Chair scene) and the samples of the corresponding displacement field at level 1. The results of motion estimation are given in Table 2.

5. SUMMARY AND DISCUSSION

An approach is presented to compute displacement field 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 continuities and discontinuities in displacement field, and occlusion are taken into account to analyze complicated real world scenes.

Table 1

Data and Results for the Mac Scene			
Translation	0.016152	0.990948	0.133270
Rotation axis	0.966355	0.175804	-0.187752
Rotation angle		1.611214°	
Image error		0.000326	
Pixel width		0.000938	

Table 2

Data and Results for the Chair Scene			
Translation	0.094865	-0.057348	0.993837
Rotation axis	0.708815	0.406920	0.576193
Rotation angle		0.125879°	
Image error		0.000316	
Pixel width		0.000938	

In the current implementation of the algorithm, intensity, edgeness, and cornerness are used as matching attribute. The algorithm does not require extensively textured images. It allows discontinuities and occlusions in the scene. From the matches obtained, dense 3-D surface and occlusion maps are computed for real world scenes, assuming the scene is rigid. The discrepancy between the projection of the computed 3-D points and the matched image plane points (standard image error) is about one half of pixel width.

To compare the algorithm presented with others, let us first make some observations about the role of J given in (3.16). The top four rows of J determine the matching, and the bottom three rows determine the intra-regional smoothness. At a point of image $i'(\mathbf{u})$ where there are strong transitions of intensity, edgeness and cornerness, or subset of them, the first four rows of J are relatively strong and determine the optimal δ_d to update displacement vector. The bottom three rows are relatively week and are used to adjust the intraregional uniformity of the field in the neighborhood. At a point where the intensity, edgeness and cornerness are flat, the top four rows of J are weak and the three bottom rows play a major role. The displacement is updated such that it is consistent with the neighboring displacement vector of the same region. The first four linear equations of

$$\mathbf{r}(\mathbf{d}+\delta_d) = \mathbf{r}(\mathbf{d})+\frac{\partial \mathbf{r}(\mathbf{d})}{\partial \mathbf{d}}\delta_d = 0 \qquad (3.19)$$

yield four linear equations in term of two components of δ_d, which determine four lines in the space of δ_d. Since the measurements are noisy, those lines are unreliable and generally do not intersect at a single point. A weighted least squares solution determines a point that minimizes the weighted sums of squared residuals.

Existing gradient-based methods use only one linear equation based on intensity similarity. Namely, only the first of the four lines is used. This line does not determine a point in the plane (an underdetermined system). The methods resort to smoothness constraint. However, many incorrect solutions that satisfy the intensity constraint can also be very smooth, and very often can be even smoother than the correct solution. In other words, there is a huge class of solutions that satisfy, numerically, both the linear equation and the smoothness constraint but may be very different from the correct solution. The final solution obtained by these (usually iterative) methods can be any one in this class. Therefore, these methods do not give correct solution in general. This partially accounts for the problems of gradient methods.

In our approach, the system is generally overdetermined and smoothness is used mainly for filling in uniform regions. Though the available information for matching is just intensity images, the matching criteria here are based on not only individual intensity values, but also relationships between those intensity values.

Edgeness and cornerness characterize some meaningful local relationships at a point. These attributes provide additional information that is needed to guide the matching. More importantly, they lead to a generally overdetermined system based solely on attribute matching, instead of regularization (smoothness). Such overdetermination significantly improves the stability of the solution. Smoothness plays a major role only inside uniform regions.

Since intensity, edgeness and cornerness used in our algorithm are point-based local properties, the algorithm is pixel oriented: simple, uniform and easy to implement on certain parallel architectures. This is an advantage over symbolic matching approaches that use high level primitives and provide only sparse matches.

ACKNOWLEDGEMENTS

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

REFERENCES

[Heeg87] D. J. Heeger, Optical flow from spatiotemporal filters, in Proc. *Inter. Conf. Computer Vision*, London, England, June, pp. 181-190, 1987.

[Hild83] E. C. Hildreth, *The Measurement of Visual Motion*, MIT press, Cambridge and London. 1983.

[Horn81] B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artificial Intelligence*, 17 (1981) pp 185-203.

[Nage86] H. -H. Nagel and W. Enkelmann, An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, No. 5, 1986, pp. 565-593.

[Weng87] J. Weng, T. S. Huang, and N. Ahuja, 3-D motion estimation, understanding and prediction from noisy image sequences, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, No. 3, 1987, pp. 370-389.

[Weng88a] 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.*.

[Weng88b] 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.

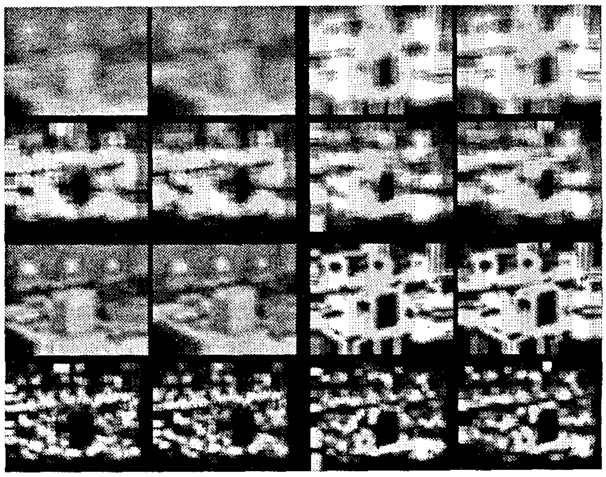

Figure 13. Blurred attribute images. Upper two rows: level 6; Left pair in the first row: blurred intensity image pair; Right in the first row: blurred edgeness image pair; Left in the second row: blurred positive cornerness image pair; Right in the second row: blurred negative cornerness image pair; Lower two rows: level 5 in the same arrangement as the upper two rows.

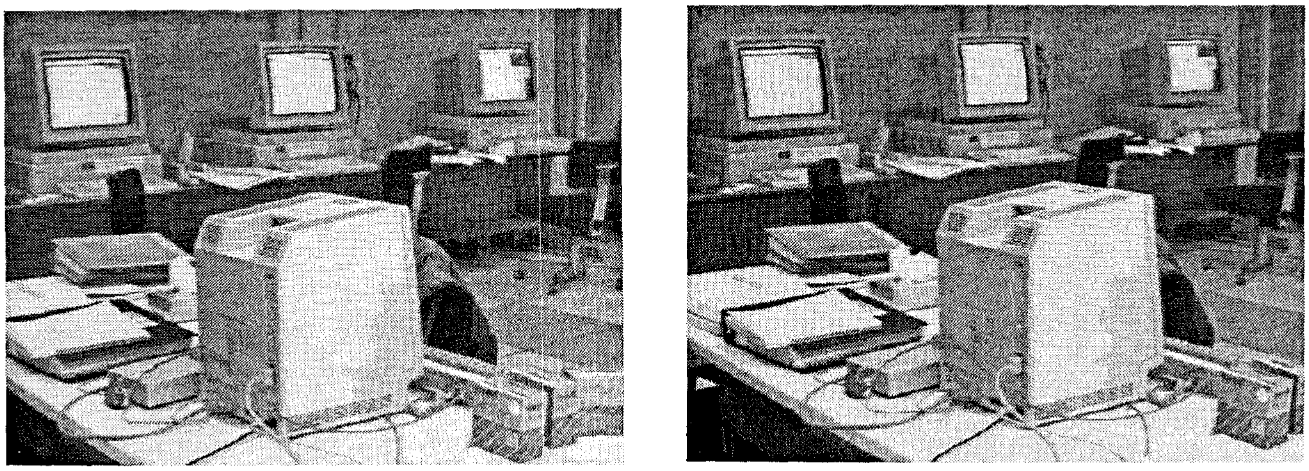

Figure 10. Two views of a laboratory scene (the Mac scene)

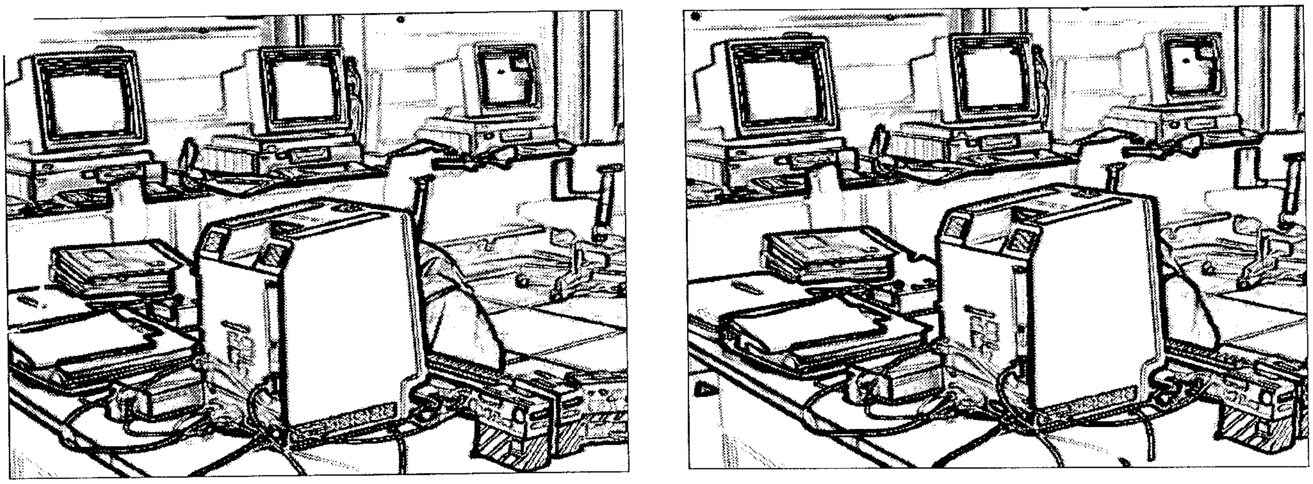

Figure 11. Two edgeness images for the Mac scene.

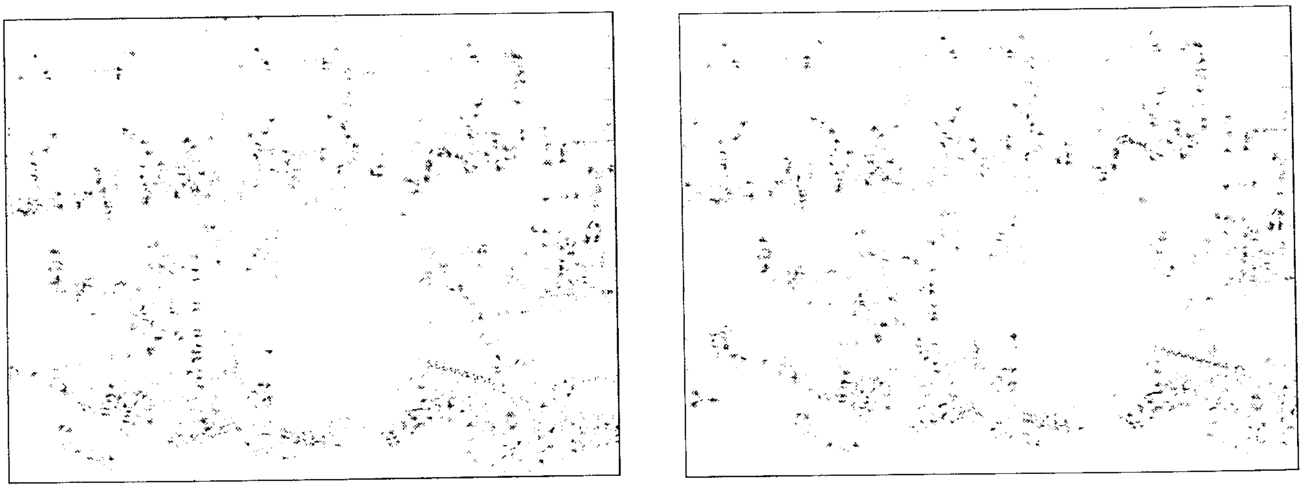

Figure 12. Two positive cornerness images for the Mac scene.

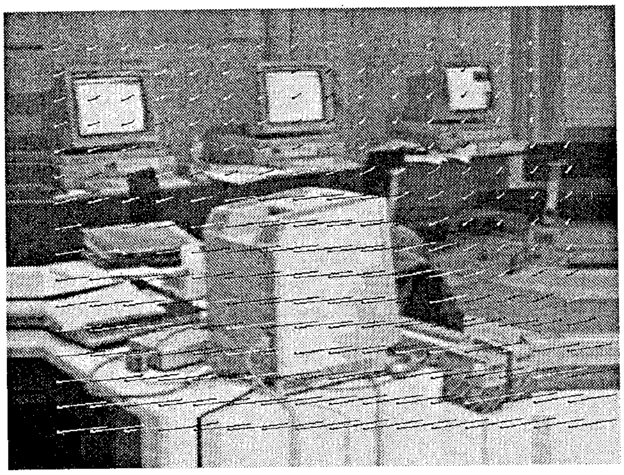

Figure 14. Samples of the computed displacement field at level 1 for the Mac scene, superimposed on the blurred extended intensity image.

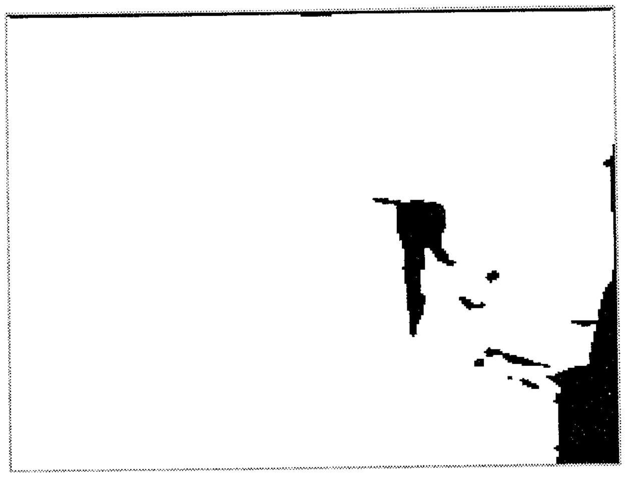

Figure 15. Computed occlusion map 1 for the Mac scene. Black areas in occlusion map 1 indicate that the corresponding areas in image 1 (the first image in Figure 1) are not visible in image 2 (the second image in Figure 1).

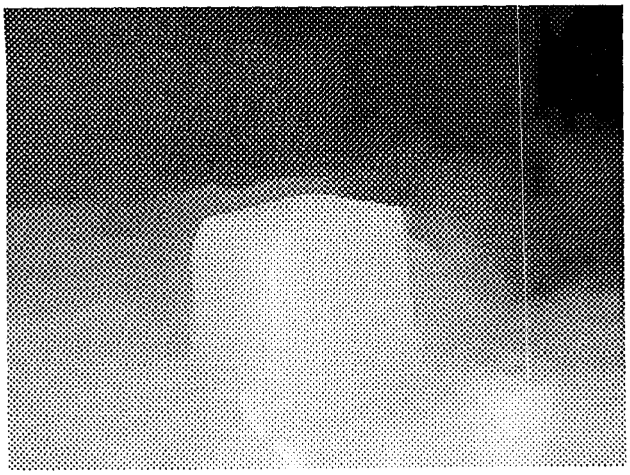

Figure 16. The computed 3-D surface ($1/z$) shown as intensity image for the Mac scene (from the viewpoint used for image 1).

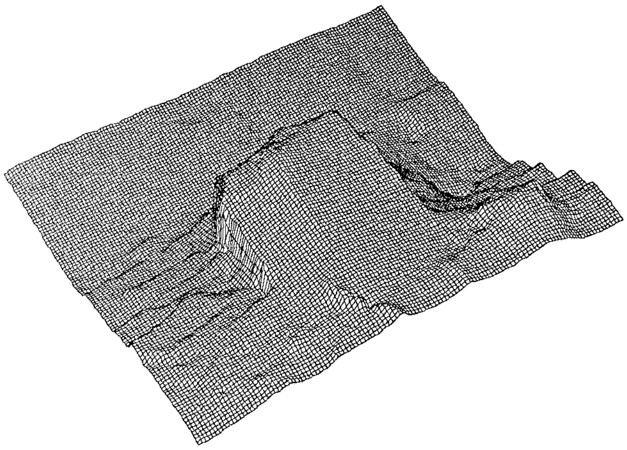

Figure 17. Perspective plot of computed 3-D surface for the Mac scene (from the viewpoint used for image 1).

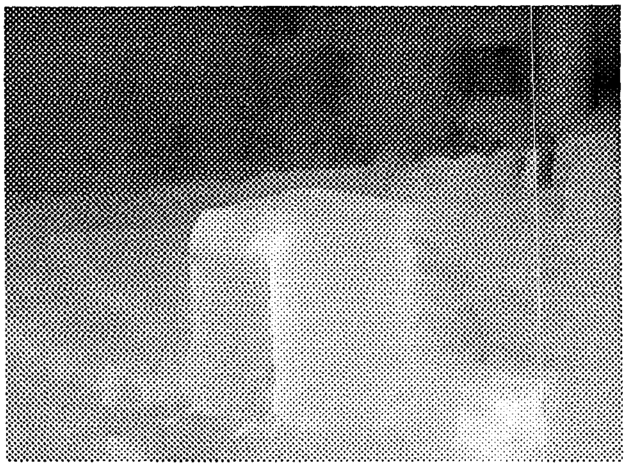

Figure 18. Performance using only intensity: The computed 3-D surface ($1/z$) shown as intensity image for the Mac scene (from the viewpoint used for image 1).

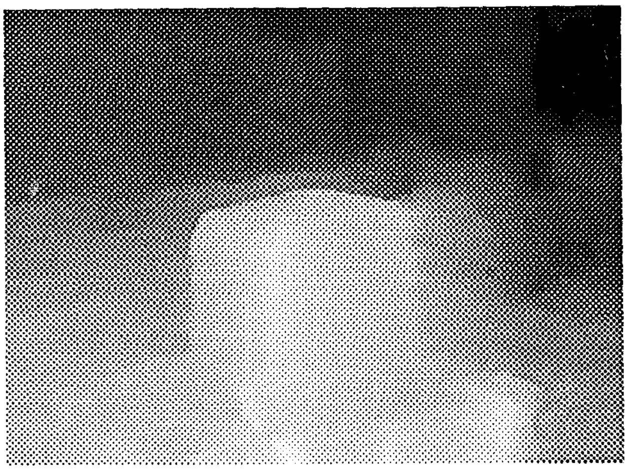

Figure 19. Performance without identifying occlusion regions: The computed 3-D surface ($1/z$) shown as intensity image for the Mac scene (from the viewpoint used for image 1).

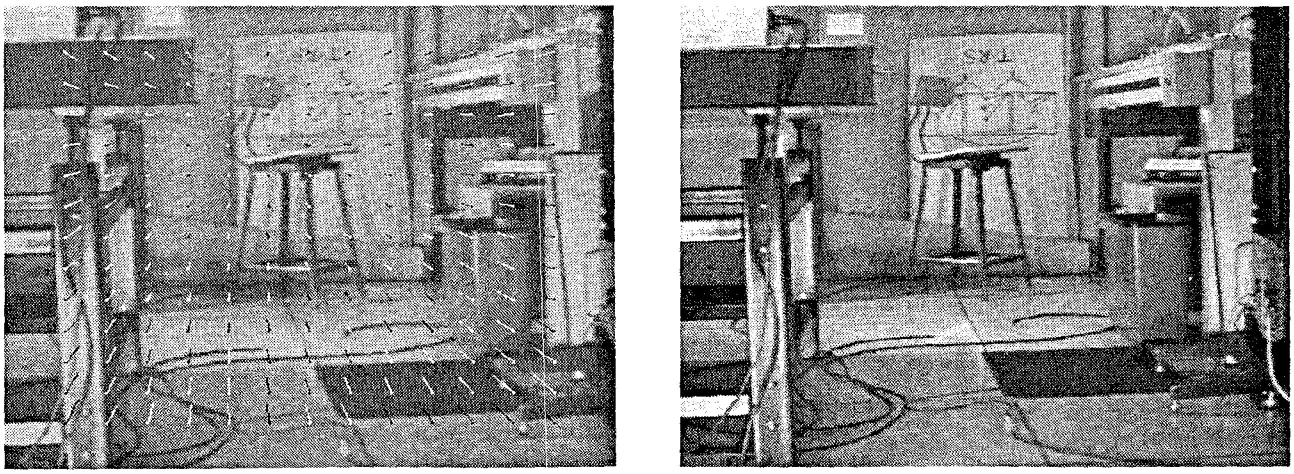

Figure 20. Two images of the Chair scene, and samples of the computed displacement field at level 1 from the first image.