Matching Two Perspective Views

Juyang Weng, *Member, IEEE*, Narendra Ahuja, *Fellow, IEEE*, and Thomas S. Huang, *Fellow, IEEE*

Abstract—**Establishing correspondences between different perspective 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 3-D 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 each image point 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 along the pixel grids. The intensity, edgeness, and cornerness are invariant under rigid motion in the image plane. In order to cope with large disparities, a multiresolution multigrid structure is employed. Coarser level edgeness and cornerness measures are obtained by blurring the finer level measures. The algorithm has been tested on real-world scenes with depth discontinuities and occlusions. A special case of two-view matching is stereo matching, where the motion between two images is known. The algorithm can be easily specialized to perform stereo matching using the epipolar constraint.**

Index Terms—**Dynamic scene analysis, motion estimation, optical flow, stereo matching, structure from motion, two-view matching.**

I. Introduction

TIME-VARYING images of real-world scenes can provide kinematical, dynamical, and structural information of the world. To estimate from the image sequences the 3-D motion and structure of objects, it is often necessary to establish correspondences between images, i.e., to identify in the images the projections that correspond to the same physical part of the sensed scene. 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.

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

1) Continuous Approaches: Although the approaches in this category compute the image velocity field instead of performing explicit matching between features, the computed velocity field amounts to image matching. Each velocity vector approximates the correspondence between two points in different images. In practice, an optical flow field, which is the field derived from sensed optical projections of the scene, is used to approximate the actual image plane velocity field with small magnitudes. The techniques in this category typically need the condition that the interframe motion is small and the intensity function is smooth and well behaved. The optical flow may be computed based on the spatiotemporal variation of the image intensity function [16], [29], [1], [24], [17] [13]. Although the flow of intensity is not exactly the same as the projection of 3-D velocity [16], [28], they are considered to be similar under some conditions. From the assumption that the intensity of a point is conserved from image to image, a linear equation in the two components of the velocity vector $(\alpha, \beta) \overset{\Delta}{=} (\frac{du}{dt}, \frac{dv}{dt})$ at the point can be derived:

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

This equation alone is insufficient to determine the two components of the vector (α, β). A variety of smoothness constraints is proposed to solve this underdetermined problem. A typical one is minimizing $\int\int ||\nabla\alpha||^2 + ||\nabla\beta||^2 du dv$ proposed by Horn and Schunck [16]. Since this isotropic smoothness constraint is inappropriate across the image of occluding edges, Nagel and Enkelmann introduced some controlled smoothness constraints with a goal of smoothing along an edge direction at edge points and smoothing isotropically at points having small spatial gradient [24]. These types of methods are commonly called (intensity) gradient-based methods. Another method is based on spatiotemporal filters [1], [13]. A family of Gabor-energy filters tuned to different spatial orientations and temporal frequencies are applied to a dense image sequence. Image velocity (α, β) is determined by minimizing the difference between the measured motion energies (from the response of Gabor filters) and those predicted for a pattern with a flat power spectrum.

2) Discrete Approaches: The techniques in this category treat the images as temporal samples taken at discrete times and select discrete features that are to be matched. Points with high intensity variation are often used as the matching features [27], [4], [9]. Other features used for matching include closed contours of zero crossings of Laplacian-of-Gaussian images to compute velocity field [14], edges for stereo matching [21], [22], [11], [25], [15], straight lines for stereo matching [3], correlation of intensity patterns [10] [2], and other aspects of the scene structure [12], [20].

Continuous approaches usually compute optical flow field along a pixel grid. There is no need for explicit feature extraction and matching. The matching is performed through

Manuscript received September 18, 1989; revised December 23, 1991. This work was supported by the National Science Foundation under grants ECS-83-52408 and IRI-86-05400. Part of this work was presented at the International Conference on Computer Vision, 1988. Recommended for acceptance by Associate Editor J. Mundy.

The authors are with the Beckman Institute, University of Illinois, Urbana, IL 61801.

IEEE Log Number 9200109.

numerical minimization. These approaches can potentially derive dense depth maps. However, they face the following problems:

1. The existing approaches resort to a smoothness constraint to make the underdetermined problem solvable. When discontinuities occur in the velocity field, severe errors occur.
2. Since the interframe motion is restricted to be small, the magnitudes of flow vectors are also small (usually within a couple of pixels). Therefore, the velocities computed can be easily overridden by pixel-level perturbations. Such a flow with very low SNR limits the inherent stability of motion analysis [31].

   Moreover, contrary to a common belief that the interframe disparity is small in a video rate sequence, one often must deal with large disparities even if a complete video rate sequence is used. For example, suppose a mobile robot travels at a typical human walking speed (6 km/hr) on which a CCD video camera mounted with an f=16 mm normal lens is aimed to the side scene; then, in the video rate sequence (30 frames/s) taken by the camera, the interframe disparity of the objects at 2 m away is around 30 pixels. With such a disparity, most optical flow estimation algorithms that assume small interframe disparities will fail. Camera pan actions often result in much larger interframe disparities (whereas an electronic shutter can guarantee a sharp frame).
3. The assumption that the intensity of the same object patch is constant in different images is not strictly true.
4. Well-behaved and well-textured intensity images are required for the techniques to be applicable.

Discrete approaches allow either small motion or large motion, corresponding to short range process and long range process, respectively [7], [8]. Accurate estimation for the 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 with continuous approaches since the existence of the discrete features is relatively more stable than intensity values. Moreover, the intensity surfaces need not be smooth. However, the approaches in this category also have problems:

1. It is difficult to reliably match discrete features because a feature cannot be easily distinguished from others. There exists a large number of potential candidates for matching, and no powerful scheme is available to select the correct one.
2. Since the features are generally sparse, only sparse depth data can be obtained. This makes it harder to estimate surfaces. Usually, surface interpolation has to be performed to give a complete surface from the sparse depth data. However, the interpolated surfaces are often quite different from the real surfaces.
3. Features may be detected in one image but not in the other, e.g., due to occlusion. These create more problems of mismatches.
4. To make the matching possible, various smoothness constraints are usually used, which may be invalid at occlusion and motion boundaries. Because the features are sparse, the lack of neighboring information makes the detection of discontinuities and occlusions very difficult.

In this paper, we present a new approach to image matching that takes advantages of both continuous and discrete approaches. Our method is characterized by the following features:

A. Discrete features are represented by their values in the corresponding attribute images, and these attributes are blurred to different resolution levels to provide information needed for matching. Therefore, although the method is based on discrete features, it has essentially avoided the problem of inconsistency of feature detection between different frames.

B. Multiple attributes associated with the images are fully employed to yield an overdetermined system of matching constraints. This helps to combat noise and accommodates, to a certain degree, slight changes in image intensity due to changes in viewing position, lighting, shading, and reflection. Futher, and more importantly, the displacement vector is completely determined by those matching constraints.

C. A mechanism is incorporated into the approach to deal with uniform nontextured object surfaces that are often present in real-world images. We also present a simple and effective method to cope with discontinuities of displacement field. In fact, handling nontextured surfaces and preserving disparity discontinuities are two challenging issues in image matching.

D. Although the difficult problem of occlusion has been largely ignored in the literature, it cannot be ignored here due to the presence of large disparities. The algorithm presented in this paper introduces a technique of computing the occlusion maps. The occluded regions are marked to prevent mismatching. Our matching algorithm has been tested on images of real-world scenes with significant occlusions.

E. Our method can deal with large disparities. The algorithm has been tested with disparities as large as over 80 pixels. As we mentioned above, large image disparities are very important to the accuracy of the estimated motion and structure of a moving scene and are often present in video rate image sequences. In our method, a multiresolution multigrid computational scheme is employed to deal with large disparities. Various coarse-to-fine strategies have been used by a number of researchers for stereo matching (a partial list includes [21], [23], [26], [11], [5], [15]). In our work, the multiresolution multigrid scheme makes the computation of large disparity reliable and efficient, whereas it is integrated with the use of multiple attributes, the detection of disparity discontinuities, and the computation of occluded regions.

The results computed by our matching algorithm have been used by our motion and structure estimation algorithm, and consequently, we are able to compute dense depth maps of real-world scenes under large unknown motions. The computational approach and the experimental results presented in this paper serve to bridge the gap between the point-based methods (which assume point correspondences) and the image inputs and indicate, to certain degree, whether structure from motion is possible in real-world situations.

The next section presents our approach to image matching. Section III proceeds with formal definitions, analyses, and details of the algorithm. Section IV discusses some further refinements. The experimental results are shown in Section V. Finally, Section VI presents a summary and discussion.

II. An Approach to Image Matching

Formally, two images I and I′ are defined by two functions: $i : U \rightarrow B$ and $i' : U \rightarrow B$, where $U \subset R^2$, and $B \subset R$ for monochrome images ($B \subset R^3$ for color images). U defines the image plane. Functions i and i' map each point in the image plane to an intensity value. Without loss of generality, we assume that the image intensity is given in a finite normalized image plane

$$U = \{(u, v)^{\mathrm{T}} \mid 0 \leq u \leq 1, 0 \leq v \leq 1\}$$

with intensity in a normalized range $B = \{i \mid 0 \leq i \leq 255\}$ (for monochrome images). In a camera-centered coordinate system, a time-varying 3-D scene Π at time t is a collection of measurable sets in R^3, $\Pi \subset R^3$, and similarly $\Pi' \subset R^3$ is the 3-D scene at time t'. A motion is a mapping from Π to Π': $\boldsymbol{m} : \Pi \rightarrow \Pi'$ (although a term "displacement" or "transformation" may be more appropriate here, we still use the term "motion" since it is a conventional term for image sequence analysis). A motion $\boldsymbol{m}$ is 1 to 1 and onto (1 to 1 correspondence). The projection of a point $\boldsymbol{x} = (x, y, z)^{\mathrm{T}}$ in a scene Π is defined by $\boldsymbol{p} : \Pi \rightarrow R^2$ such that

$$\boldsymbol{p}(\boldsymbol{x}) = (x/z, y/z)^{\mathrm{T}} \tag{2.1}$$

which corresponds to the perspective projection of a pinhole camera with a normalized unit focal length [30]. Since the universe is unlimited, we assume for any image point $\boldsymbol{u}$, there will always be some point in the scene whose projection is $\boldsymbol{u}$. Namely, the projections of Π and Π' cover entire image plane U: $U \subset \boldsymbol{p}(\Pi)$, $U \subset \boldsymbol{p}(\Pi')$.

A point $\boldsymbol{x} \in \Pi$ is visible if and only if $\boldsymbol{p}(\boldsymbol{x}) \in U$ and there is no $\boldsymbol{y} \in \Pi$, $\|\boldsymbol{y}\| < \|\boldsymbol{x}\|$, such that $\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{y})$. An occlusion map O for image I consists of those image points in U whose corresponding 3-D point $\boldsymbol{x} \in \Pi$ is visible, but the moved point $\boldsymbol{m}(\boldsymbol{x})$ is not visible. Similarly, we define O' as the occlusion map for image I′.

An image matching from I to I′ is a mapping $\boldsymbol{\kappa} : U \rightarrow U$ that satisfies the following. For any $\boldsymbol{u} \in U - O$ (the symbol $-$ denotes set subtraction), letting its corresponding 3-D point be $\boldsymbol{x}$, then $\boldsymbol{\kappa}(\boldsymbol{u}) = \boldsymbol{p}(\boldsymbol{m}(\boldsymbol{x}))$. Similarly, we define $\boldsymbol{\kappa}'$ as the matching from I′ to I. Notice that the mapping from an occluded point $\boldsymbol{u} \in O$ is arbitrary, and therefore, it can be assigned, e.g, according to some smoothness constraint. The displacement field is defined by $\boldsymbol{d} = \boldsymbol{\kappa} - \boldsymbol{e}$, where $\boldsymbol{e}$ is an identity mapping $\boldsymbol{e}(\boldsymbol{u}) = \boldsymbol{u}$ for all $\boldsymbol{u} \in R^2$. Therefore, the matched image point for $\boldsymbol{u}$ is $\boldsymbol{u}' = \boldsymbol{\kappa}(\boldsymbol{u}) = \boldsymbol{u} + \boldsymbol{d}(\boldsymbol{u})$. We will use the term "displacement field" to refer to the result of image matching.

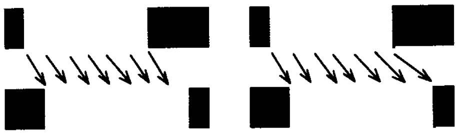


Fig. 1. Matching based on intensity is not sufficient to determine matching: (a) Point can be matched to any point with a similar intensity; (b) use of edges reduces the uncertainty in matching.

A. Image Attributes

What information do we need to compute the displacement field from two images? A type of attribute associated with images is motion invariant if its value does not change from image to image under any motion. Motion invariant attributes are desirable since the conservation of such attributes for the matched image points can be used as a criterion for matching process. Unfortunately, there exist no motion invariant attributes under general situations. We instead proceed to search for motion-insensitive attributes, i.e., those attributes that generally sustain only small changes under motion. In fact, the intensity is motion insensitive under some appropriate conditions (e.g., with matte surfaces and extended light sources). Fig. 13 shows a pair of monochrome images of a laboratory scene taken at two different positions. From those two images, it can be seen that the corresponding regions have similar intensity values. We call this *intensity similarity criterion.*

If the matching is based on intensity only, a point can be matched to any point with the same or similar intensity. A 1-D example is shown in Fig. 1(a). The problem is more severe in 2-D images, where a point can be potentially matched to a wide region with similar intensities.

In fact, the information used for image matching by human vision is not just individual unrelated points with intensity values. What is matched over time by human vision is the structure of images [27], [8], i.e., the spatial relationships among image points. Higher level structural information, e.g., shape of region and spatial relationships between regions is obviously useful for matching. However, this type of information is very unstable under motion and occlusion. For example, the shape of a 3-D region may change significantly if they are viewed from different positions due to foreshortening. We are interested in low-level structural attributes that are insensitive to motion and involve only a very small neighborhood around an image point so that occlusions will not cause significant match errors in a large area.

A candidate for structural information for matching relates to sharp transitions of intensity—edges. As shown in Fig. 1(b), if we match two edges and tolerate a small difference in intensity, the resulting matching is correct. Edges are motion-insensitive since sharp intensity transition will generally remain a sharp transition after a moderate amount of motion. 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 often not sufficient to determine matches. For example, if a closed contour is rotated as shown in Fig. 2, the intensity

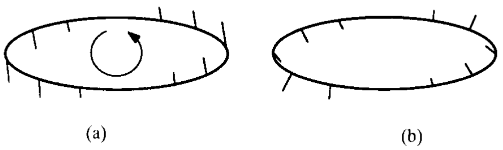


Fig. 2. Intensity and edges are not sufficient to yield a correct match: (a) Closed contour is rotated. The "needles" show true displacement vectors; (b) displacement vectors determined from local edge flow. Even if the variation of displacement field is minimized along the contour, the resulting displacement field is still not correct (see [14]).

similarity, edgeness similarity, and smoothness are not sufficient to determine the correct match. Even if the variation of displacement field is minimized along the contour, the resulting displacement field is still not correct [14]. The problem here is that the similarity of the contour shape is neglected. Corners or high curvature points indicate the shape of a contour. Matching corner points unambiguously determine 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 compared with that of the 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 an iso-intensity contour is very unstable in a flat region where intensity gradient is small, the cornerness measurement at a flat region should be low. In other words, we should assign high cornerness measure only to those corners that are on edges. An example of the formal definition of the cornerness will be given in Section III-D. The criterion that a point should be matched to a point with similar cornerness measure is called the *cornerness similarity criterion*.

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.

B. Relationships of the Attributes

We have discussed intensity, edgeness, and cornerness as the primary attributes for matching. These attributes measure different properties of the local intensity surfaces associated with an image point.

A corner point is isolated; it constitutes a zero dimensional point set. Matched corners constrain the displacement vector completely. Edges usually form a contour: a 1-D point set. Locally, if a section of edge is matched with another edge, the displacement vector, starting from an edge point, can be terminated 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. This is a *2-D 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. Fig. 3 shows an example, where without edge and intensity information, the corners tend to be mismatched.

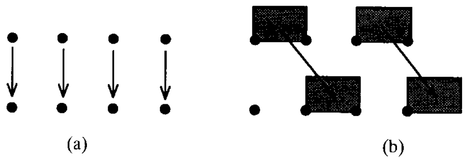


Fig. 3. Without edge and intensity information, corner points tend to be mismatched: (a) Without intensity and edge information; (b) with intensity and edge information.

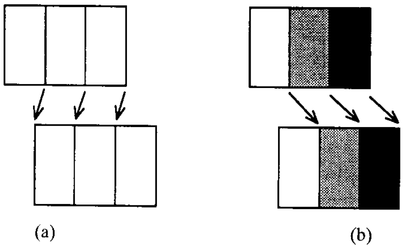


Fig. 4. Without intensity information, edges tend to be mismatched, although, in principle, edge contours can be matched globally. Intensity gives information that makes matching more reliable and easier: (a) Without intensity information; (b) with intensity information.

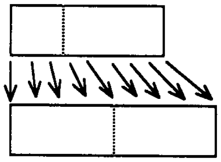


Fig. 5. Sheet of white paper on top of another. Neither the boundary between the sheets (shown dashed) nor the texture of the paper is visible. When the top sheet slides along the bottom sheet, the motion is perceived as a stretching of the union of both sheets.

For similar reasons, corners and edges may not suffice without the intensity information (see Fig. 4).

Together, intensity, edgeness, and cornerness attributes constrain the matching process and generally provide an overdetermination for image matching.

C. Intraregional Smoothness and Occlusion

In order to deal with real-world images, we must be able to handle regions with uniform intensity (see Fig. 13). 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 the *intraregional smoothness criterion*. The objective of this criterion is to fill displacement information into those areas where no significant intensity variation occurs. However, we cannot impose smoothness across different regions.

Obviously, this intraregional smoothness assumption is not always satisfied. Fig. 5 gives such an example, from which one can see that visual information is not always enough to correctly infer physical phenomena. The *intraregional smoothness criterion* may be consistent with what we perceive but may be inconsistent with the reality.

To correctly match two images, those scene regions that 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 the field of view in the other image. If the occluded regions are not detected, they may be incorrectly matched to nearby regions, interfering with the correct matching of these regions. To identify the occluded 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 (see Fig. 6, where black areas denote the occluded 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 in Fig. 6) into parts of image 1 (e.g., the right-most section in Fig. 6). This generally will not affect the computation of the occlusion map 1 since the occluded regions of image 1 may only occur on the opposite side across the "jammed" region (in Fig. 6, e.g., the occluded region of image 1 is to the right of a "jammed" region). Those regions in image 1 that have not been matched (in Fig. 6, no arrows pointing to them) are occluded in image 2 and are therefore marked in the occlusion map 1 (black in Fig. 6). These unmatched patches may also be located at the center of the images if they are occluded by other parts of the scene. Once the 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 (see Fig. 6).

Formally, for any image point $\boldsymbol{u} \in U$, there is a scene point $\boldsymbol{x} \in \Pi$ such that $\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{u}$. From the definition of $\boldsymbol{\kappa}$ and $\boldsymbol{\kappa}'$, it is clear that $\boldsymbol{\kappa}$ and $\boldsymbol{\kappa}'$ are 1 to 1 correspondences from $U - O$ to $U - O'$ and from $U - O'$ to $U - O$, respectively. Therefore, the occlusion map O can be determined by $O = U - \boldsymbol{\kappa}'(U - O')$, and similarly, $O' = U - \boldsymbol{\kappa}(U - O)$. However, this procedure is recursive. Once one occlusion map is determined, the other can also be determined. The procedure outlined in Fig. 6 used preliminary $\boldsymbol{\kappa}'$ that is computed to determined O without information about O'. Since regions in O and O' are generally far apart, this preliminary $\boldsymbol{\kappa}'$ may be good enough to determine O.

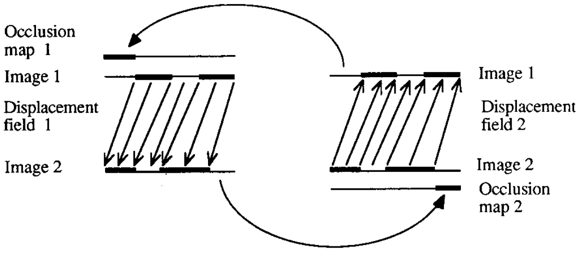


Fig. 6. One-dimensional illustration of determining occlusion maps (see text). Images are represented by lines as 1-D images. The displacement fields shown just illustrate the correspondences between two 1-D images and not as actual displacement fields.

D. Multiresolution Multigrid Structure

To find matches over a large disparity 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, a 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 the occurrence of the corresponding features or to the texture content of the original images. Although 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 applies 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 applies to both textured or nontextured 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, and a low sample rate suffices. A coarse grid also helps to speed up the propagation of results within uniform regions. 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 projection-and-refinement procedure 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 Fig. 7.

E. Limitations

It should be noted that our approach is not intended for situations where matching criteria involve image interpretation. In some cases, lack of texture in the surface makes correct matching impossible unless high-level knowledge is used.

Another limitation of our approach is that the criteria for the similarity of matching attributes may be violated in some situations. For example, corners may not always correspond to a physical point: Two lines that do not intersect in 3-D space may intersect in the images, and the corner arising from such an intersection may correspond to different scene points as the viewing position changes. Since, at coarse levels, corners are blurred to contribute to texture measure, a limited presence of nonphysical corners or edges at coarse levels is expected to be overcome by other attributes: intraregional smoothness and occlusion information. 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

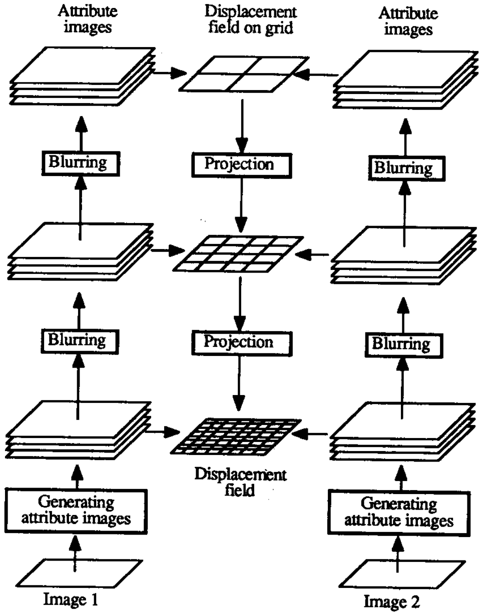


Fig. 7. Computational structure and the data flow.

similar but slower reduction is performed for edgeness weights relative to the intensity weights for the same reasons.

III. Analysis and Algorithm

The approach presented in Section II is general in the sense that the actual definitions of attributes and details of the computational steps can vary considerably. In this section, we will introduce some exact definitions and perform analysis to provide some insight into the approach. Details of the implemented algorithm will be discussed as well.

A. Locally Rigid Motion

In Section II, the motion $\boldsymbol{m}$ is required to be a 1 to 1 correspondence between two scenes so that inverse motion exists to define matching $\boldsymbol{\kappa}'$. For the approach to perform well, the motion has to be such that the assumptions for the similarity of attributes and the intraregional smoothness are approximately true. Let us investigate one type of motion that can be locally modeled by piecewise rigid motion. In other words, we examine those motions that may be nonrigid globally and may involve individually moving objects, but locally, each object does not deform drastically.

Suppose that the visible scene $\Pi \subset R^3$ can be partitioned into a finite number of subsets $\Pi = \{\Pi_i\}$ such that each Π_i has continuously differentiable visible surfaces. We define a type of motion called locally rigid motion for each of such surfaces.

Definition 1: A motion $\boldsymbol{m} : \Pi \rightarrow \Pi'$ is *locally rigid* if for any nonboundary point $\boldsymbol{x}_0$ on a continuously differentiable surface of Π_i, there is a positive number $\delta > 0$ such that for any $\boldsymbol{x}$ with $\|\boldsymbol{x} - \boldsymbol{x}_0\| < \delta$, $\boldsymbol{m}(\boldsymbol{x})$ can be expressed by

$$\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{m}(\boldsymbol{x}_0) + R(\boldsymbol{x}_0)\{\boldsymbol{x} - \boldsymbol{x}_0\} + o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)$$

or

$$\boldsymbol{m}(\boldsymbol{x}) = R(\boldsymbol{x}_0)\boldsymbol{x} + \boldsymbol{T}(\boldsymbol{x}_0) + o(\|\boldsymbol{x} - \boldsymbol{x}_0\|) \qquad (3.1)$$

where

$$R(\boldsymbol{x}_0) = \left.\frac{\partial \boldsymbol{m}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0}$$

is a rotation matrix, $\boldsymbol{T}(\boldsymbol{x}_0) = \boldsymbol{m}(\boldsymbol{x}_0) - R(\boldsymbol{x}_0)\boldsymbol{x}_0$, and $o(v)$ denotes a term that satisfies $\lim_{v\to 0}\{o(v)/v\} = 0$. Similarly, we define locally rigid motion in 2-D space R^2.

A rigid motion of Π_i is a special case of the locally rigid motion in which $R(\boldsymbol{x}_0)$ and $\boldsymbol{T}_{\boldsymbol{x}_0}$ are constant (does not depend on $\boldsymbol{x}_0$), and the higher order term $o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)$ is exactly equal to zero. In a locally rigid motion, the motion in a small neighborhood around a point is a rigid motion if the higher order term $o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)$ is neglected. However, the global motion may still be significantly different from the rigid motion because the definition only restricts the infinitesimal behavior of the motion. Since $R(\boldsymbol{x}_0)$ and $\boldsymbol{T}(\boldsymbol{x}_0)$ may vary with $\boldsymbol{x}_0$, the nonzero term $o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)$ allows significant deviation from a rigid motion globally.

B. 3-D Motion and Image Plane Motion

The motion we will study in the following arises from surfaces where each undergoes a locally rigid motion as discussed above. We need to relate the 3-D motions to the image plane. Obviously, the projection of the 3-D rigid motion onto the image plane is not, in general, a 2-D rigid motion. In the following, we investigate to what degree the image plane displacement that corresponds to a 3-D locally rigid motion can be locally approximated by a locally rigid image plane motion.

According to (3.1), the locally rigid 3-D motion in the neighborhood of a point $\boldsymbol{x}_0$ can be represented by

$$\boldsymbol{x}' = R\boldsymbol{x} + \boldsymbol{T} + o(\|\boldsymbol{x} - \boldsymbol{x}_0\|) \qquad (3.2)$$

where $\boldsymbol{x}' = \boldsymbol{m}(\boldsymbol{x})$, and R and $\boldsymbol{T}$ depend on $\boldsymbol{x}_0$. As defined in (2.1), let the perspective projection of $\boldsymbol{x} = (x, y, z)^{\mathrm{T}}$ and $\boldsymbol{x}' = (x', y', z')^{\mathrm{T}}$ be $\boldsymbol{u}$ and $\boldsymbol{u}'$, respectively. Equation (3.2) gives

$$\boldsymbol{u}' = \{z/z'\}\{R_{11}\boldsymbol{u} + \boldsymbol{R}_{12} + \boldsymbol{T}_1/z + o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)/z\} \qquad (3.3)$$

where R_{11} is the 2×2 upper left submatrix of R, $\boldsymbol{R}_{12}$ is the 2×1 upper right submatrix (vector) of R, and $\boldsymbol{T}_1$ consists of the first two components of $\boldsymbol{T}$. In general, the submatrix R_{11} is not a rotation matrix. For a rotation about unit vector $\boldsymbol{n}$ by a small angle θ, however, the rotation matrix can be approximated by

$$R \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix}$$

where $(\alpha, \beta, \gamma) = \theta\boldsymbol{n}$. Therefore

$$R_{11} \approx \begin{bmatrix} 1 & -\gamma \\ \gamma & 1 \end{bmatrix}$$

which approximates a 2-D rotation with an angle γ. In other words, under a small rotation, R_{11} can be approximated by a

2-D rotation matrix M. In fact, a more careful analysis leads to the conclusion that R_{11} can be approximated by a 2-D rotation matrix M as long as the rotation is small about the x and y axes, and it can still have a relatively large rotational component about the Z axis.

The term z/z' in (3.3) represents a scaling. If the change in depth $|z' - z|$ is significantly smaller than the depth itself (i.e., $|z' - z|/z \ll 1$, which is usually the case), then

$$z/z' = 1/\{1 + \{z' - z\}/z\} \approx 1.$$

Using the above two approximations, we can rewrite (3.3) by

$$\boldsymbol{u}' \approx M_{11}\boldsymbol{u} + \boldsymbol{V} + o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)/z$$

where $\boldsymbol{V} = \boldsymbol{R}_{12} + \boldsymbol{T}_1/z$ depends on the depth of the point $\boldsymbol{x}$. In a small neighborhood of $\boldsymbol{x}_0$, if the depth difference $|z - z_0|$ is small compared with z_0 (i.e., $|z - z_0|/z_0 \ll 1$), we have

$$z = z_0\{1 + \{z - z_0\}/z_0\} \approx z_0.$$

Then, $\boldsymbol{V}$ can be approximated by $\boldsymbol{V}_0 \triangleq \boldsymbol{R}_{12} + \boldsymbol{T}_1/z_0$. Equation (3.3) is approximated by

$$\boldsymbol{u}' \approx M_{11}\boldsymbol{u} + \boldsymbol{V}_0 + o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)/z_0 \tag{3.4}$$

where M_{11} is a rotation matrix depending only on $\boldsymbol{x}_0$, and $\boldsymbol{V}_0$ is a vector that also depends only on $\boldsymbol{x}_0$. If we can prove

$$\lim_{\boldsymbol{u} \to \boldsymbol{u}_0} \frac{o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)}{z_0\|\boldsymbol{u} - \boldsymbol{u}_0\|} = 0 \tag{3.5}$$

we can rewrite (3.4) by

$$\boldsymbol{u}' \approx M_{11}\boldsymbol{u} + \boldsymbol{V}_0 + o(\|\boldsymbol{u} - \boldsymbol{u}_0\|). \tag{3.6}$$

To obtain (3.5), we assume that in a small neighborhood of $\boldsymbol{x}_0$, the depth of the surface z can be expressed as a continuously differentiable function of image coordinate vector $\boldsymbol{u}$:

$$z = f(\boldsymbol{u}). \tag{3.7}$$

This implies that the viewing line is not tangential to the surface at $\boldsymbol{x}_0$ so that the surface does not degenerate into a curve infinitesimally. Using (3.7), the 3-D point $\boldsymbol{x}$ on the surface can be represented as a function of $\boldsymbol{u}$:

$$\boldsymbol{x} = f(\boldsymbol{u})\begin{bmatrix}\boldsymbol{u}\\ 1\end{bmatrix}.$$

Therefore, $\boldsymbol{x}$ is continuously differentiable with respect to $\boldsymbol{u}$. According to the law of the mean, there exists a positive number k such that

$$\|\boldsymbol{x} - \boldsymbol{x}_0\| \le k\|\boldsymbol{u} - \boldsymbol{u}_0\|.$$

For all $\boldsymbol{u}$ in a sufficiently small neighborhood of $\boldsymbol{u}_0$, it follows that

$$\begin{aligned}\lim_{\boldsymbol{u} \to \boldsymbol{u}_0} \frac{o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)}{z_0\|\boldsymbol{u} - \boldsymbol{u}_0\|} &= \lim_{\boldsymbol{x} \to \boldsymbol{x}_0} \frac{o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)}{\|\boldsymbol{x} - \boldsymbol{x}_0\|} \frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|}{z_0\|\boldsymbol{u} - \boldsymbol{u}_0\|}\\ &\le \frac{k}{z_0} \lim_{\boldsymbol{x} \to \boldsymbol{x}_0} \frac{o(\|\boldsymbol{x} - \boldsymbol{x}_0\|)}{\|\boldsymbol{x} - \boldsymbol{x}_0\|}\\ &= 0.\end{aligned}$$

This proves (3.5). Equation (3.6) concludes that the projection of a locally rigid 3-D motion onto the image plane can be approximated by a locally rigid 2-D image plane motion in a small neighborhood of any nonboundary point $\boldsymbol{x}_0$, provided 1) change in depth due to the motion is small compared with the original depth; 2) the rotation does not have large components about the x and y axes; 3) surface depth variation in the small neighborhood of $\boldsymbol{x}_0$ is small compared with the depth of $\boldsymbol{x}_0$; 4) the surface can be represented by a continuously differentiable function of image coordinates in a neighborhood of $\boldsymbol{x}_0$. Since those conditions are usually fairly well satisfied within each piece of smooth surface, we expect that the projection of locally rigid motion of piecewise smooth scene surface can be approximately represented by the piecewise locally rigid image plane motions.

C. Motion-Insensitive Image Attributes

We have discussed the relationships between the 3-D piecewise locally rigid motion and image plane motion. Even though the image plane motion can be generally approximated by a piecewise locally rigid motion, the intensity of an image is influenced by a series of factors in a complicated way. For example, the measured intensity of an object point varies due to changes in geometry between light sources and objects (motion between the object and the light sources), in geometry between objects and image sensors (motion between the sensors and the object), in optical attenuation (e.g., lens peripheral attenuation), and in photo-electrical sensitivity of image sensors (e.g., inconsistency in CCD sensor array) [6], [18], [28]. The factors related to the structure of image sensors can be controlled in manufacturing or calibrated so that they are well compensated for. However, the geometry among lighting, objects, and image sensors cannot be completely controlled in general. We must exclude specular or glossy surfaces from consideration since a slight change in lighting direction or viewing direction will significantly change the apparent brightness of such surfaces. With extended lighting sources, matte surfaces do not drastically change their apparent brightness if slight changes in the geometry among lighting, objects, and image sensors occur. Fine micro surface structures of those surfaces constitute a macro structure that reflects light relatively uniformly in all directions. For image matching in this paper, we will only be concerned with these types of surfaces. In other words, we assume the object surfaces are such that the image intensity of object surfaces varies only slightly between two images.

Let the corresponding image positions of a point in two images be $\boldsymbol{u}$ and $\boldsymbol{u}'$, respectively, and the corresponding intensity be $i(\boldsymbol{u})$ and $i'(\boldsymbol{u}')$, respectively. The above assumption can be expressed by

$$i'(\boldsymbol{u}') = i(\boldsymbol{u}) + s_i(\boldsymbol{u})$$

where $s_i(\boldsymbol{u})$ is a small value.

We introduce image plane motion-invariant attributes. First, we need to extend the domain of an image; an intended image i is an element in $S = \{i \mid R^2 \rightarrow B\}$. S is the set of all possible infinite images. Let $h = fg$ denote a product of two mappings f and g: $h(\boldsymbol{u}) = f(g(\boldsymbol{u}))$. A 2-D rigid motion of images is a mapping $m : S \rightarrow S$ such that $mi = i_m$, where $i_m(\boldsymbol{u}) = i(R_2\boldsymbol{u} + \boldsymbol{T}_2)$, R_2 is a 2×2 rotation matrix, and $\boldsymbol{T}_2$ is a 2-D vector. Namely, the 2-D rigid motion m is a mapping that maps an infinite image to another rotated and translated image. Now, we are ready to define the planar rigid motion invariant (PRMI) operator.

Definition 2: A PRMI operator $g : S \rightarrow S$ is a mapping from S to S such that

$$gmi = mgi \tag{3.8}$$

for all $i \in S$ and all possible 2-D rigid motion m.

The PRMI operators are those operators with which the mapping from a rigidly moved image is the same as that resulting from moving the result of the mapping from the original image. The attribute defined by a PRMI operator is called a PRMI attribute. The PRMI attribute at the moved point is equal to the attribute of the corresponding original point. In other words, under an image plane rigid motion, the matched points have the same value of the PRMI attribute.

Obviously, intensity is a PRMI attribute since g here is an identity mapping, and (3.8) holds trivially. The edgeness and cornerness defined in the following section are also PRMI attributes.

D. Edgeness and Cornerness

To get a continuous measure of edgeness, we define *edgeness* as the magnitude of the gradient of intensity. Namely, $e = gi = \|\nabla i\|$, where g is the operator that maps i to e:

$$e(\boldsymbol{u}) = \left\| \frac{\partial i(\boldsymbol{u})}{\partial \boldsymbol{u}} \right\|. \tag{3.9}$$

Property 1: Edgeness defined in (3.9) is a PRMI attribute.

Proof: Given any intensity image i and any image plane motion m, let $gmi = e'$. Then

$$\begin{aligned} e'(\boldsymbol{u}) &= \left\| \frac{\partial i(R_2\boldsymbol{u} + \boldsymbol{T}_2)}{\partial \boldsymbol{u}} \right\| = \left\| \frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}} R_2 \right\|_{\boldsymbol{v}=R_2\boldsymbol{u}+\boldsymbol{T}_2} \\ &= \left\| \frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}} \right\|_{\boldsymbol{v}=R_2\boldsymbol{u}+\boldsymbol{T}_2} = e(\boldsymbol{v}) \,|_{\boldsymbol{v}=R_2\boldsymbol{u}+\boldsymbol{T}_2}. \end{aligned} \tag{3.10}$$

Equation (3.10) holds because R_2 is a rotation matrix. Therefore, we get $e' = me$, which gives $gmi = e' = me = mgi$. □

Various methods for detecting corners can be found in the literature (e.g, [9], [19], [33]). Usually, a local polynomial fit to image intensity is performed, and then, the corners are detected based on the fitted polynomial. Since a polynomial fitting process is computationally expensive (e.g., a 7×7 neighborhood is needed) and the preprocessing discussed in Section III-H considerably removes image noise, the polynomial fitting is less desirable here. The cornerness at a point might be defined by instantaneous rate of change in the direction of gradient along an edge curve that passes through the point [19], [33]. However, it has been reported that instantaneous change performs more poorly than incremental change, which is measured between the direction of gradients at two points of intersection between a circle centered at the point and the iso-intensity contour passing through the point.

There are some problems with these corner detectors. First, the value of the directional change of gradient is thresholded to select corners. Then, a change of 180° is more likely to be selected than a change of 90°. However, the location of a right angle corner is more reliable than a corner with an acute angle or an obtuse angle. Second, cornerness measure should include a sign to distinguish a corner of a black rectangle on a white background from that of a white rectangle on a black background.

We define the *cornerness* in the following way without using computationally expensive polynomial fitting but achieving very good performance on real-world images. As we mentioned earlier, we define positive and negative cornerness separately. Roughly speaking, the edgeness at a point $\boldsymbol{u}$ measures the changes of the direction of gradient at two nearby points, weighted by the gradient at the point. These two points $\boldsymbol{u} + \boldsymbol{r}_a$ and $\boldsymbol{u} + \boldsymbol{r}_b$ (see Fig. 8) are located on a circle centered at $\boldsymbol{u}$. The radius of the circle is determined by the level of resolution. We choose $\boldsymbol{r}_a$ and $\boldsymbol{r}_b$ such that the directional derivative along the circle reaches the minimum and the maximum values, respectively (see Fig. 8). Let $\boldsymbol{a} = \nabla i(\boldsymbol{u} + \boldsymbol{r}_a)$, $\boldsymbol{b} = \nabla i(\boldsymbol{u} + \boldsymbol{r}_b)$, and angle $(\boldsymbol{a}, \boldsymbol{b})$ be the angle from $\boldsymbol{a}$ to $\boldsymbol{b}$ measured in radians counterclockwise, ranging from $-\pi$ to π. The closer the angle is to $\pi/2$, the higher the positive cornerness measure should be. In addition, the measure should be weighted by the magnitude of gradient at the point $\boldsymbol{u}$. Mathematically, the *positive cornerness* and *negative cornerness* are defined, respectively, by

$$p(\boldsymbol{u}) = \begin{cases} e(\boldsymbol{u})\{1 - |1 - \text{angle }(\boldsymbol{a}, \boldsymbol{b}) \cdot \{2/\pi\}|\} & 0 \leq \text{angle }(\boldsymbol{a}, \boldsymbol{b}) \leq \pi \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

and

$$n(\boldsymbol{u}) = \begin{cases} e(\boldsymbol{u})\{1 - |1 + \text{angle }(\boldsymbol{a}, \boldsymbol{b}) \cdot \{2/\pi\}|\} & -\pi \leq \text{angle}(\boldsymbol{a}, \boldsymbol{b}) \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

where column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are intensity gradients at $\boldsymbol{u}+\boldsymbol{r}_a$, and $\boldsymbol{u} + \boldsymbol{r}_b$, respectively:

$$\boldsymbol{a}^{\mathrm{T}} = \left. \frac{\partial i(\boldsymbol{s})}{\partial \boldsymbol{s}} \right|_{\boldsymbol{s}=\boldsymbol{u}+\boldsymbol{r}_a}$$

$$\boldsymbol{b}^{\mathrm{T}} = \left. \frac{\partial i(\boldsymbol{s})}{\partial \boldsymbol{s}} \right|_{\boldsymbol{s}=\boldsymbol{u}+\boldsymbol{r}_b}$$

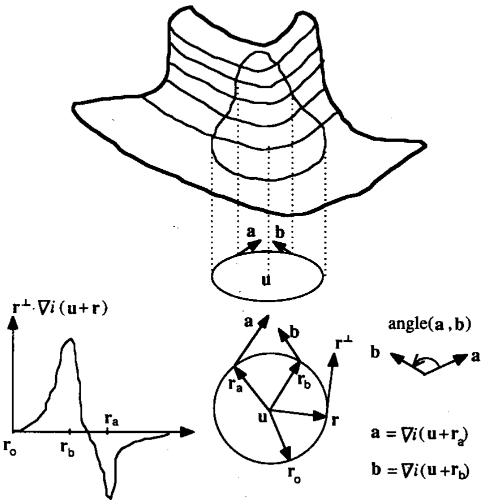

a
b
u
$\mathbf{r}^{\perp}\cdot\nabla i(\mathbf{u}+\mathbf{r})$
$\mathbf{r}_a$
$\mathbf{r}_o$
$\mathbf{r}_b$
a
b
$\mathbf{r}^{\perp}$
$\mathbf{r}_a$
$\mathbf{r}_b$
u
r
$\mathbf{r}_o$
angle(a, b)
b
a
$\mathbf{a} = \nabla i(\mathbf{u}+\mathbf{r}_a)$
$\mathbf{b} = \nabla i(\mathbf{u}+\mathbf{r}_b)$

Fig. 8. Definition of cornerness (see text).

where $\|\boldsymbol{r}_a\| = \|\boldsymbol{r}_b\| = r$, $\boldsymbol{r}_a$ and $\boldsymbol{r}_b$ are such that

$$\left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}_a} \cdot \boldsymbol{r}_a^{\perp} = \min_{\|\boldsymbol{r}\|=r} \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}} \cdot \boldsymbol{r}^{\perp} \quad (3.13)$$

and

$$\left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}_b} \cdot \boldsymbol{r}_b^{\perp} = \max_{\|\boldsymbol{r}\|=r} \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}} \cdot \boldsymbol{r}^{\perp}. \quad (3.14)$$

The superscript $\perp$ denotes the corresponding perpendicular vector; if $\boldsymbol{r} = (r_u, r_v)^{\mathrm{T}}$, then $\boldsymbol{r}^{\perp} = (-r_v, r_u)^{\mathrm{T}}$. If $\boldsymbol{r}_a$ and $\boldsymbol{r}_b$ that reach the minimum in (3.13) and the maximum in (3.14), respectively, are not unique, choose those that minimize $p(\boldsymbol{u})$ in (3.11) and (3.12) in addition to satisfying (3.13) and (3.14). The value r is a parameter of cornerness and is directly related to image resolution. In the discrete version, r is equal to the pixel size.

Property 2: The positive cornerness and negative cornerness defined above are PRMI attributes.

Proof: See Appendix.

From the definition of edgeness and cornerness, it can be seen that the edgeness and cornerness of a point are defined in a small neighborhood around the point. For a digital image, the neighborhood involves just nearby 3 × 3 pixels in our implementation. Therefore, as long as the motion does not exhibit considerable nonrigidity in the small window centered at a point, the attributes are approximately motion invariant. The relationships between the 3-D motion and image plane motion discussed in Section III-B indicate that a 3-D locally rigid motion results in a locally rigid 2-D image plane motion under some regularity conditions. Although the locally rigid motion is not a necessary condition for the approach to image matching discussed here (we only need the validity of the similarity of attributes), the study of locally rigid motion, its relationships to image plane motion, and the plane motion invariance of the attributes leads to a class of motion for which similarity of attributes is approximately true.

E. 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 toward the wrong direction if the orientational smoothness is not emphasized.

We represent the displacement vector field in the vicinity of a point $\boldsymbol{u}_0$ by a vector $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$. It is intended to approximate the displacement field within the region to which $\boldsymbol{u}_0$ belongs. In the implementation, $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$ is computed as

$$\bar{\boldsymbol{d}}(\boldsymbol{u}_0) = \int\!\!\int_{0<\|\boldsymbol{u}-\boldsymbol{u}_0\|<r} w(i(\boldsymbol{u})-i(\boldsymbol{u}_0), \boldsymbol{d}(\boldsymbol{u})-\boldsymbol{d}(\boldsymbol{u}_0))\boldsymbol{d}(\boldsymbol{u})d\boldsymbol{u} \quad (3.15)$$

where $0 < \|\boldsymbol{u} - \boldsymbol{u}_0\| < r$ denotes a region around $\boldsymbol{u}_0$, and $w(\cdot,\cdot)$ is a weight. In digital implementation, the integration is replaced by a summation over $\boldsymbol{u}_0$'s eight neighboring pixels. The weight is a function of intensity difference $i(\boldsymbol{u}) - i(\boldsymbol{u}_0)$ and displacement vector difference $\boldsymbol{d}(\boldsymbol{u})-\boldsymbol{d}(\boldsymbol{u}_0)$. The objective that $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$ represents the neighboring displacement vectors of the region of $\boldsymbol{u}_0$ suggests the following requirements on the weight.

1. The weight is large if the intensity difference is small. We assume that the small intensity difference is observed when two neighboring points $\boldsymbol{u}$ and $\boldsymbol{u}_0$ belong to the same region, and therefore, their displacement vectors should be similar.
2. If $\boldsymbol{u}$ and $\boldsymbol{u}_0$ have similar intensity but the corresponding displacement vectors are different, the weight should still be large. 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 $\boldsymbol{u}$ and $\boldsymbol{u}_0$ have different intensities and their displacement vectors are very different, the weight should be extremely small to suppress the influence of $\boldsymbol{u}$ on $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$.

Let $\eta_i = |i(\boldsymbol{u}) - i(\boldsymbol{u}_0)|$ and $\boldsymbol{\eta}_{\boldsymbol{d}} = \boldsymbol{d}(\boldsymbol{u}) - \boldsymbol{d}(\boldsymbol{u}_0)$. A definition of weight that satisfies the above criteria is as follows:

$$w(\eta_i, \boldsymbol{\eta}_{\boldsymbol{d}}) = \frac{c}{\epsilon + |\eta_i|(1 + \|\boldsymbol{\eta}_{\boldsymbol{d}}\|^2)} \quad (3.16)$$

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

$$\int\!\!\int_{0<\|\boldsymbol{u}-\boldsymbol{u}_0\|<r} w(i(\boldsymbol{u})-i(\boldsymbol{u}_0), \boldsymbol{d}(\boldsymbol{u})-\boldsymbol{d}(\boldsymbol{u}_0))d\boldsymbol{u} = 1. \quad (3.17)$$

To ensure that requirement 2) is met, a small scale factor can be applied to the term $\|\boldsymbol{\eta}_{\boldsymbol{d}}\|^2$, or alternatively, it can be set to zero, which gives a simpler form:

$$w(\eta_i, \boldsymbol{\eta}_{\boldsymbol{d}}) = \frac{c}{\epsilon + \mid \eta_i \mid}. \tag{3.18}$$

When the displacement field is computed with a computed occlusion map, the weights in (3.16) and (3.17) should be modified if, in (3.15), $\boldsymbol{u}_0$ is not an occluded point but $\boldsymbol{u}$ is. In this case, the weight corresponding to the occluded point $\boldsymbol{u}$ should be zero $w = 0$ since $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$ should not take the meaningless $\boldsymbol{d}(\boldsymbol{u})$ into account. If $\boldsymbol{u}_0$ is an occluded point, the weight need not be set to zero no matter whether $\boldsymbol{u}$ is an occluded point or not since the displacement vector at an occluded point is arbitrary and thus may conveniently take the value of such a $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$.

Thus, the weight is automatically determined based on the intensity difference and the displacement difference. The smoothness constraint imposes similarity of $\boldsymbol{d}(\boldsymbol{u}_0)$ and $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$. 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 $\|\boldsymbol{\eta}_{\boldsymbol{d}}\|^2$ results in a 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 (3.16) (unlike $\boldsymbol{\eta}_{\boldsymbol{d}}$); otherwise, the weight will be too sensitive to small changes in intensity. The weights thus implicitly take into account discontinuities and occlusions. The registered value $\bar{\boldsymbol{d}}(\boldsymbol{u}_0)$ allows us to perform matching using uniform numerical optimization despite the presence of discontinuities and occlusions. This is discused below.

F. Minimizing Residuals

Any given displacement vector field leads to measures of similarity, or residual errors, between the attributes of estimated corresponding points. The residual of intensity is defined by

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

Similarly, we define the residual of edgeness $r_e(\boldsymbol{u}, \boldsymbol{d})$, that of positive cornerness $r_p(\boldsymbol{u}, \boldsymbol{d})$, and that of negative cornerness $r_n(\boldsymbol{u}, \boldsymbol{d})$. The residual of orientation smoothness is defined by

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

where $(a, b)x(c, d) = ac - bd$ and the residual of displacement smoothness by

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

Under the conditions we discussed above, the similarity of attributes approximately holds. Therefore, we determine the displacement vector $\boldsymbol{d}$ such that the weighted sum of squares of residuals is minimized:

$$\min_{\boldsymbol{d}} \sum_{\boldsymbol{u}} \{r_i^2(\boldsymbol{u}, \boldsymbol{d}) + \lambda_e r_e^2(\boldsymbol{u}, \boldsymbol{d}) + \lambda_p r_p^2(\boldsymbol{u}, \boldsymbol{d}) + \lambda_n r_n^2(\boldsymbol{u}, \boldsymbol{d}) + \lambda_o r_o^2(\boldsymbol{u}, \boldsymbol{d}) + \lambda_d r_d^2(\boldsymbol{u}, \boldsymbol{d})\}$$

where r_e, r_p, r_n, r_o, and r_d are weighting parameters that are dynamically adjusted at different resolutions. Let

$$\boldsymbol{r} \stackrel{\Delta}{=} (r_i, r_e, r_p, r_n, r_o, r_d)^{\mathrm{T}}. \tag{3.19}$$

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

$$\boldsymbol{r}(\boldsymbol{d} + \delta_{\boldsymbol{d}}) = \boldsymbol{r}(\boldsymbol{d}) + \frac{\partial \boldsymbol{r}(\boldsymbol{d})}{\partial \boldsymbol{d}} \delta_{\boldsymbol{d}} + o(\|\delta_{\boldsymbol{d}}\|) \stackrel{\Delta}{=} \boldsymbol{r} + J\delta_{\boldsymbol{d}} + o(\|\delta_{\boldsymbol{d}}\|) \tag{3.20}$$

where

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

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

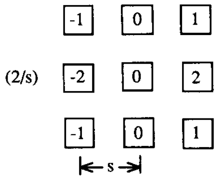
$$\Lambda = diag(1, \lambda_e, \lambda_p, \lambda_n, \lambda_o, \lambda_d).$$

We need to solve for $\delta_{\boldsymbol{d}}$ such that the sum of squared residuals is minimized. Neglecting high-order terms and minimizing $\|\Lambda(\boldsymbol{r} + J\delta_{\boldsymbol{d}})\|^2$, from (3.19), we get the formula for updating $\boldsymbol{d}$:

$$\boldsymbol{\delta}_{\boldsymbol{d}} = -(J^{\mathrm{T}}\Lambda^2 J)^{-1} J^{\mathrm{T}}\Lambda^2 \boldsymbol{r}(\boldsymbol{u}). \tag{3.21}$$

The partial derivatives in the entries of J are computed by a finite difference method in our implementation. Let s denote the distance between two adjacent points on a grid, along which the finite deference of the attributes is to be computed, assuming a unit spacing between adjacent pixels. Then, s should vary with the resolution. In addition, s should also vary with successive iterations within a resolution level. A large spacing is necessary for a rough displacement estimate when iterations start at each 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 Fig. 9, 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 the pixel grid.

For each grid point, the displacement vector $\boldsymbol{d}$ is replaced by $\boldsymbol{d} + \delta_{\boldsymbol{d}}$ according to (3.21). An iteration consists of such an updating for every grid point. At each resolution level, a fixed number of iterations (e.g., 20) are performed before the displacement field along the grid is projected to the next finer level. The final displacement field is obtained at the original image resolution.



$$(2/s)\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$$

Fig. 9. Mask for computing derivatives.

G. Recursive Blurring

As shown in Fig. 7, the images need to be blurred to higher levels. The original images are first preprocessed by the methods to be discussed in Section III-H. Then, four attribute image 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 k ($k = 4$ for intensity, $k = 3$ for edgeness, and $k = 2$ for cornerness). The locations of these four pixels are such that each is centered at a quadrant of a square of $a \times a$ (see Fig. 10). a 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 $a \times a$. To enhance sparse edges and corners, k is smaller than 4 for edgeness and cornerness. Therefore, 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.

H. Preprocessing, Normalization, and Parameter Selection

The matching algorithm has to cope with images of a wide variety of scenes. The purposes of preprocessing are 1) to normalize the images so that the algorithm can use a set of standard parameters for different scenes and 2) to filter out noise in the images.

The pair of intensity images to be matched is first normalized by a linear function so that the maximum and minimum intensities are equal to 255 and 0, respectively. (This range from 0 to 255 is to adapt to the hardware representation we used.) Then, it is filtered with a small (3 by 3) low-pass filter to suppress gray-level noise.

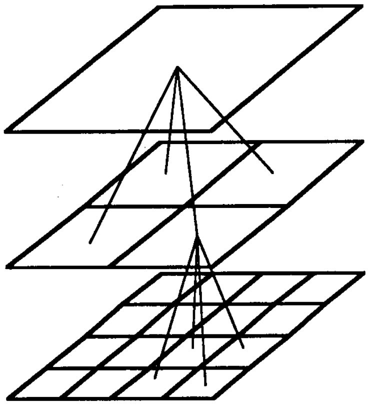


Fig. 10. Recursive blurring and limiting (see text).

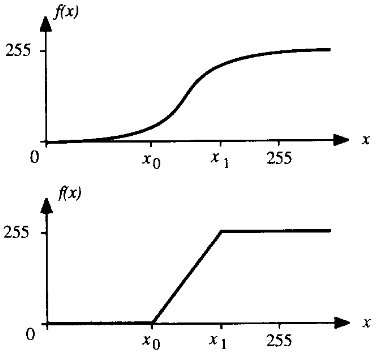


Fig. 11. Two normalization functions.

Similarly, the edgeness and the cornerness also need to be normalized. The following considerations motivate the normalization. First, small gradients are more susceptible to intensity noise and are not reliable. Second, strong gradients may excessively override other moderate gradients in edgeness measurement. Third, different scenes have different ranges of gradient magnitude, and the algorithm should treat them in a systematic way. Therefore, we slightly modify the definition of edgeness presented in (3.9). Edgeness is the magnitude of the gradient, normalized, and transformed by a function f shown in Fig. 11:

$$e(\boldsymbol{u}) = f(\|\nabla i(\boldsymbol{u})\|). \tag{3.22}$$

The function f maps the magnitude of gradient onto the whole range [0, 255]. It has two transition points x_0 and x_1. From $x = 0$ to $x = x_0$, $f(x) \approx 0$ 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 the gradient with a moderate magnitude. For $x > x_1$, $f(x) \approx 255$ to limit strong edges and relatively enhance moderate edges. The values of two transition points 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.

The edgeness $e(\boldsymbol{u})$ used in the definition of cornerness (3.11) and (3.12) should also use the modified definition (3.22) as well. Note that such modified edgeness and cornerness are still PRMI attributes, and all the related proofs still hold if g is replaced by fg.

The preprocessing and normalization steps make the algorithm perform consistently for a wide variety of images using a set of standard parameters that are selected based on a moderate number of image examples. At the present implementation, the parameters are determined through trials. A set of parameters (e.g, those in (3.16) or (3.17) and (3.18)) are determined for each level of resolution. At coarse levels, the edgeness, cornerness, and smoothness have relatively large weights. Their weights are reduced gradually down to finer levels since the smoothness constraint should be reduced at finer levels, where details of the displacement are obtained, and the cornerness and edgeness measurements at finer levels are more susceptible to noise than the significantly blurred measurements.

I. Outline of the Matching Algorithm

The following summarizes the steps of the procedure that computes the displacement field from one image to the other (see Fig. 7):

1. Filter two images using a 3×3 low-pass filter to remove noise (Section III-H).
2. Normalize the pair of images (Section III-H).
3. Generate attribute images: intensity, edgeness, positive cornerness, and negative cornerness (Section III-D).
4. Set the level to the highest, e.g., $l = 6$, and set the displacement field on the grid (level 6) to zero.
5. Blur attribute images to level l (Section III-G).
6. Compute the displacement field along the grid (Section III-F). Perform a number of iterations (e.g., 20).
7. If $l = 0$, the procedure returns with the resulting displacement field; otherwise, go to 8.
8. Project the displacement field on the grid of level l to the grid of level $l - 1$ (copying the vector at each grid point to the four corresponding grid points of level $l-1$); decrement l by 1 and go to 5.

Suppose we need to determine the displacement field from image 1 to image 2. In order to first obtain occlusion map 1, first compute the displacement field from image 2 to image 1 using the above procedure without occlusion information (assuming image 2 has no occluded region). The displacement field computed is used to determine the occlusion map 1 for image 1 as discussed in Section II-C (see Fig. 6). In the implementation, the occlusion maps are filtered by 3×3 median filters to remove single pixel wide occlusion and noise. Then, the displacement field from image 1 to image 2 is computed using the occlusion map 1 by calling the above procedure starting from step 4. In step 6, if a point in image 1 is marked in the occlusion map 1, it is not visible in image 2, and therefore, the displacement vector from this point cannot be determined. We simply copy the vector $\bar{\boldsymbol{d}}$ to this occluded point. The final computed displacement field assigns a displacement vector to every pixel in image 1.

IV. Refinements

The computational structure illustrated in Fig. 7 can be extended further in order to improve the computed displacement field. This section discusses two techniques that we have implemented: 1) refinement through a bottom-up and top-down computational scheme and 2) refinement using the rigidity constraint.

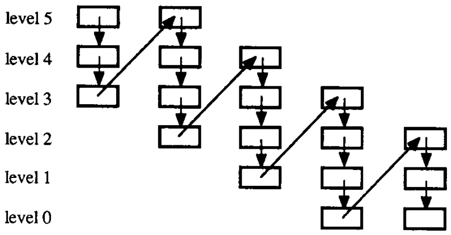


Fig. 12. Bottom-up and top-down refinement.

A. Bottom-Up and Top-Down

The data flow characterized by the projections shown in Fig. 7 is of a top-down fashion in the sense that the displacement fields are computed from high levels (coarse resolution) down to low levels. At high levels, coarse estimates of the field are computed to cope with large disparities. 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 this issue in a slightly different way: The result of a coarse level needs to be verified and refined at low level, where a more detailed image is available. Such a 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 back 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 top-down, bottom-up structures can be embedded in the entire algorithm. Fig. 12 shows an example of the computational structure that we have implemented, which leads to a tangible improvement for some images over a straight top-down scheme.

B. Refinement Based on Rigidity

Mathematically, the displacement field is equivalent to point correspondences between two images. If the scene is rigid, the algorithm that computes the motion parameters and the structure of the rigid scene can be used to determine the motion between two images and the surface of the scene (since we have a very dense displacement field).

All displacement vectors should satisfy the rigidity condition, i.e., they describe a rigid motion. Due to noise, this is generally not true. Since the motion parameters are estimated from a large number of displacement vectors based on some optimality criteria, they are generally more reliable than an individual displacement vector. Each displacement vector can be modified based on the estimated motion parameters. Since we determine displacement on a fixed grid, the starting image

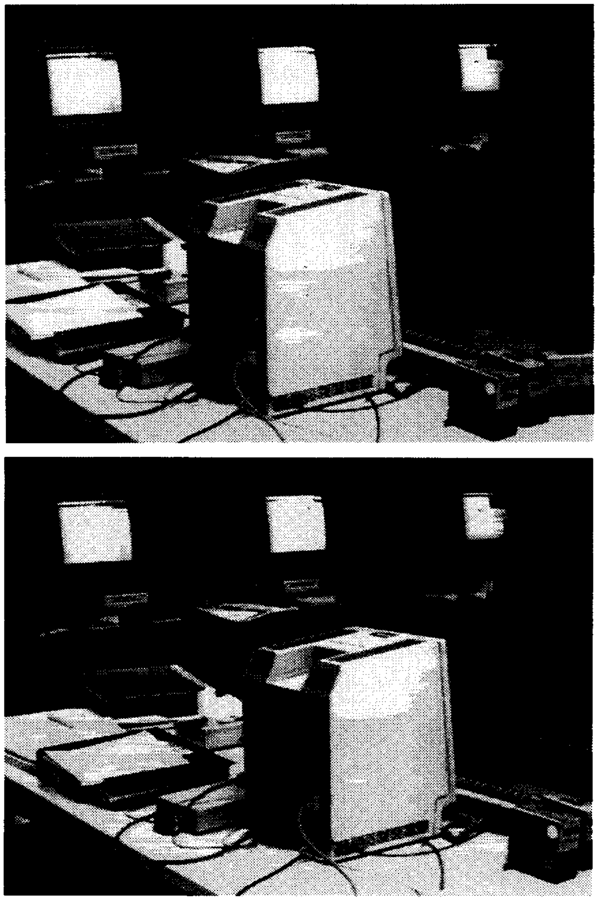


Fig. 13. Two views of a laboratory scene (called the Mac scene).

coordinates of each displacement vector are fixed. To satisfy the rigidity condition, we move the estimated structure at the viewpoint at the first image (sampled on the pixel grid) using the estimated motion parameters to the new position. Then, the projection of the structure at the new position determines the new displacement field. Then, the resulting displacement field exactly satisfies the rigidity constraint.

V. Experimental Results

Experiments have been conducted on a variety of real-world scenes. A CCD monochrome video camera with roughly 512 × 512 pixels was used as an image sensor. The focal length of the camera was calibrated, but no correction has been made for the camera nonlinearity. The camera took two images at different positions for each scene. The number of resolution levels used for experiments is equal to 7. Twenty iterations are performed at each level. The matching results shown here have not been refined by the method discussed in Section IV.

First, we present the results for the pair of images shown in Fig. 13, which is called the 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. With this pair of 512 × 512 images, the largest disparity is about 80 pixels. The edgeness image is shown in Fig. 14, and the positive and negative cornerness images are shown in Figs. 15 and 16, respectively. The blurred attribute images at the highest level ($l = 6$) and the next finer level ($l = 5$) are shown in Fig. 17. It can be seen that the corresponding blurred attribute images are quite different. Although all the attribute images are derived from the original intensity images, different attribute images characterize different properties of the intensity images, and they show drastically different attribute images. Such a difference among different attribute images is what we need to provide an overdetermined system of matching criteria. At the coarsest level, the grid on which displacement is to be computed is very sparse (8×8 at $l = 6$), and it is also the grid on which the finite differences are computed in the first 10 iterations. The samples of the computed displacement field along the grid are presented at different levels and superimposed on the corresponding blurred intensity images of image 1, which have been extended to provide context for

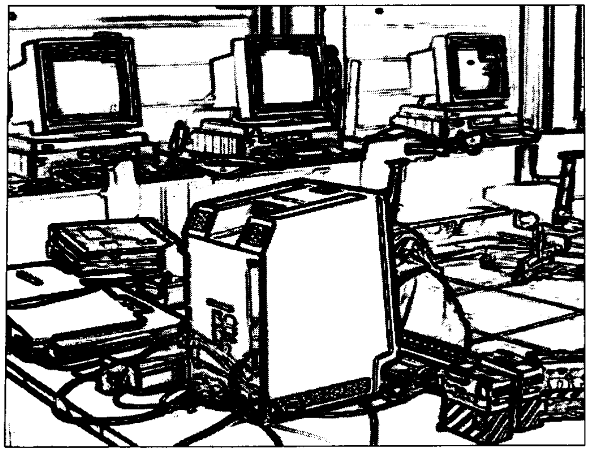


Fig. 14. Edgeness of the first image of the Mac scene.

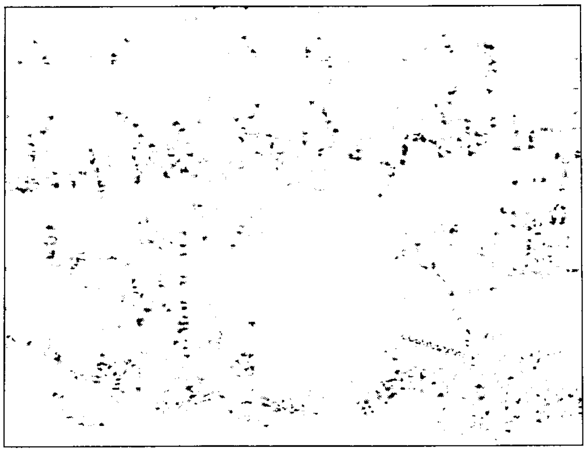


Fig. 15. Positive cornerness of the first image of the Mac scene.

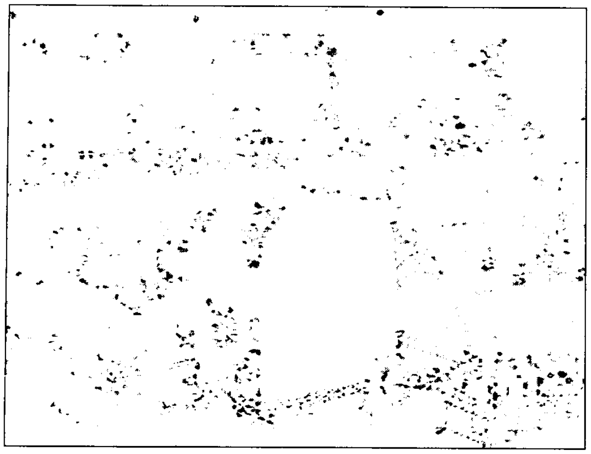


Fig. 16. Negative cornerness of the first image of the Mac scene.

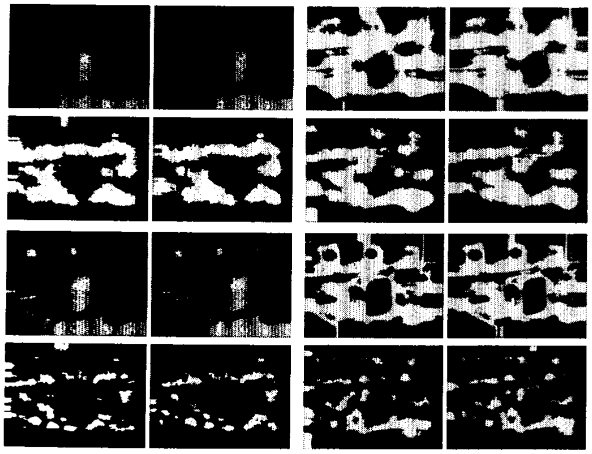


Fig. 17. Blurred attribute images. Upper two rows: level 6; left pair in the first row: blurred intensity image pair; right pair in the first row: blurred edgeness image pair; left pair in the second row: blurred positive cornerness image pair; right pair in the second row: blurred negative cornerness image pair; lower two rows: level 5 in the same arrangement as the upper two rows.

the borders. The computed displacement fields are shown in Fig. 18 at level 6 and in Fig. 19 at level 5. At levels lower than 5, the grid for the displacement field is too dense to display the field completely. Instead, only samples along a sparse 16 × 16 grid are shown for lower levels. Fig. 20 shows the sample of the displacement field at level 4. The samples of the displacement field at level 1 is shown in Fig. 21. Examing by flickering between two images on a Sun workstation, 95% of the vectors shown in Fig. 21 appear to have no visible errors. Between the top of the Macintosh computer in the foreground and the central workstation is a dark region corresponding to the wall. Due to drastic depth changes across this region and absence of texture inside the region, this region appears to undergo a deformation. The resulting displacement vectors in this region are consistent with the deforming interpretation, which is not physically correct. A correct solution to the displacement field in this region may require greater resolution

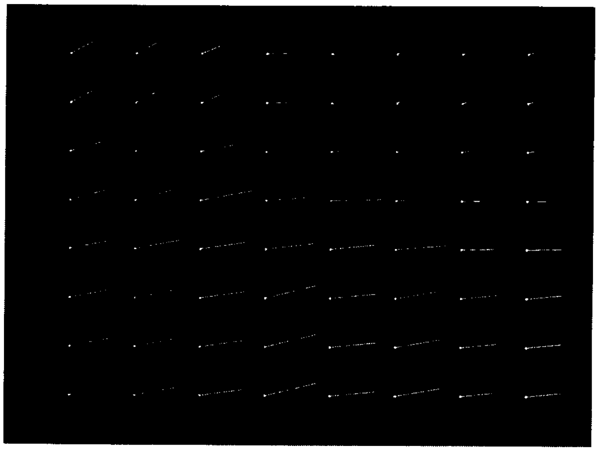


Fig. 18. Computed displacement field at level 6 for the Mac scene, superimposed on the blurred extended intensity image. (The intensity image is extended in four directions to provide context for the border areas.)

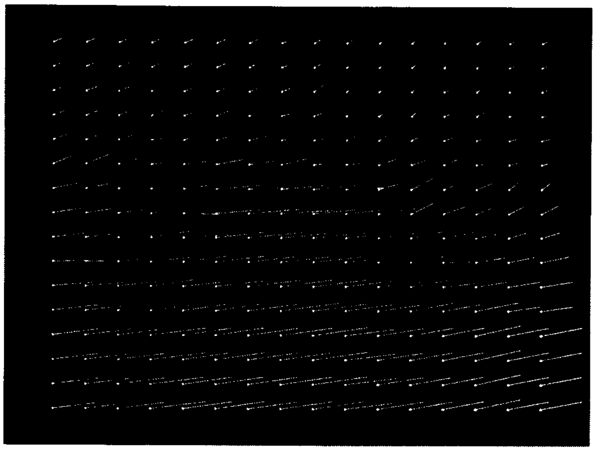


Fig. 19. Computed displacement field at level 5 for the Mac scene, superimposed on the blurred extended intensity image.

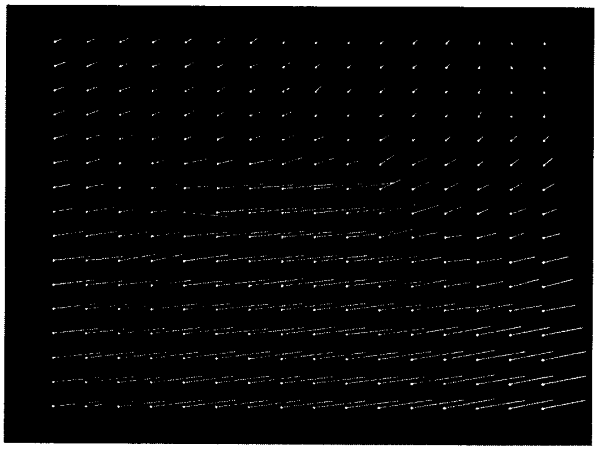


Fig. 20. Samples of the computed displacement field at level 4 for the Mac scene, superimposed on the blurred extended intensity image.

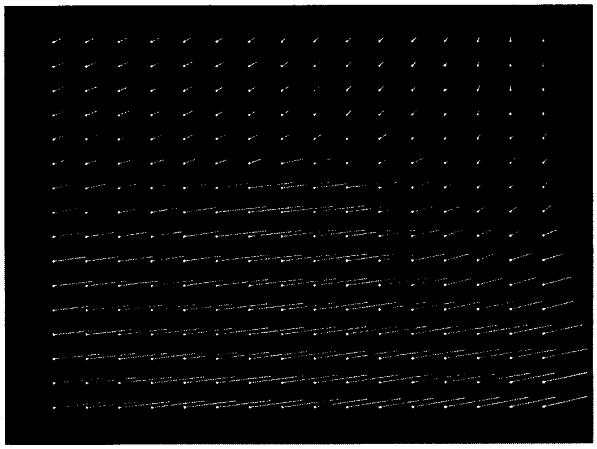


Fig. 21. Samples of the computed displacement field at level 1 for the Mac scene, superimposed on the blurred extended intensity image.

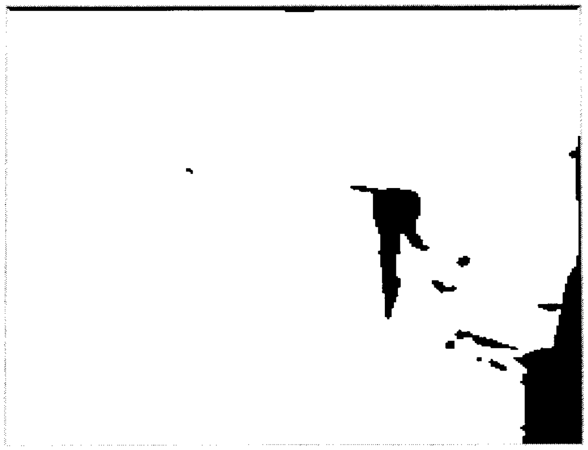


Fig. 22. Computed occlusion map 1 for the Mac scene. Black areas in occlusion map 1 indicate that the corresponding areas in image 1 (the first image in Fig. 13) are not visible in image 2 (the second image in Fig. 13).

and brighter lighting in order to pick up the fine texture on the wall. The occlusion map 1 is shown in Fig. 22, where black areas indicate that the corresponding areas in image 1 (the first image in Fig. 13) are not visible in image 2 (the second image in Fig. 13). The occlusion map is to show relatively large occluded regions (more than one pixel wide) instead of occlusion boundaries that can be easily detected by analyzing discontinuities in the constructed depth map.

Since the scene is rigid in this case, the algorithms presented in [30] and [32] were employed to compute the motion parameters and the 3-D structure of the scene from the computed displacement field. The reconstructed 3-D surface is shown with the value of $1/(z)$, where z is the depth as an intensity image in Fig. 23, and is plotted in Fig. 24. Those surfaces agree fairly well with those observed in the real scene. It is worth mentioning that the complicated surfaces of the manuals and books on the front table have been recovered. The parameters of the motion of the scene

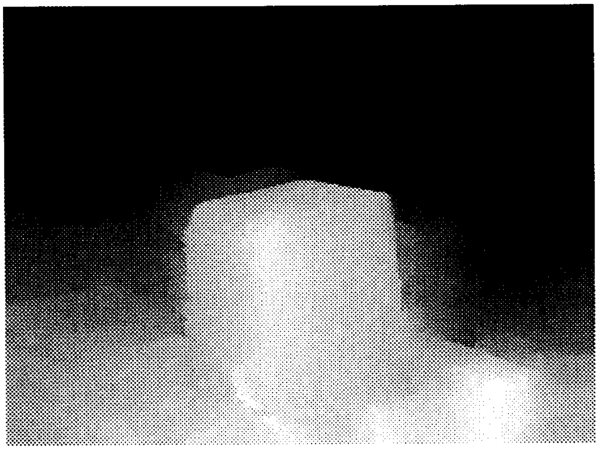


Fig. 23. Computed 3-D surface $(1/z)$ shown as intensity image for the Mac scene (from the viewpoint used for image 1).

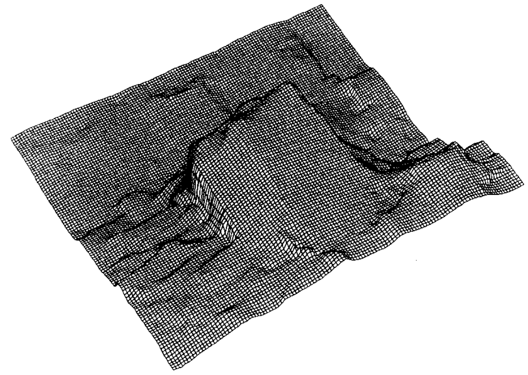


Fig. 24. Perspective plot of computed 3-D surface $(1/z)$ for the Mac scene (from the viewpoint used for image 1).

relative to the camera are shown in Table I. The translation direction and rotation axis are represented by three components (up, right, forward). Because the ground truth of the camera motion was not available (obtaining ground truth requires extensive calibrations), we were not able to determine the actual accuracy of those motion parameters. However, we can measure the discrepancies between the projection of the recovered 3-D position of the scene points and the actual observed projection. Let us define the *image error* as

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

where n is the number of points in each image (number of displacement vectors), 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 the analogous distance for image 2. Remember that the computed 3-D positions of the points at two time instances exactly satisfy the rigidity constraint. If the displacement field is not correct and does not correspond to the motion of a rigid scene, the image error will be large, no matter how good the performance

TABLE I
DATA AND RESULTS FOR THE MAC SCENE

Parameters	x (upward)	y (rightward)	z (forward)
Translation	0.016	0.991	0.133
Rotation axis	0.966	0.176	−0.188
Rotation angle	1.61°		
Image error	0.00033		
Pixel width	0.00094		

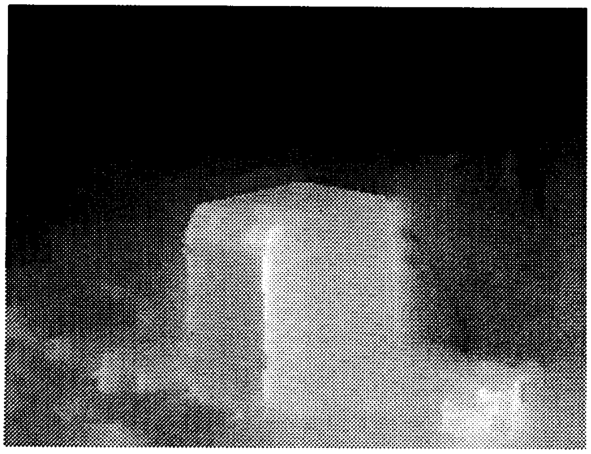


Fig. 25. Performance using only intensity: The computed 3-D surface $(1/z)$ shown as intensity image for the Mac scene (from the viewpoint used for image 1).

of the motion and structure estimation algorithm is. On the other hand, if the errors in the displacement field do not violate the rigidity constraint, the image error can still be small provided the performance of the motion and structure estimation algorithm is good. As shown in Table I, 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 the rigidity constraint. The displacement field could still conceivably make systematic errors to depict a rigid scene different from the real one, but such systematic errors are unlikely except those caused by the existence of multiple interpretations. A quantitative estimation for the accuracy of the motion parameters based on the image errors is presented in [32].

In order 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 zeroes. The depth from the resulting matching is shown in Fig. 25, which can be seen to contain many errors. The result without using the occlusion map is shown in Fig. 26, from which we can see that severe errors occur around the occluded regions.

Two images of another scene (called the Desk scene) are shown in Fig. 27. The samples of the computed displacement field are presented in Fig. 28, and the estimated motion parameters are given in Table 2. The resulting 3-D surface is shown with value $1/z$ as intensity image in Fig. 29.

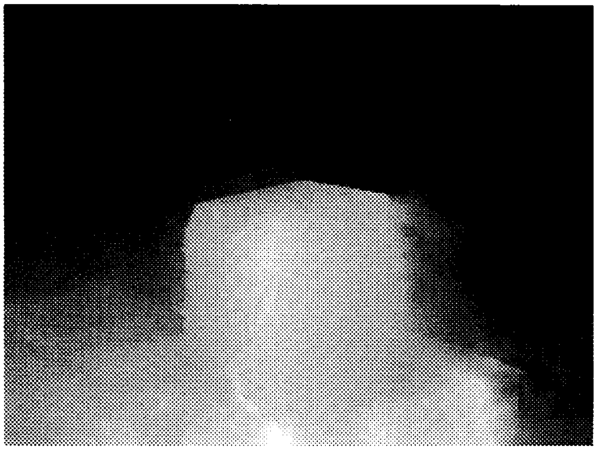


Fig. 26. Performance without identifying occluded regions: The computed 3-D surface $(1/z)$ shown as intensity image for the Mac scene (from the viewpoint used for image 1).

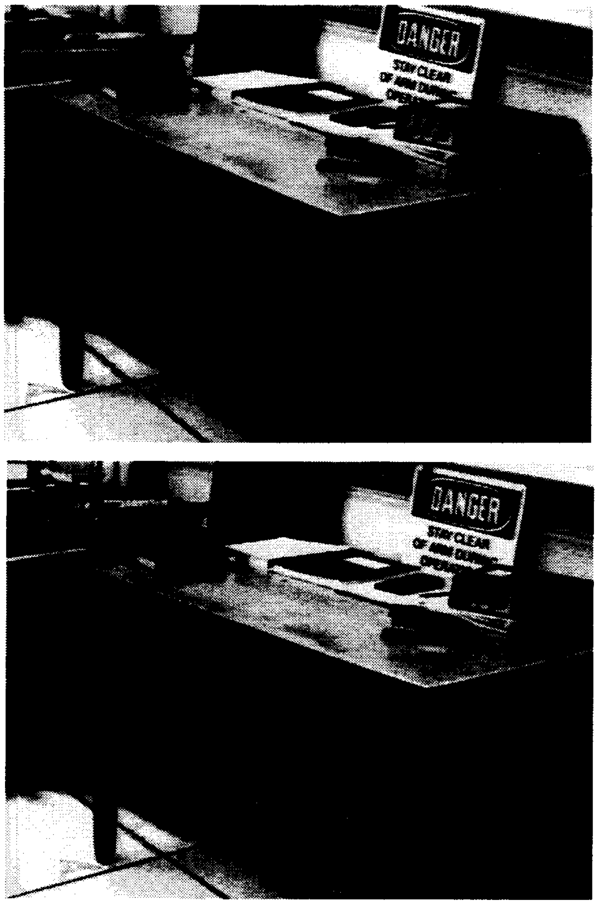


Fig. 27. Two images of the Desk scene.

Fig. 30 gives two images of one more scene (called the Path scene), and the samples of the computed displacement field are presented in Fig. 31. The results of motion estimation are shown in Table III.

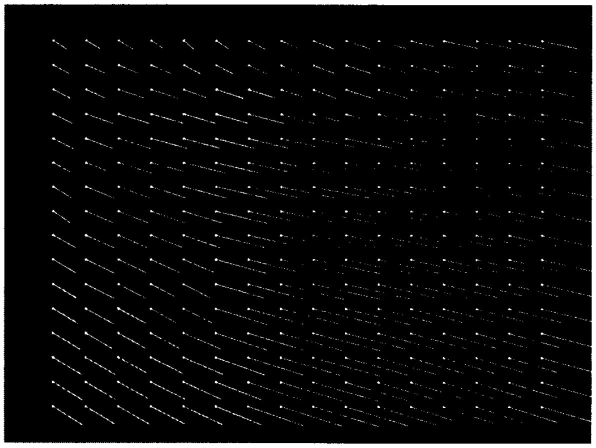


Fig. 28. Samples of the computed displacement field at level 1 for the Desk scene, superimposed on the blurred extended intensity image.

TABLE II
Data and Results for the Desk Scene

Parameters	x (upward)	y (rightward)	z (forward)
Translation	−0.045	0.943	−0.329
Rotation axis	−0.872	0.385	0.303
Rotation angle		−1.35°	
Image error		0.00051	
Pixel width		0.00094	

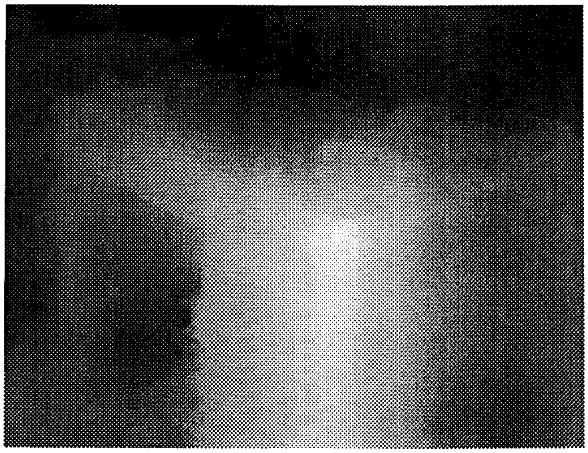


Fig. 29. Computed 3-D surface ($1/z$) shown as intensity image for the Desk scene (from the viewpoint used for image 1).

VI. Summary and Discussion

We have presented in this paper an approach to computing the 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 the displacement field and occlusion are taken into account to analyze complicated real-world scenes. This approach is capable of dealing with large disparities.

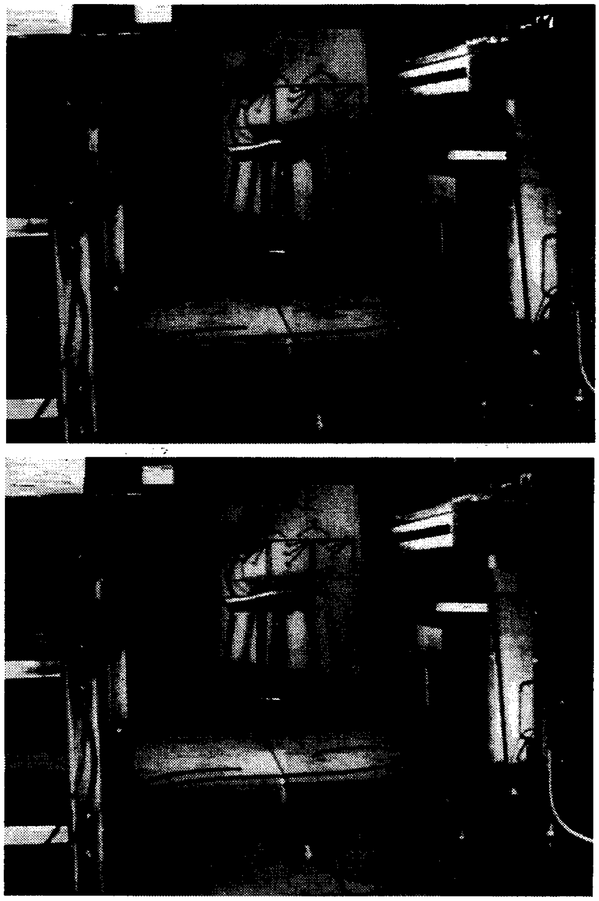


Fig. 30. Two images of the Path scene.

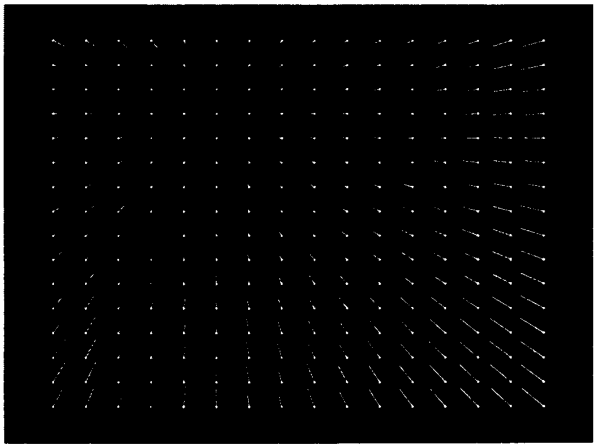


Fig. 31. Samples of the computed displacement field at level 1 for the Path scene, superimposed on the blurred extended intensity image.

In the current implementation of the algorithm, intensity, edgeness, and cornerness are used as matching attributes. Those attributes are invariant under image plane rigid motion.

TABLE III
DATA AND RESULTS FOR THE PATH SCENE

Parameters	x (upward)	y (rightward)	z (forward)
Translation	0.095	−0.057	0.994
Rotation axis	0.709	0.407	0.576
Rotation angle		0.13°	
Image error		0.00032	
Pixel width		0.00094	

Locally rigid motion is introduced, and its relationships with image plane motion motivate the use of image plane (rigid) motion-invariant attributes for matching. Since the edgeness and cornerness attributes are low-level attributes defined in a very small neighborhood around a point (specifically a 3 × 3-pixel neighborhood), the attributes are insensitive to those motions that do not exhibit significant deformation in the small neighborhood.

The matching algorithm does not require extensively textured images. From the matches obtained, dense 3-D surface and occlusion maps are computed for real-world scenes. The discrepancy between the projection of the computed 3-D points and the observed image points (image error) is about one half of the pixel width.

In order to relate the presented algorithm with others, let us first make some observations about the role of J given in (3.20). The top four rows of J determine the matching, and the bottom three rows account for the intraregional smoothness. At a point of image $i'(\boldsymbol{u})$, where there are strong transitions of intensity, edgeness, and cornerness, or a subset of them, the first four rows of J are relatively strong and determine the optimal $\delta_{\boldsymbol{d}}$ to update the displacement vector. The bottom three rows are relatively weak, and they 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 resulting effect is extrapolating across a uniform region. The first four linear equations of

$$\boldsymbol{r}(\boldsymbol{d}+\delta_{\boldsymbol{d}}) = \boldsymbol{r}(\boldsymbol{d}) + \frac{\partial \boldsymbol{r}(\boldsymbol{d})}{\partial \boldsymbol{d}}\delta_{\boldsymbol{d}} = 0 \tag{6.1}$$

yield four linear equations in terms of two components of $\delta_{\boldsymbol{d}}$, which determine four lines in the space of $\delta_{\boldsymbol{d}}$. Since the measurements are relatively noisy, those lines are not very reliable and generally do not intersect at a single point. A weighted least squares solution of (6.1) 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). Those methods resort to some smoothness constraints. 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 those (usually iterative) methods can be any one in this class. Therefore, those methods do not give the correct solution in general.

In our approach, the system is generally overdetermined, and smoothness is used mainly for filling in uniform regions. Although the available information for matching is just the original intensity images, the matching criteria here are based on not only individual intensity values but also on the relationships between those intensity values. Edgeness and cornerness characterize some meaningful local relationships at a point, and they are approximately invariant under locally rigid image plane displacement. These attributes provide additional information that is needed to guide the matching. At a coarse level, they provide texture content of the original images. More importantly, they lead to a generally overdetermined system based solely on attribute matching instead of regularization (smoothness). Such an overdetermination significantly improves the stability of the solution.

Computationally, since the 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 computer architectures. This is an advantage over symbolic discrete matching approaches that use high-level discrete primitives and provide only sparse matches.

Some questions could be raised about the difference between the multiattribute scheme discussed here and one using spatiotemporal Gabor filters [13]. In fact, the framework presented here differs from that of the Gabor energy-based method in several fundamental ways. First, the Gabor energy is not rotationally invariant, and therefore, it is not a PRMI attribute. Second, the spatiotemporal method is meant for small displacements and is not suited for the task of matching with large interframe disparities. Third, the computational scheme used here is based on exact matching, whereas that of Gabor filters is based on prediction in the sense that the predicted energies and the velocities are exact *if* the pattern has a flat spectrum. (The normalization of each spatial orientation may alleviate, to some degree, the problem with local patterns that do not have a flat spectrum).

APPENDIX

Property 2: The positive cornerness and negative cornerness defined are PRMI attributes.

Proof: Let $p = gi$, where g is the operator that maps i to p, which is the positive cornerness image. For convenience, denote the moved image mi by i_m: $i_m = mi$, and the edgeness image of i_m by e_m. We need to prove $gmi = mgi$, or equivalently using the above notation, $gi_m = mp$. According to the definition of the positive cornerness, we have

$$g\ i_m(\boldsymbol{u}) = \begin{cases} e_m(\boldsymbol{u})\{1 - |1 - \text{angle}(\boldsymbol{a},\boldsymbol{b})\{2/\pi\}|\} & 0 \le \text{angle}(\boldsymbol{a},\boldsymbol{b}) \le \pi \\ 0 & \text{otherwise} \end{cases} \tag{A.1}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are intensity gradients at $\boldsymbol{u} + \boldsymbol{r}_a$ and $\boldsymbol{u} + \boldsymbol{r}_b$, respectively:

$$\boldsymbol{a}^{\mathrm{T}} = \left.\frac{\partial i_m(\boldsymbol{s})}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}=\boldsymbol{u}+\boldsymbol{r}_a}$$

$$\boldsymbol{b}^{\mathrm{T}} = \left.\frac{\partial i_m(\boldsymbol{s})}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}=\boldsymbol{u}+\boldsymbol{r}_b}$$

where $\|\boldsymbol{r}_a\| = \|\boldsymbol{r}_b\| = r$, and $\boldsymbol{r}_a$ and $\boldsymbol{r}_b$ are such that

$$\left.\frac{\partial i_m(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}_a} \cdot \boldsymbol{r}_a^{\perp} = \min_{\|\boldsymbol{r}\|=r} \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}} \cdot \boldsymbol{r}^{\perp} \qquad \text{(A.2)}$$

and

$$\left.\frac{\partial i_m(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}_b} \cdot \boldsymbol{r}_b^{\perp} = \max_{\|\boldsymbol{r}\|=r} \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}} \cdot \boldsymbol{r}^{\perp}. \qquad \text{(A.3)}$$

Since the edgeness is a PRMI attribute, we have

$$e_m(\boldsymbol{u}) = e(\boldsymbol{v})|_{\boldsymbol{v}=R_2\boldsymbol{u}+\boldsymbol{T}_2} \qquad \text{(A.4)}$$

where R_2 and $\boldsymbol{T}_2$ represent the image plane motion m. Let $\boldsymbol{r}'_a \triangleq R_2\boldsymbol{r}_a$, $\boldsymbol{r}'_b \triangleq R_2\boldsymbol{r}_b$, $\boldsymbol{b}' \triangleq R_2\boldsymbol{b}$, and $\boldsymbol{r}' \triangleq R_2\boldsymbol{r}$. According to the definition of the function angle, we have

$$\text{angle } (\boldsymbol{a},\boldsymbol{b}) = \text{ angle } (\boldsymbol{a}',\boldsymbol{b}'). \qquad \text{(A.5)}$$

From (A.4) and (A.5), we can rewrite (A.1) as the equation at the bottom of the page: Next, we need to derive the relationships among $\boldsymbol{a}'$, $\boldsymbol{b}'$, and the image i. Since $i_m(\boldsymbol{s}) = i(R_2\boldsymbol{s} + \boldsymbol{T}_2)$, we have

$$\left.\frac{\partial i_m(\boldsymbol{s})}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}=\boldsymbol{u}+\boldsymbol{r}_a} = \left.\frac{\partial i_m(\boldsymbol{v})}{\partial \boldsymbol{v}}R_2\right|_{\boldsymbol{v}=R_2(\boldsymbol{u}+r_a)+\boldsymbol{T}_2} = \left.\frac{\partial i_m(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=R_2u+\boldsymbol{T}_2+\boldsymbol{r}'_a} \qquad \text{(A.7)}$$

or

$$(\boldsymbol{a}')^T = \left.\frac{\partial i(\boldsymbol{v})}{\partial(\boldsymbol{v})}\right|_{\boldsymbol{v}=R_2u+\boldsymbol{T}_2+r'_a}. \qquad \text{(A.8)}$$

Similarly

$$(\boldsymbol{b}')^T = \left.\frac{\partial i(\boldsymbol{v})}{\partial(\boldsymbol{v})}\right|_{\boldsymbol{v}=R_2u+\boldsymbol{T}_2+r'_b}. \qquad \text{(A.9)}$$

From (A.7), it follows that

$$\left.\frac{\partial i_m(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{r}_a} \cdot \boldsymbol{r}_a^{\perp} = \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=R_2\boldsymbol{u}+\boldsymbol{T}_2+\boldsymbol{r}'_a} \cdot R_2\boldsymbol{r}_a^{\perp} = \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=R_2u+\boldsymbol{T}_2+\boldsymbol{r}'_a} \cdot (\boldsymbol{r}'_a)^{\perp}.$$

Therefore, (A.2) and (A.3) lead to

$$\left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{R}_2\boldsymbol{u}+\boldsymbol{T}_2+\boldsymbol{r}'_a} \cdot (\boldsymbol{r}_a{}')^{\perp} = \min_{\|\boldsymbol{r}'\|=r} \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=R_2u+\boldsymbol{T}_2=\boldsymbol{r}'} \cdot (\boldsymbol{r}')^{\perp} \qquad \text{(A.10)}$$

and

$$\left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\boldsymbol{R}_2\boldsymbol{u}+\boldsymbol{T}_2+\boldsymbol{r}'_b} \cdot (\boldsymbol{r}_b{}')^{\perp} = \max_{\|\boldsymbol{r}'\|=r} \left.\frac{\partial i(\boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=R_2u+\boldsymbol{T}_2+\boldsymbol{r}'} \cdot (\boldsymbol{r}')^{\perp} \qquad \text{(A.11)}$$

From (A.6) and (A.8)–(A.11) and the definition of the positive cornerness, it follows that

$$gi_m(\boldsymbol{u}) = p(\boldsymbol{v})|_{\boldsymbol{v}=R_2\boldsymbol{u}+\boldsymbol{T}_2}$$

or, in terms of operators, $gi_m = mp$. An analogous proof leads to the corresponding conclusion for the negative cornerness.□

References

[1] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer. A*, vol. 2, no. 2, pp. 284–299, 1985.

[2] P. Anandan and R. Weiss, "Introducing a smoothness constraint in a matching approach for the computation of optical flow fields," in *Proc. Workshop Comput. Vision: Representation Contr.* (Bellaire, MI), 1985, pp. 186–194.

[3] N. Ayache and O. Faugeras, "Building, registration, and fusing noisy visual maps," in *Proc. First Int. Conf. Comput. Vision* (London), 1987, pp. 73–82.

[4] S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, no. 4, pp. 333–340, 1980.

[5] S. T. Barnard, "A stochastic approach to stereo vision," in *Proc. Fifth Nat. Conf. Artificial Intell.* (Philadelphia, PA), 1986, pp. 676–680.

[6] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light.* Oxford: Pergamon, 1975.

[7] O. J. Braddick, "A short-range process in apparent motion," *Vision Res.*, vol. 14, pp. 519–527, 1974.

[8] ——, "Low-level and high-level processes in apparent motion," *Philo. Trans. Royal Soc. London B*, vol. 290, pp. 137–151, 1980.

[9] L. Dreschler and H. -H. Nagel, "Volumetric model and 3-D trajectory of a moving car derived from monocular TV frame sequences of a street scene," *Comput. Graphics Image Processing*, vol. 20, pp. 199–228, 1982.

[10] F. Glazer, G. Reynolds, and P. Anandan, "Scene matching by hierarchical correlation," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1983, pp. 432–441.

[11] W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, no. 1, pp. 17–34, 1985.

$$g\ i_m(\boldsymbol{u}) = \begin{cases} e(\boldsymbol{v})|_{\boldsymbol{v}=R_2u+\boldsymbol{T}_2}\{1 - |1 - \text{angle}(\boldsymbol{a}',\boldsymbol{b}')\{2/\pi\}|\} & 0 \le \text{angle}(\boldsymbol{a}',\boldsymbol{b}') \le \pi \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(A.6)}$$

[12] W. K. Gu, J. Y. Yang, and T. S. Huang, "Matching perspective views of a polyhedron using circuits," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 3, pp. 390–400, 1987.

[13] D. J. Heeger, "Optical flow from spatiotemporal filters," in *Proc. Int. Conf. Comput. Vision* (London), 1987, pp. 181-190.

[14] E. C. Hildreth, *The Measurement of Visual Motion.* Cambridge, MA: MIT Press, 1983.

[15] W. Hoff and N. Ahuja, "Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-11, no. 2, pp. 121–136, 1989.

[16] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185-203, 1981.

[17] J. K. Kearney, W. B. Thompson, and D. L. Boley, "Optical flow estimation: An error analysis of gradient-based methods with local optimization," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 2, pp. 229–244, 1987.

[18] R. Kingslake, *Lens Design Fundamentals.* New York: Academic, 1978.

[19] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Patt. Recogn. Lett.*, vol. 1, pp. 95–102, 1982.

[20] H. S. Lim and T. O. Binford, "Stereo correspondence: A hierarchical approach," in *Proc. Image Understanding Workshop*, 1987.

[21] D. Marr and T. Poggio, "A theory of human stereo vision," in *Proc. R. Soc. London*, 1979, pp. 301–328, vol. B 204.

[22] J. E. W. Mayhew and J. P. Frisby, "Psychophysical and computational studies toward a theory of human stereopsis," *Artificial Intell.*, vol. 17, pp. 349–385, 1981.

[23] H. P. Moravec, "Obstacle avoidance and navigation in the real world by seeing a robot rover," Stanford Artif. Intell. Lab. Memo 340, 1980.

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

[25] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, pp. 139–154, 1985.

[26] L. H. Quam, "Hierarchical warp stereo," in *Proc. DARPA Image Understanding Workshop* (New Orleans, LA), 1984, pp. 149–155.

[27] S. Ullman, *The Interpretation of Visual Motion.* Cambridge, MA: MIT Press, 1979.

[28] A. Verri and T. Poggio, "Against quantitative optical flow," in *Proc. First Int. Conf. Comput. Vision* (London), 1987, pp. 171–180.

[29] A. M. Waxman, "An image flow paradigm," in *Proc. Workshop Comput. Vision: Representation Contr.* (Annapolis MD), 1984, pp. 49–57.

[30] J. Weng, T. S. Huang, and N. Ahuja, "Motion and structure from two perspective views: Algorithm, error analysis and error estimation," *IEEE Trans. Patt. Anal. Machine Intell.* vol. 11, no. 5, pp. 451–476, 1989.

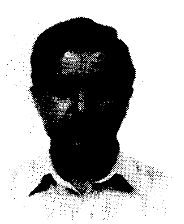
[31] ——, "Motion from images: Image matching, parameter estimation, and intrinsic stability," in *Proc. IEEE Workshop Visual Motion* (Irvine, CA), 1989, pp. 359–366.

[32] J. Weng, N. Ahuja, and T. S. Huang, "Optimal motion and structure estimation," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.* (San Diego, CA), 1989, pp. 144-152.

[33] O. A. Zuniga and R. M. Haralick, "Corner detection using the facet model," *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1983, pp. 30–37.

Juyang Weng (S'85–M'88) received the B.S. degree from Fudan University, Shanghai, China, in 1982 and the M.S. and Ph.D. degrees from the University of Illinois, Urbana-Champaign, in 1985 and 1988, respectively, all in computer science.

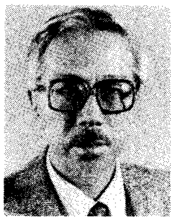
From September 1984 to December 1988, he was a research assistant at the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign. In the summer of 1987, he was employed at IBM Los Angeles Scientific Center. Since January 1989, he has been a researcher at the Centre de Recherche Informatique de Montréal, Canada, while he has also been associated with Ecole Polytechnique de Montréal. Since October 1990, he has been a visiting assistant professor at the University of Illinois, Urbana-Champaign. His current research interests include computer vision, image processing, object modeling and representation, parallel architecture for image processing, autonomous navigation, and artificial intelligence.



Narendra Ahuja (S'79–M'79–SM'85–F'92) received the B.E. degree with honors in electronics engineering from the Birla Institute of Technology and Science, Pilani, India, in 1972, the M.E. degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1974, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1979.

From 1974 to 1975, he was Scientific Officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979, he was with the Computer Vision Laboratory, University of Maryland, College Park. Since 1979, he has been with the University of Illinois at Urbana-Champaign, where he is currently a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute. His interests are in computer vision, robotics, image processing, and parallel algorithms. He has been involved in teaching, research, consulting, and organizing conferences in these areas. His current research emphasizes integrated use of multiple image sources of scene information to construct 3-D descriptions of scenes, the use of acquired 3-D information for object manipulation and navigation, and multiprocessor architectures for computer vision.

Dr. Ahuja was selected as a Beckman Associate in the University of Illinois Center for Advanced Study during 1990–1991. He received the University Scholar Award in 1985, the Presidential Young Investigator Award in 1984, the National Scholarship from 1967–1972, and the President's Merit Award in 1966. He coauthored the book *Pattern Models* (New York: Wiley, 1983) with B. Schachter. He is an Associate Editor of the journals *Pattern Analysis and Machine Intelligence; Computer Vision, Graphics, and Image Processing*, and *Journal of Mathematical Imaging and Vision.* He is a member of the American Association for Artificial Intelligence, the Society of Photo-Optical Instrumentation Engineers, and the Association for Computing Machinery.



Thomas S. Huang (S'61–M'63–SM'76–F'79) received the B.S. degree in electrical engineering from National Taiwan University, Taipai, Taiwan, China, and the M.S. and Sc.D degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge (MIT).

He was on the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973 and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Ilinois at Urbana-Champaign, where he is now Professor of Electrical and Computer Engineering and Research Professor at the Beckman Institute and the Coordinated Science Laboratory. During his sabbatical leaves, he has worked at the MIT Lincoln Laboratory, the IBM Thomas J. Watson Research Center, and the Rheinishes Landes Museum, Bonn, West Germany, and held Visiting Professor positions at the Swiss Institutes of Technology in Zurich and Lausanne, the University of Hannover, West Germany, and INRS-Telecommunications of the University of Quebec in Montréal, Canada. He has served as a consultant to numerous industrial firms and government agencies both in the United States and abroad. His professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 10 books and over 200 papers on network theory, digital filtering, image processing, and computer vision.

Dr. Huang is a Fellow of the Optical Society of America. He received a Guggenheim Fellowship (1971–1972), the A.V. Humboldt Foundation Senior U.S. Scientist Award (1976–1977), a Fellowship from the Japan Society for the Promotion of Science (1986), the IEEE Acoustics, Speech, and Signal Processing Society Technical Achievement Award (1987), and the University Scholar Award from the University of Illinois at Urbana-Champaign (1990). He is an Editor of the international journal *Computer Vision, Graphics, and Image Processing*, Editor of the *Springer Series in Information Sciences* published by SpringerVerlag, and Editor of the *Research Annual Series in Computer Vision and Image Processing* published by JAI Press.