FEATURE EXTRACTION AND MATCHING AS SIGNAL DETECTION*

XIAOPING HU

Sun Microsystems Computer Company, 2550 Garcia Ave
MS MPK 14-203, Mountain View, CA 94043, USA
email: xiaoping.hu@Eng.Sun.Com

NARENDRA AHUJA

Beckman Institute and Department of Electrical and Computing Engineering
University of Illinois at Urbana-Champaign
405 N. Mathews Avenue
Urbana, IL 61801, USA
email: ahuja@vision.csl.uiuc.edu

This paper discusses detection and matching of arbitrary image features or patterns. The common characteristics of feature extraction and matching are summarized which show that they can be considered as special cases of a more general problem—signal detection. However, the existing signal detection theories do not solve feature extraction and matching problems readily. Therefore, a general formulation of feature extraction and matching as a problem of signal detection is presented. This formulation unifies feature extraction and matching into a more general framework so that the two can be better integrated to form an automatic system for image matching or object recognition. Following this formulation, guidelines for designing algorithms for detection or matching of arbitrary image features or patterns which can be easily implemented or reconfigurated for many practical applications are derived. Sample algorithms resulting from this formulation and the associated experimental results with real image data are provided which demonstrate the performance and robustness of the methods.

Keywords: Detection, disparity constraint, extraction, feature, geometric constraint, matching, rigidity constraint, signal detection.

1. INTRODUCTION

This paper presents a general formulation of feature extraction and matching as a problem of signal detection. We shall consider detection of arbitrary features and template matching in which a window image or a pattern is to be matched to another image or pattern. Matching algorithms can be classified into two kinds: intensity-based matching and feature-based matching. In this paper, intensity based matching is treated as a waveform matching problem[40] and feature-based matching is considered through *chamfer matching*[2,7] as an example.

Traditionally, feature extraction and matching have been considered as two different processes. This causes the problem that the extracted features and their representations may not be particularly good for matching purposes. However, the

*The support of the National Science Foundation and Defense Advanced Research Projects Agency under grant no. IRI-89-02728, and the US Army Advanced Construction Technology Center under grant no. DAAL 03-87-K-0006, is gratefully acknowledged.

International Journal of Pattern Recognition and Artificial Intelligence Vol. 8 No. 6 (1994) 1343–1379

two processes have similar characteristics and requirements and can be considered as special cases of a more general process—signal detection: in the case of feature extraction, idealized features (such as edges, corners, and spots) or image patterns (such as characters, parts flaws), are to be detected and located in the images; in the case of image matching, particular image waveforms or patterns in one image (or in a template) are to be located (or registered) in the other image.

In this paper, a waveform is a piecewise continuous two-dimensional function $f(x, y)$ (neglecting the digitization effect). For pattern, we focus on edge images. We discuss the principles for optimal detection which guide the design of a method. An optimal method often corresponds to a particular combination of these principles. Detailed discussion on the intuition behind these principles and how to use them in applications is presented. Then, five kinds of measures for comparing the similarity of two signals are summarized. Mathematical descriptions of these measures and a comparison of them are given. In general, different similarity criteria lead to different feature extraction and matching algorithms. The merits and shortcomings of these measures are briefly discussed.

We then formulate feature extraction and matching as problems of signal detection. Schemes for general feature extraction and matching are proposed and sample algorithms and experimental results are presented. Through examples we show how to develop robust and efficient algorithms according to the formulation. There are several useful results of formulating feature extraction and matching as a signal detection process:

1. Some existing signal detection and estimation concepts and methods such as signal-to-noise ratio and matched filter can be applied.
2. All algorithms for feature extraction and matching can be developed on the same basis and hence easily integrated.
3. The formulation yields a family of methods, instead of a single algorithm by most other formulations, for detecting and matching arbitrary features.

Situations leading to mismatch are first discussed, which are common to all matching methods. Then constraints and heuristics that can be used to remove ambiguous matches are developed, including geometric and rigidity constraints and new disparity constraints. By a geometric argument, it can be shown that the application of these constraints can essentially rule out all mismatches except for a rare situation which occurs with zero probability. An integrated use of these constraints and image intensity properties is a key to a successful solution of the general matching problem. Based on these ideas, robust matching algorithms using intensity images and/or edge images have been developed for matching point features and edges. Descriptions of the algorithms and experiments with diverse real image data are provided to show that the matching algorithms yield only correct matches for a large class of images related by small motions (in this paper, a correct match is a match within two pixels distance to the true match).

Section 2 discusses general aspects of signal detection such as general principles and similarity measures. Section 3 presents a general formulation of feature extrac-

tion and matching as signal detection, which is followed by the resulting edge/spot detection and matching algorithms and experimental results (Secs. 4–6). Finally, Sec. 7 summarizes the paper.

2. GENERAL ASPECTS OF SIGNAL DETECTION

2.1. The Traditional and Extended Signal Detection Problem

A typical physical model for the traditional signal detection[10,39] problem is that a waveform which consists of one of two possible signals corrupted by additive noise, is observed. The objective is to decide which signal is present. This model cannot be directly applied to feature extraction and matching as will be seen soon.

A more general signal detection model that characterizes the feature detection and matching problems is as follows. Assume $I(\mathbf{x})$, $\mathbf{x} \in D$, is an observed image and $T(\mathbf{x})$, $\mathbf{x} \in d$, is a template to be detected or matched. The problem is to determine whether for some θ and $\mathbf{u}$, the window image $I(\mathbf{r}_\theta \mathbf{x} + \mathbf{u})$ represents the template $T(\mathbf{x})$ corrupted with some noise $n(\mathbf{x})$, or whether the following equation

$$I(\mathbf{r}_\theta \mathbf{x} + \mathbf{u}) = T(\mathbf{x}) + n(\mathbf{x}), \quad \mathbf{x} \in D, \tag{2.1}$$

holds for some θ and $\mathbf{u}$, where

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \mathbf{r}_\theta = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}. \tag{2.2}$$

Here, in addition to the unknown location parameters $\mathbf{u}$, there is still an orientation parameter θ to be determined. Clearly, if there is no restriction on the noise $n(\mathbf{x})$, then, for any θ and $\mathbf{u}$, Eq. (2.1) can be satisfied by some $n(\mathbf{x})$. Thus, some similarity measure M between I and T has to be used to determine whether I can be considered as T corrupted with noise n. That is, if $M(I(\mathbf{r}_\theta \mathbf{x} + \mathbf{u}), T(\mathbf{x}))$ is larger (or smaller) than a threshold, then I represents T; otherwise, not. The use of a similarity measure often makes the noise model (2.1) invisible. Therefore, *the extended signal detection problem is to determine whether the signal truly exists in the observation and if so, where it is truly located.*

In this paper, we will only formulate feature extraction and matching as problems of the extended signal detection. Numerically optimal solutions are not attempted since optimal solutions depend on the criteria used.

2.2. Principles for Optimal Detection

Before we obtain a general formulation, we need to know the common requirements for feature extraction and matching. For feature extraction, the three criteria that Canny[8] proposed for edge detection can all be applied: *good detection principle* (which states that the detector should indicate the presence of the signal when it is indeed present), *good localization principle* (which states that the detector should locate the position of the signal close to the true position), and *unique response principle* (which states that the detector should report one and only correct response

for each presence of a signal). The same criteria also applies to waveform matching. The only difference is that for matching, a unique and correct match is sought over the whole image, but not simply over the region containing the signal. This characteristic makes matching more difficult than feature extraction.

A fourth principle is called the *non-maximum (or non-minimum) suppression principle* which is often used to determine the location of a signal. This principle considers the local maximum (or minimum) of the responses of the detector in a region containing the signal as the location of the signal.

Another useful but often ignored principle is the *zero-mean principle* which states that if only the shape and not the absolute values of a signal is of interest, then the model signal used for detection should have a zero-mean. This is very important when the correlation method is used to determine the similarity of two waveforms, since large DC components in both waveforms may suppress the shape information present in the waveform. This can be easily seen from the following example. The correlation coefficient of signal $A + sin(t)$ and signal $A - sin(t)$ over $(0, 2\pi)$ is $(A^2-1)/(A^2+1)$, while the correlation coefficient of $sin(t)$ and $-sin(t)$ over $(0, 2\pi)$ is -1. When A has a magnitude larger than 1, error may result if the correlation coefficient of $A + sin(t)$ and $A - sin(t)$ is used to determine if the two signals are similar.

These principles will be used to guide the development of optimal feature detection algorithms and matching algorithms. However, the non-maximum suppression and zero-mean principles are not the objectives in feature extraction and matching, but rather useful means to attain them.

2.3. Similarity Measures

One common problem for feature extraction and matching is to determine the similarity of two signals: one template, the other input. This raises the need to examine similarity measures.

There are many well known measures for comparing the similarity of two waveforms. The oldest one is probably the distance measure described by the Riemann integral. Assume $W(x, y)$ and $S(x, y)$ are two two-dimensional signals defined in the same domain Δ. Then the squared distance of the two signals is defined by

$$M_1(W, S) = \int_\Delta (W(x, y) - S(x, y))^2 dxdy\,, \tag{2.3}$$

This measure examines the differences of the shapes and magnitudes of two waveforms and is therefore a very stringent one. Another is the correlation coefficient of the two signals

$$M_2(W, S) = \frac{\int_\Delta W(x, y)S(x, y)dxdy}{\sqrt{\int_\Delta W^2(x, y)dxdy}\sqrt{\int_\Delta S^2(x, y)dxdy}}\,. \tag{2.4}$$

This measure examines only the shapes of two waveforms and ignores the difference of magnitudes. Sometimes, the normalization terms in the denominator are replaced

by 1. If we consider $W(x,y)$ as the sum of $S(x,y)$ with a noise signal $n(x,y)$, i.e.

$$W(x,y) = S(x,y) + n(x,y)\,, \tag{2.5}$$

then a third one is the signal-to-noise ratio

$$M_3(W,S) = \frac{\int_\Delta S^2(x,y)dxdy}{\int_\Delta n^2(x,y)dxdy}\,. \tag{2.6}$$

This measure examines the relative strengths of the signal and the noise. Therefore, it can be used not only as a similarity measure but also as a confidence measure.

The above measures give different quantitative assessments of the similarity of two signals. One problem with these measures is that they do not reflect the spatial structure of deviation. As a result, the noise tends to influence the similarity measure collectively and uniformly. The fourth and fifth measures, discussed below, depend on the structure of deviation.

The fourth one measures the distance of two sets of points (cf. Ref. 32). We need to first introduce a set distance. Assume W and S are two sets of 2-D points. Let R be a relation (or property) defined among the points. The distance of a point $\alpha = (u,v)$ in W to the set S is defined as

$$D_R(\alpha, S) = min\,(\|\alpha - \beta\| : \alpha R\beta,\ \beta \in S)\ , \tag{2.7}$$

where $\|\cdot\|$ denotes the Euclidean distance, $\alpha R\beta$ means that α and β are related by property R, and $\beta \in S$ means that β is a point of S. The above equation states that, among all points in S that satisfy relation R with α, if β has the shortest distance with α, then $\|\alpha - \beta\|$ is the distance of α to set S. If there is no point in S that satisfies relation R with α, $D_R(\alpha, S)$ is defined as infinity. Then the fourth similarity measure is the distance of set W to set S defined as

$$M_4(W,S) = D_R(W,S) = \sqrt{\frac{1}{N_W}\sum_{\alpha \in W} D_R^2(\alpha, S)}\,, \tag{2.8}$$

where N_W is the number of points in W. $D_R(S,W)$ can be similarly defined, though $D_R(W,S) \neq D_R(S,W)$ in general. That is, the commutative law does not hold for D_R. To obtain a metric of symmetry, one can use the *Hausdorff* distance measure

$$H_R(W,S) = D_R(W,S) \vee D_R(S,W)\,, \tag{2.9}$$

where $A \vee B$ means $\max(A,B)$. However, D_R is more efficient and is satisfactory for general purposes. This fourth measure examines the similarity of two patterns consisting of points or lines. We call this structure deviation measure.

The fifth one extends the fourth measure to continuous waveforms. In this measure, the model signal $S(x,y)$ is treated as a three-dimensional surface and $(x, y, W(x,y))$, where $W(x,y)$ is the observed signal waveform, is considered as a three-dimensional point. Then, the sum of the minimum distances of points

$(x, y, W(x,y))$ to the surface $(x, y, S(x,y))$ over the window describes the *shape deviation* of the two waveforms $S(x,y)$ and $W(x,y)$. Mathematically, let Δ be the domain in which $S(x,y)$ and $W(x,y)$ are defined, and let

$$D(x,y;\, S) = \min_{(u,v)\in\Delta} \left\{ \sqrt{(x-u)^2 + (y-v)^2 + r(W(x,y) - S(u,v))^2} \right\}, \quad (2.10)$$

where r is a weighting factor regulating the relative importance of positional difference and waveform strength difference; then the fifth measure, called shape deviation measure, is defined as follows

$$M_5(W,S) = \int_\Delta D^2(x,y)dxdy\,. \quad (2.11)$$

All measures but the fifth have been used in the previous research in different forms for either feature detection or matching (e.g. Refs. 3, 8, 16, 31, 32). All of them can be used for matching, but for feature extraction, M_2 has been overwhelmingly used because only M_2 can adapt to signals of different magnitudes without introducing a scale parameter. There are many other measures (such as those used in signal detection) which are defined by probabilities and maximum likelihood. However, we will not discuss them in this paper.

To simplify the discussion in the rest of this paper, we assume that the similarity measures have been properly signed (e.g. M_1, M_4, and M_5 are negatively signed) so that maximum values are to be sought for any similarity measure.

3. FEATURE EXTRACTION AND MATCHING AS SIGNAL DETECTION

In this section we discuss the relationships of feature extraction and matching with the general signal detection problem. Since there exist no standard solutions for the general signal detection problem, formulating feature extraction and matching as a problem of signal detection does not lead to the immediate solution of the existing problems. However, the formulation provides the guidelines for designing robust algorithms for feature extraction and matching.

3.1. A Formulation for Feature Extraction

There are two types of feature extraction problems. In the first, feature extraction itself is the goal (as is the case in parts flaw inspection); in the second, feature extraction is a means for another higher level task such as matching. In general, *given an image $I(\mathbf{x})$ of size $R \times C$ and certain property P (which is often represented by an idealized signal T),*

1. *The first type of feature extraction problem is to choose a model signal (or detector) $S(\mathbf{x})$ of size $m \times n$ and a similarity measure M such that a window image centered at $\mathbf{u}$ has the property P when and only when (1) $(\mathbf{u}, \theta)$ is a local maximum of the following function*

$$M(\mathbf{u},\theta) = M(S(\mathbf{x}), I(\mathbf{r}_\theta \mathbf{x} + \mathbf{u})) \quad (3.1)$$

for some angle θ and (2) $M(\mathbf{u},\theta)$ is above a certain threshold.

2. *The second type of feature extraction problem is to choose a model signal (or detector) $S(\mathbf{x})$ of size $m \times n$ and a similarity measure M such that all local maxima of the following function*

$$M(\mathbf{u}, \theta) = M(S(\mathbf{u}), I(\mathbf{r}_\theta \mathbf{x} + \mathbf{u})) \tag{3.2}$$

with $M(\mathbf{u}, \theta)$ above a certain threshold have the property P.

That is, in the second type, missing a feature is not a problem but wrong detection is. In this paper we mainly discuss the second type of feature extraction problem.

The properties or the idealized signals are often given by applications or need to be determined by a human. The similarity measure can often be determined in advance by the task or later by choosing the one from a set of measures that yields the best feature detector. The threshold is determined by the higher level process (e.g. matching) such that sufficient features are detected. Therefore the major problem is the design of the model signal or the detector. This problem is a typical signal detection problem.

For feature extraction, M_2 has been overwhelmingly used since only M_2 is adaptive to signals of different magnitudes. When M_2 is used as the sole optimality criterion, the design of the model signal for a given ideal feature is equivalent to the design of the optimal filter which, for Gaussian noise model, has a standard solution known as *matched filter*.[10,39] However, in practice, many other (often conflicting) concerns, such as the principles discussed in Sec. 2.2, are to be considered in the process of feature extraction. Therefore, a particular optimal detector often corresponds to a particular weighting of these principles. Instead of proposing a particular optimality criterion (cf. Refs. 8, 38) and obtaining the optimal detector for a particular feature, here we shall consider only the requirements that each principle imposes on the model signal. Such an understanding allows fast implementation of a robust, if not the best, detector for any feature.

Let us use the design of model signals of some symmetric features such as edges and spots as an example to illustrate the ideas. Many features such as edges have orientations, for which the model signals should also have orientations. We will consider only model signals of a particular orientation which can be rotated to yield model signals of other orientations. For convenient discussion, let us define

$$M(\mathbf{u}) = M(u, v) = \max_\theta (M(\mathbf{u}, \theta)) . \tag{3.3}$$

$M(\mathbf{u})$ represents the maximum similarity between the model signal $S(\mathbf{x})$ and the window image centered at $\mathbf{u}$. We shall call $M(\mathbf{u})$ (between the model signal and a window image at $\mathbf{u}$) as the similarity function.

The zero-mean principle imposes a restriction that the sum of the model signal waveform over the window must be zero. That is,

$$\int_\Delta S(x, y) dx dy = 0 , \tag{3.4}$$

where Δ denotes the signal window.

The good detection principle requires that if $\mathbf{u}_0$ is the location of a window image of property P, the similarity measure $M(\mathbf{u}_0)$ is larger than the threshold δ, or

$$M(\mathbf{u}_0) = \max_{\theta}(M(S(\mathbf{x}), I(\mathbf{r}_\theta \mathbf{x} + \mathbf{u}_0))) > \delta\,. \tag{3.5}$$

The non-maximum suppression principle requires that the derivatives of the similarity function $M(\mathbf{u})$ between the model signal and a window image of property P centered at $\mathbf{u}_0$ have zero values at $\mathbf{u}_0$. This requirement can be expressed as follows

$$\left.\frac{\partial M}{\partial u}\right|_{\mathbf{u}_0} = \left.\frac{\partial M}{\partial v}\right|_{\mathbf{u}_0} = 0\,. \tag{3.6}$$

The good localization principle requires that if $\mathbf{u}_0$ is the location of a window image of property P, then the similarity function $M(\mathbf{u})$ has a sharp peak at $\mathbf{u}_0$. The mathematical expression of this requirement is not unique and varies according to the type of signal to be detected. The following method is based on the Hessian matrix of the similarity function, which is defined as

$$\mathbf{H}_M(u,v) = \begin{bmatrix} \dfrac{\partial^2 M(u,v)}{\partial u^2} & \dfrac{\partial^2 M(u,v)}{\partial u \partial v} \\ \dfrac{\partial^2 M(u,v)}{\partial u \partial v} & \dfrac{\partial^2 M(u,v)}{\partial u \partial v} \end{bmatrix}. \tag{3.7}$$

Let λ_1 and λ_2 be the eigenvalues of $\mathbf{H}_M$ at $\mathbf{u}_0$ with $|\lambda_1| \geq |\lambda_2|$. If the model signal is shift-invariant along a direction (e.g. a 2D step edge is shift-invariant along the edge direction), $\mathbf{H}_M$ is degenerate at $\mathbf{u}_0$ and $\lambda_2 = 0$. In this case, the good localization principle requires that $|\lambda_1|$ has a large value so that $M(u,v)$ has a sharp ridge. If the signal is not shift-invariant along any direction, the good localization principle requires that both $|\lambda_1|$ and $|\lambda_2|$ have large values so that $M(u,v)$ has a sharp peak at $\mathbf{u}_0$. In other words, the sharpness of the similarity function $M(u,v)$ at $\mathbf{u}_0$ can be evaluated by the eigenvalues of the Hessian matrix of $\mathbf{H}_M$ at $\mathbf{u}_0$. *To make $M(u,v)$ have a sharp peak or ridge at $\mathbf{u}_0$, in general the model signal should consist of only smoothed edge patterns and should not contain nonzero flat plateaus.*

The unique response principle requires that if $\mathbf{u}_0$ is the location of a window image of property P, then the similarity function $M(\mathbf{u}_0)$ have only one maximum within a certain region around $\mathbf{u}_0$. In general, this requires that the model signal have a smooth waveform. To detect features in unsegmented images, the signal waveform should also degrade the impact of the peripheral areas of the signal. Otherwise, random noise and neighboring image content may produce artificial response peaks in the similarity function. Degrading of the impact of the peripheral areas of a signal can be performed by convolving the signal with a hat-shape filter.

In practice, it is often convenient to characterize the image property P by an idealized signal $T(\mathbf{x})$ and consider all other window images of property P as $T(\mathbf{x})$ corrupted with additive noise $n(\mathbf{x})$ (cf. Eq. (2.1)). Then, if a noise model is assumed and a particular weighting of the principles is given, numerical methods can be used

to obtain an optimal detector for a given feature. However, we will not discuss further on the effect of noise.

From the above discussion, we conclude that the following steps can be used to build a general feature detector:

1. Characterize the feature of the desired properties by an ideal signal.
2. Smooth the edges of the ideal feature; degrade the nonzero plateaus; convolve the resulting waveform with a hat-shaped filter to reduce the influence of the peripheral areas.
3. Normalize the waveform so that it has a zero mean.
4. Check if the similarity function between the waveform and the ideal feature has a sharp peak or ridge at the center of the window. If so, the resulting waveform is the desired feature detector; otherwise other methods have to be used.

When this method is applied to the detection of edge, spot, roof, thin bar, or wide bar using M_2 as the similarity measure, the shapes of the resulting optimal detectors all comply with either numerically obtained optimal detector (Canny,[8] Shen and Castan[38]) or impulse responses of biological receptive fields (Marr[26]). Edge and spot detection algorithms resulting from this formulation will be presented as examples. The elegance of this method is that it results in a family of detectors for various features with the same simple idea. In principle, this method can be applied to detect any features that can be defined by a waveform. However, for some features which cannot be represented by a single waveform, other methods may be more efficient. For example, a corner has two degrees of freedom: one is the orientation, the other is the angle of the corner. Therefore, to detect corners of any angle and any orientation, other methods that do not use templates may be more efficient.[18]

3.2. A Formulation for Matching

Matching has been a difficult problem that remains not well-solved. There are many kinds of matching problems. Let us restrict our discussion to the matching of two views of a scene containing rigid objects with the motions unknown. The two-view matching problem can be formally stated as: *Given two images of a scene, register a point in one image to another point in the other image such that the two image points correspond to the same scene point.* Two points are *matched* if the similarity measure about them passes a preset threshold. They are *correctly matched* if they correspond to the same scene point. Clearly, this problem also falls within the extended signal detection framework. However, matching is more difficult than feature extraction, because (1) for matching, a unique and correct correspondence for a given point is sought in the whole image, while for feature extraction, many responses can be admitted in different locations of the image; and (2) the number of waveforms involved in matching is much larger than that involved in feature extraction.

We shall consider only feature matching which differs from pixelwise matching (e.g. Refs. 40 and 41), optical-flow-based matching (e.g. Refs. 1, 17 and 30), and

the (affine) transformation method (e.g. Refs. 4, 11, 24) in that correspondences are obtained for feature points only. In this sense, our discussion here is closest to the works of Scott and Longuet-Higgins,[34,35] Shapiro and Brady,[37] Sethi and Jain,[36] Salari and Sethi,[33] and Cheng and Aggarwal.[9]

For matching, all five similarity measures can be used. The first three examine the intensity similarities of two images, while the fourth and fifth ones examine the spatial structure of deviation between two edge or intensity images. Clearly, different measures result in different matching methods. The five principles described earlier also apply to matching. We now discuss the requirements of these principles on matching.

The zero-mean principle requires that the two waveforms to be matched have zero means. Therefore any two window images should be normalized (by subtracting the window mean) before they are matched. We must emphasize here that the zero-mean principle has mostly been ignored in the correlation based matching algorithms, leading to poorer performance. This principle may not apply to feature-based matching.

The good detection principle requires that the similarity measure must accommodate perspective image distortion and noise. This problem is dealt with by choosing a proper similarity measure. For example, if $I'(\mathbf{x}) = I(\mathbf{x}) + n(\mathbf{x})$ for a reasonable amount of noise $n(\mathbf{x})$, then the chosen similarity measure M between I and I' should still be high. Much of the research on matching is spent on finding a similarity measure which uses view-point invariant properties.

The good localization principle suggests that only well-defined features (such as edges or corners) whose location accuracy is generally high should be used for matching. The following method can be used to determine the suitability of a feature for matching. For a given feature $S(\mathbf{x})$, let its *auto-similarity* be defined as

$$R(\mathbf{u}) = R(u, v) = M(S(\mathbf{x}), S(\mathbf{x} + \mathbf{u})) . \tag{3.8}$$

If $R(\mathbf{u})$ has a sharp peak (or a sharp ridge when epipolar constraint is available) at $(0, 0)$, then $S(\mathbf{x})$ is good for matching; otherwise, not. Again, the sharpness of $R(\mathbf{u})$ can be determined using the Hessian matrix of $R(\mathbf{u})$.

The non-maximum suppression principle can be used to select candidate matches as follows. Assume $\mathbf{p}_i$, $i = 1, \ldots, N_1$ is the set of detected features in the first image and $\mathbf{p}'_j$, $j = 1, \ldots, N_2$ is the set of detected features in the second images. Let $Q(i, j) = M(\mathbf{p}_i, \mathbf{p}'_j)$ be the matching quality matrix defined by the similarity measures between the two sets of points. If $\mathbf{p}_i$ and $\mathbf{p}'_j$ are the correct match of each other, then $Q(i, j)$ should be both a row maximum and a column maximum in the matrix. A more practical method that uses two thresholds δ_1 and δ_2 and the non-maximum suppression principle to select potential matches is illustrated in Fig. 1.

For matching, the unique response principle states that a feature in the first image can have at most one correct match in the second image and vice versa. The non-maximum suppression principle can be used to find the potential unique

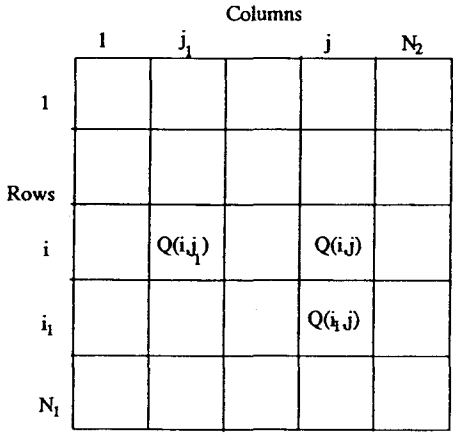


Fig. 1. A pair (i, j) is a candidate match if and only if (a) $Q(i,j) > \delta_1$ (b) $Q(i,j)$ is the maximum in the ith row and jth column in the matching quality matrix Q (c) $Q(i,j)$ is larger than the second best match $Q(i, j_1)$ in the ith row by an amount δ_2 (d) $Q(i,j)$ is larger than the second best match $Q(i_1, j)$ in the jth column by an amount δ_2.

matches. However, these potential matches, though unique, may not be correct. Mismatch is a serious problem in matching. To overcome this problem, we need first to examine the common causes of mismatching so that proper methods can be developed to remove mismatches.

The main causes of mismatching include the presence of similar (e.g. periodic) patterns, occlusion, image distortion, and inconsistent detection of point features in two images (e.g. some points are detected in the first image but not detected in the second image), and noise. The consequence of these causes is that some points in the first image are wrongly matched to other points in the second image which have similar local intensity or edge environments. The mismatched points have two characteristic properties. One property is that if point $\mathbf{p}$ in the first image has a mismatch $\mathbf{p}'$ in the second image, then $\mathbf{p}'$ is dislocated from its true position. This dislocation causes a distortion in the geometric structures defined by the matched points in the first and second images, respectively. The other is that if a point $\mathbf{p}$ is mismatched to $\mathbf{p}'$, then, in general, either $\mathbf{p}$, or $\mathbf{p}'$, or both will have a good match with another point if it were detected. We can therefore use these two properties to remove all possible mismatches, irrespective of how the mismatches are caused.

The idea is to use the structural information contained in the feature points in addition to the local intensity or edge image similarities. This information includes the 2D geometry defined by the feature points and 3D rigidity. A geometrical relationship of different objects in an image is more fundamental and reliable than intensity values since intensities are more liable to change. But it is computationally intractable to use a geometric relationship alone to match a large number of point features. A more profound invariant property of feature points is 3D rigidity which requires that the shape and size of a rigid object be invariant from view to view.

Another efficient constraint for removing mismatches is the *disparity constraint* to be discussed later. The methods for using the geometric, rigidity, and disparity constraint to enforce correct matches are discussed in detail later. Here it suffices to know that it is more difficult to enforce the unique response principle in matching than in feature extraction.

From the above discussion, we propose the following method for general feature matching. The idea is to first obtain a set of unique matches using intensity-based or edge-based matching methods. Then, geometric consistency and rigidity are enforced to identify and remove the possible mismatches. However, mismatches may still survive over the geometric consistency and rigidity examinations for points detected in only one image. Then all matched points need to be re-examined to see if they can be rematched and yet preserving the geometrical similarity. This is a very time-consuming process and an efficient way for doing this is to use the disparity constraint.

The general matching method can be divided into five stages:

1. Point Feature Detection: detect point features in both images.
2. Wave form or Feature-Based Matching: obtain candidate pairs of matches based on similarity of local intensities or edges.
3. Geometry-Based Elimination: identify and remove points that do not preserve 2D geometric consistency in the image plane.
4. Rigidity-Based Elimination: apply rigidity constraint to eliminate such points that do not construct a rigid triangle with their two closest neighbors.
5. Disparity-Based Elimination: apply the disparity constraint to eliminate all point pairs for which one of the points could be rematched and yield a disparity value possessed by some other matched pair.

Algorithms resulting from this method are presented in a later section.

3.3. Integration of Feature Extraction and Matching

In this formulation we consider feature extraction and matching as subsequent processes and extract features according to the needs of matching. This is implemented as follows. First, the criteria used for feature extraction are the same as those used for matching such that the extracted features are suitable for matching. Second, many parameters such as thresholds used in feature extraction are determined by the matching algorithms. Third, features are quantized and represented according to the requirements of matching. Fourth, features are extracted continuously across images to ensure a high reappearance rate by using thresholding methods insensitive to lighting conditions. In this way, feature extraction and matching have been combined together to yield an automatic system for a general matching purpose.

4. ALGORITHMS FOR FEATURE EXTRACTION

In this section we apply the general formulation of feature extraction to edge and spot detections.

4.1. An Adaptive Edge Detection Scheme

To adapt to edges of different strengths, M_2 (with the normalization terms dropped) is used as a similarity measure. In the case of edge detection, an ideal one-dimensional edge is the step edge (Fig. 2(a)) which has been commonly used as the model signal in the previous research. The unique response principle requires that the edge should be smoothed as shown in Fig. 2(b), otherwise additional maxima may be caused by noise. Coincidently, most real image edges have such a shape due to the blur effect which is often modelled by a step edge convolved with a Gaussian kernel.[22,29] A detector represented by Fig. 2(b) may result in two maxima (of opposite directions) for a slope-shaped edge (similar to the double-edge effect of zero-crossings in Fig. 12(d)), due to the influence of the surrounding image content. To ensure a unique response, an effective way is to degrade at the peripheral area of the model signal so that the faraway image content will not affect local edge detection. This can be easily performed by convolving the waveform in Fig. 2(b) with a hat-shaped filter, resulting in a waveform of the shape in Fig. 2(c). The good localization principle suggests that the similarity function (in this case, the correlation function) between the model waveform and the ideal waveform should have a sharp peak/ridge at (0,0). The correlation function between a model signal of the shape in Fig. 2(c) and the step edge indeed has a sharp ridge centered at $(0, 0)$. That means any filter of an impulse response of the shape in Fig. 2(c) should serve as an edge detector with a reasonably good performance. Such a shape of impulse response can also be found in biological visual receptive fields.[12,14,15,20,21,23,26,28]

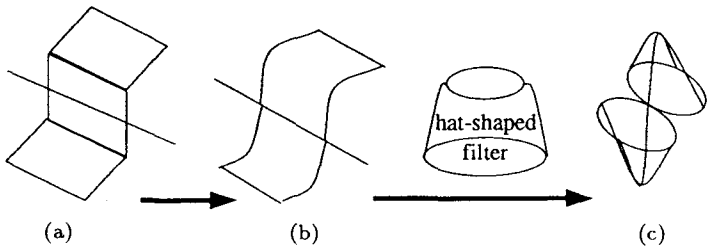


Fig. 2. The design process for edge detector. (a) Ideal step edge (b) smoothed edge (c) general shape of an optimal edge detector.

Canny[8] obtained his optimal detectors using his optimality criterion, while Shen and Castan[38] obtained optimal detectors of shape shown in Fig. 2(b) using another optimality criterion. The particular waveform of the optimal detector depends on the optimality criterion to be used. Although the waveform in Fig. 2(c) and the biological retina receptive responses can be very well approximated by many low-pass filters (e.g. the Butterworth filter), these waveforms have been overwhelmingly modelled by a single Gabor filter,[14,23] DoG (difference of Gaussians,[26,27] or ∇G (gradient of Gaussian,[26,27]). The elegance of the Gabor filter and Gaussian filter is that they achieve the lower bound of uncertainty for linear filters.[13,25] However, if we stay in the temporal domain to do edge detection, then this lower bound does not need to be reached to make the detector optimal.

We now present a complete algorithm. The algorithm in many aspects is similar to the Canny edge detector[8] but is significantly different in some other aspects. We first transform the original image to an edge strength image and an edge direction image. The edge directions are quantized for further use. Only local maxima orthogonal to the edge directions in the edge strength image are considered possible edges. The reason that we prefer maxima to zero-crossings[26] is that for large-window operators the zero-crossings are shifted from the positions of the maxima and the locations of the maxima are more robust against noise. Another reason is that using maxima instead of zero-crossings can threshold off some noisy edges because noisy edges tend to have smaller edge strengths. Probably the major reason is that the operator may introduce artificial zero-crossings which do not correspond to any edges visible in the image.

An automatic thresholding method is used to select the maxima of highest responses as edges. To do this, an edge/maximum ratio or a total number of edges needs to be specified by the user or determined by a higher-level process (such as matching). Then the histogram of the edge strengths of maxima are computed and used to determine the threshold such that the number of edges whose responses are above the threshold, is close to the desired number of edges. This method makes the algorithm adaptive to a wide range of images and eliminates most edges caused by noise. The combined use of window size, thresholds, and edge directions also makes it possible to select edges of different strengths, widths, and directions.

After thresholding, maxima of edge strengths larger than the threshold are classified as edges. To remove edges caused by random noise, we require that the nearby points along the edge direction have nearly the same edge strength. After thresholding and removing noise edges, sometimes connective edges become disrupted at some points due to various causes. To recover those edges at the disruption points, non-edge maxima that are, e.g. 2 pixels or less distance away from an edge and have nearly the same edge direction as that of the edge are also classified as edges. This is done recursively.

The stepwise algorithm below requires that the user or a higher level process specify several numbers: the edge/maximum ratio or the total number of edges, the minimum edge strength that is acceptable for edges, and the size of the window (σ of Gaussian filter). The output of the algorithm is an edge image with quantized values indicating the edgehood of the points and the directions of the edges.

1. For a given σ, apply the Gaussian operator

$$G(x,y) = exp\left(-\frac{x^2+y^2}{\sigma^2}\right) \tag{4.1}$$

to image $I(x,y)$ to find the edge directions

$$\mathbf{n} = \nabla(G * I)/|\nabla(G * I)|\,, \tag{4.2}$$

where $|\nabla(G * I)|$ is considered as the edge strength. The edge direction $\mathbf{n}$ is quantized into 16 values (16 is sufficient) over a 360° range instead of a 180° range (two edges of opposite direction means that they have different polarity,

e.g. a vertical edge is assigned to value 4 or 12 according to whether the left or right side of the edge is brighter). An easy way to compute $\nabla(G * I)$ is to use the following formula

$$\nabla(G * I) = (G_x * I)\mathbf{i} + (G_y * I)\mathbf{j}, \tag{4.3}$$

where **i**, **j** are the unit vectors along the x and y directions, and

$$G_x = \frac{\partial G}{\partial x} = -\frac{2x}{\sigma^2} exp\left(-\frac{x^2+y^2}{\sigma^2}\right), \quad G_y = \frac{\partial G}{\partial y} = -\frac{2y}{\sigma^2} exp\left(-\frac{x^2+y^2}{\sigma^2}\right). \tag{4.4}$$

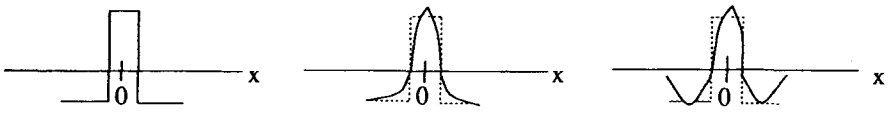
2. A local (edge strength) maximum (instead of zero-crossing) along the direction orthogonal to the edge direction is marked as a potential edge.
3. Let V be the variance of the edge strengths traversing a line of 5 points tangent to a candidate edge point in the edge direction, and let E be the average edge strength of the 5 points; then if

$$V < \varepsilon E, \tag{4.5}$$

the candidate point is preserved for further consideration, where ε is a preset constant (e.g. 0.1).
4. Compute the histogram of the edge strengths of the surviving edge candidates. Determine the threshold using the total number of edges or edge/maximum ratio.
5. A candidate of an edge strength larger than the threshold is considered an edge.
6. A maximum connected to a chosen edge is also considered an edge. This is done recursively.
7. Repeat Steps 1–6 for other size operators if desired.

4.2. Spot Detector

Similarly, the design procedure for a spot detector is shown in Fig. 3 through the one-dimensional profile of a spot. Figure 3(a) shows the profile of an ideal (disk) spot with zero-mean within a window. Figure 3(b) shows the shape of the spot with smoothed edges. Figure 3(c) shows the general shape of a spot detector (for small size spots) which compromises the good detection, good localization, and unique response principles. A particular two-dimensional filter of such profile is Laplacian–Gaussian, $\nabla^2 G(x, y)$.[5,6] However, this detector is inferior to the spot detector that follows.



(a) Ideal Spot (b) Typical Spot (c) Spot Detector

Fig. 3. Design process for the spot detector. (a) ideal spot (b) spot with smoothed edges (c) general shape of spot detector compromising the good detection, good localization and unique response principles.

In the above we have derived the general shape of the impulse response of a spot detector. We now present in detail a detector of analytical form for detecting spots of various sizes. For a spot of radius r_0, the following filter has the desired general shape of a spot detector shown in Fig. 3(c)

$$P(r) = c(r) \cdot exp\left(-\frac{(r-r_0)^2}{2\sigma^2}\right), \tag{4.6}$$

where r is the distance to the spot center, and $c(r)$ is a coefficient evaluated according to the following formula

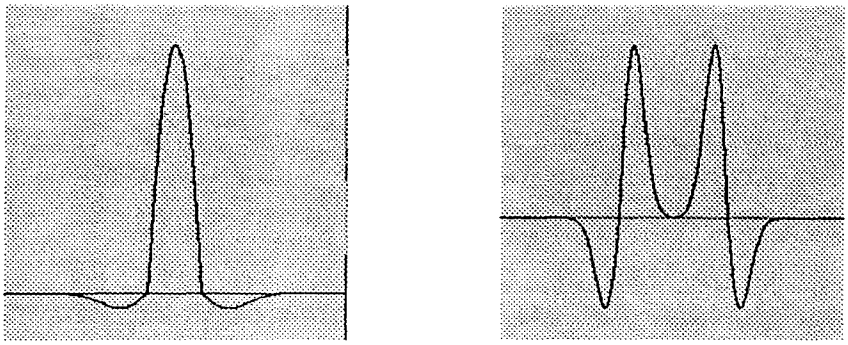
$$c(r) = \begin{cases} C_p \cdot (r_0 - r), & \text{if } r \leq r_0 \\ C_n \cdot (r_0 - r), & \text{if } r > r_0 \end{cases} \tag{4.7}$$

and C_p and C_n are positive constants to be determined using the zero-mean principle

$$\int_0^\infty P(r)dr = 0. \tag{4.8}$$

Clearly $P(r)$ is symmetric around the spot center.

The appearance of $P(r)$ is quite different for different r_0 and σ. Figure 4(a) shows the profile of $P(r)$ for $r_0 = 2.0$ and $\sigma = 2.0$, while Fig. 4(b) shows the profile of $P(r)$ for $r_0 = 4.0$ and $\sigma = 1.0$. The general shape of $P(r)$ is inbetween these two. The first type of waveform (Fig. 4(a)) is suitable for detecting spots of small size (e.g. $r_0 \leq 2$) while the second type (Fig. 4(b)) is suitable for detecting spots of large size (e.g. $r_0 > 2$). To detect spots of various sizes, a number of detectors for different sizes need to be used simultaneously.



(a) (b)

Fig. 4. Impulse responses of the spot detector: (a) The impulse response of $P(r)$ with $r_0 = 2.0$ and $\sigma = 2.0$. This type of waveform is suitable for detecting small spots. (b) The impulse response of $P(r)$ with $r_0 = 4.0$ and $\sigma = 1.0$. This type of waveform is suitable for detecting large spots.

5. ALGORITHMS FOR MATCHING

In this section we consider only the last four stages in the general matching scheme and assume that point features have been somehow detected in all images. We

shall discuss waveform and chamfer matching methods and the use of constraints to ensure a unique and correct match. Then we present a stepwise description of a matching algorithm.

First, let us discuss the choice of windows. A common practice of matching is to use window images centered at points of interest to evaluate the similarity of the points. The matching result would be the best if two corresponding window images contain the same image content. However, if the point of interest is at depth discontinuity, then a window centered at the point contains image contents from different surfaces such that some of the contents will be occluded in another view. Thus, to match points close to occlusion boundaries, other windows, containing but not centered at those points, should better be used. In our approach, four additional types of windows, with the point of interest located at the four corners of the windows, respectively, are also used. Each of these four windows respond best to corners of a specific orientation (Fig. 5) if the corners are at depth discontinuity. For convenience of discussion, these windows are represented by Δ_k, $k = 1, \ldots 5$, respectively. Thus, a total of five windows, are used at each point and the best matching quality obtained from the five windows is chosen to represent the quality of match for each pair. In general more windows give a more robust performance at the cost of more computation.

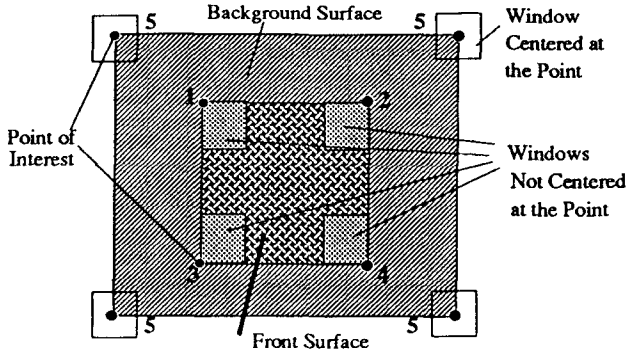


Fig. 5. If a corner is at depth discontinuity, the background surrounding the corner changes from view to view due to occlusion. Assume that the two largest squares in the figure represent two surfaces at different depths. Then, to match point features of types 1, 2, 3, or 4 shown in the figure, the windows that give the best matching results are shown in the figure.

5.1. Waveform Matching

The particular waveform matching algorithm to be discussed here is intensity-based matching. This method requires that the intensity change of the images be small.

Given a similarity measure M (M can be any of M_i, $i = 1, 2, 3, 5$) and the point features detected in each of the two images $I(\mathbf{x})$ and $I'(\mathbf{x})$, a *matching quality* Q is computed for all possible feature pairs for each window Δ_k. Let $I_i(\mathbf{x})$ be the window image at the ith feature point $\mathbf{u}_i$, i.e. $I_i(\mathbf{x}) = I(\mathbf{x} + \mathbf{u}_i)$, and let $\underline{I_i}^k$ be the

average intensity over a window Δ_k about $\mathbf{u}_i$. Let $P_i^k(\mathbf{x})$ denote the intensity of $\mathbf{x} \in \Delta_k$ with the window mean subtracted, i.e.

$$P_i^k(\mathbf{x}) = I_i(\mathbf{x}) - \underline{I}_i^k, \quad \mathbf{x} \in \Delta_k . \tag{5.1}$$

Similarly, for a feature point j in the second image I',

$$I'_j(\mathbf{x}) = I'(\mathbf{x} + \mathbf{u}'_j), \quad P_j'^k(\mathbf{x}) = I'_j(\mathbf{x}) - \underline{I_j'}^k, \quad \mathbf{x} \in \Delta_k . \tag{5.2}$$

Then, the matching quality $Q(i,j)$ for point i in the first image and point j in the second image is defined as

$$Q(i,j) = \max_{\Delta_k} \left\{ \max_{\theta} M\left(P_i^k(\mathbf{x}), P_j'^k(\mathbf{r}_\theta \mathbf{x}')\right) \right\}, \tag{5.3}$$

where $M(P_i^k(\mathbf{x}), P_j'^k(\mathbf{r}_\theta \mathbf{x}))$ is evaluated for $\mathbf{x} \in \Delta_k$ ($P_j'^k(\mathbf{r}_\theta \mathbf{x})$ is obtained from the rotated Δ_k). Literally, $Q(i,j)$ represents the highest similarity for the ith point in the first image and the jth point in the second image that can be found in any rotation angle using the five windows. Then the non-maximum suppression principle is used to select candidate matches according to the method described in Fig. 1.

Therefore the matching process is divided into two steps: given a point $\mathbf{p}$ in the first image, first search for the orientation and type of window around a candidate point $\mathbf{q}$ in the second image to find the best match between $\mathbf{p}$ and $\mathbf{q}$; then search all candidate points in the second image to obtain the best overall match for $\mathbf{p}$.

5.2. Chamfer Matching

Intensity is quite liable to change due to lighting conditions and viewpoints. A more robust matching method is to use less variable features such as edges instead of intensities. Chamfer matching[2,7] is a particular feature-based matching technique that has been used to match images consisting of edges and points. In the past, chamfer matching has been used mainly in pose estimation and recognition of planar surfaces,[2,7] where the matching problem can be reduced to the search of a few transformation parameters. We now extend this method to general image matching.

In chamfer matching, the shapes of two line drawing images (usually represented by two edge images) are used to register correspondences. To get an idea of what chamfer matching is, let us consider the problem in Fig. 6. Suppose we want to assess whether the circle or the triangle is more similar to the rectangle. Then, one criterion is the average distance from a point on the circle or the triangle to the nearest point on the rectangle. This distance is called the *chamfer distance* which is a special form of the set distance function M_4 with R being the property of edgehood. Clearly, the chamfer distance is a function of the relative scale, orientation, and position of two image patterns. In the chamfer matching method to be discussed in this subsection, scale is not optimized.

The general technique of chamfer matching is to first transform the template image into a chamfer distance map (CDM), and then use the map to compute the similarity measure M_4 for a given image to be matched. The CDM of a given feature image is an image in which the intensity of a pixel represents the distance of the pixel to the nearest feature point. For example, let $E(\mathbf{x})$ be a binary edge image, with $E(\mathbf{x}) = 1$ if $\mathbf{x}$ is an edge and $E(\mathbf{x}) = 0$ otherwise. Then the CDM $C(\mathbf{x})$ of $E(\mathbf{x})$ is evaluated as follows

$$C(\mathbf{x}) = min\{\|\mathbf{x} - \mathbf{y}\| : E(\mathbf{y}) = 1\} . \tag{5.4}$$

If each vertical or horizontal pixel distance is valued 3 and each diagonal pixel distance is valued 4, then the computation of the CDM, or the so-called 3–4 distance transform, can be obtained by sequentially executing a fast two-pass algorithm.[7]

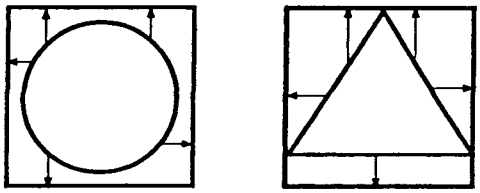


Fig. 6. The average distance from a point on the circle/triangle to the nearest point on the rectangle gives a criterion about the similarity between the circle/triangle and the rectangle.

After the CDM of one window image is obtained, the similarity function M_4 between the image and any other window image can be easily computed. To be specific, consider matching the first view to the second view. Assume the binary edge image in the first view is $E(\mathbf{x})$ and the CDM of the edge image $E'(\mathbf{x})$ in the second view is $C(\mathbf{x})$. For a given window Δ_k ($k = 1, \ldots 5$), let $E_i^k(\mathbf{x})$ be the window edge image at the ith feature point $\mathbf{u}_i$ in the first view and $E_j'^k(\mathbf{x})$ be window edge image at the jth feature point $\mathbf{u}'_j$ in the second view, i.e.

$$E_i^k(\mathbf{x}) = E(\mathbf{x} + \mathbf{u}_i), \quad E_j'^k(\mathbf{x}) = E'(\mathbf{x} + \mathbf{u}'_j), \quad \mathbf{x} \in \Delta_k . \tag{5.5}$$

Note that *feature* points here are not restricted to edges. The window CDM $C_j^k(\mathbf{x})$ for $E_j'^k(\mathbf{x})$ can be obtained as follows

$$C_j^k(\mathbf{x}) = C(\mathbf{x} + \mathbf{u}'_j), \quad \mathbf{x} \in \Delta_k . \tag{5.6}$$

Then, for a given rotation angle θ and a given window Δ_k, the chamfer distance, or the matching quality, of the two window edge images $E_i^k(\mathbf{x})$ and $E_j'^k(\mathbf{x})$ can be computed from

$$M_4(E_i^k(\mathbf{r}_\theta \mathbf{x}), E_j'^k(\mathbf{x})) = -\sqrt{\frac{1}{N_i} \sum_{\substack{E_i^k(\mathbf{r}_\theta \mathbf{x})=1 \\ \mathbf{x} \in \Delta_k}} C_j^2(\mathbf{x})} = -\sqrt{\frac{\sum\limits_{\mathbf{x} \in \Delta_k} C_j^2(\mathbf{x}) \cdot E_i^k(\mathbf{r}_\theta \mathbf{x})}{\sum\limits_{\mathbf{x} \in \Delta_k} E_i^k(\mathbf{r}_\theta \mathbf{x})}} , \tag{5.7}$$

where N_i is the number of edges in $E_i^k(\mathbf{r}_\theta \mathbf{x})$ and the negative sign is added such that a larger value represents a better match. Then the matching quality between the ith point in the first image and the jth point in the second image is represented by the optimal chamfer distance over all types of windows and orientations defined as

$$Q(i,j) = \max_{\Delta_k} \left\{ \max_{\theta} M_4\left(E_i^k(\mathbf{r}_\theta \mathbf{x}), E_j'^k(\mathbf{x})\right) \right\} . \tag{5.8}$$

Note that chamfer matching is meaningful only when the window image has a certain number of edge points. Therefore, if the window edge image at a feature point does not have enough edges, this point is discarded. The minimum number of points can be conveniently set to the number of pixels on a diameter of the window. Actually this number is not crucial to the algorithm as long as it is a reasonable fraction of the number of pixels in the window.

One of the advantages of chamfer matching is that by using the *chamfering* technique, the matching process can be much faster than waveform matching. Another advantage is that the algorithm uses the low-level tokens such as edges and corners as input and hence adapts widely to changes due to lighting. The method is especially suitable for abstract object recognition[7] such as character recognition and object classification in which objects are best defined by line drawings. The disadvantage is that more ambiguity exists in chamfer matching than in waveform matching.

5.3. Constraints for Unique, Correct Matching

A serious problem that exists for all matching methods is mismatch. We now describe the use of geometric constraint, rigidity constraint, and a newly developed disparity constraint for removing mismatches.

Geometric Constraint

The *geometric constraint* states that the projections of the parts of a rigid object have similar configurations from image to image. This similarity is measured through the 2D geometry constructed by the projected points of an object. Given a set of matched pairs $(\mathbf{p}_i, \mathbf{q}_i)$, $i = 1, 2, \ldots n$, across two images, if all pairs are correctly matched, then any line drawing that connects $\mathbf{p}_i$, $i = 1, 2, \ldots n$, in the first image must be similar to the corresponding line drawing that connects $\mathbf{q}_i$, $i = 1, 2, \ldots n$, in the second image. Possible mismatches are identified as those pairs that do not preserve the geometric relationship with other matched pairs. In general, mismatches can only accidentally satisfy the geometric constraint, but correct matches do not necessarily satisfy this constraint. Therefore, applying this constraint may also result in elimination of some correct matches.

There are many ways for applying the geometric constraint to identify mismatches (e.g. Refs. 34, 35 and 37). A simpler and more efficient method is to consider triples of points simultaneously and update the set of matched pairs recursively.

Test A (Geometric Test). This test is illustrated in Fig. 7. Each point and its two closest neighbors in the first image construct a triangle. The correspondences of the three points in the second image also construct a triangle. When the motion between the two images is rigid and small, the distortion of the triangle caused by motion is also small. Therefore, if all three pairs of correspondences are correct, the two triangles in different images must be similar. If one or more pairs are wrongly matched, then the similarity can be preserved only accidentally. The method for choosing valid matches is illustrated in Figs. 7(a) and (b). The particular waveform and chamfer matching methods are so designed that (even for regular patterns) the probability with which three nearby points in the first image wrongly match three points in the second image leading to two similar triangles is zero. However, it is indeed possible that some correct matches will be eliminated if their nearest neighbors are wrongly matched. This effect is called the *bad neighborhood effect.* To achieve higher reliability, more than 3 points can be considered together, although more correct matches could also be eliminated and more computation is needed. ■

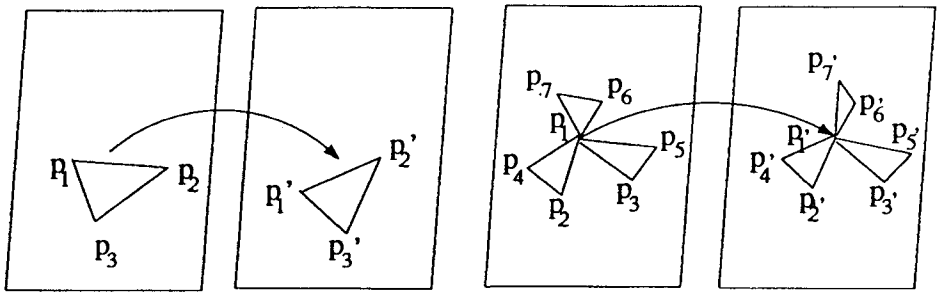


Fig. 7. (a) A feature point $\mathbf{p}_1$ is considered as having a valid match $\mathbf{p}'_1$ if the triangle constructed by $\mathbf{p}_1$ and its two closest neighbor features $\mathbf{p}_2$ and $\mathbf{p}_3$ is similar to the triangle constructed by their correspondences $\mathbf{p}'_i$, $i = 1, 2, 3$. Point $\mathbf{p}_1$ provides supports for $\mathbf{p}_2$ and $\mathbf{p}_3$. Or, (b) if $\mathbf{p}_2$ and $\mathbf{p}_3$ all have valid matches $\mathbf{p}'_2$ and $\mathbf{p}'_3$ and $\mathbf{p}_1$ is among the two closest neighbors of both $\mathbf{p}_2$ and $\mathbf{p}_3$, then $\mathbf{p}_1$ is considered as having a valid match, even if $\mathbf{p}_1$ has a bad neighbor $\mathbf{p}_7$ which has a wrong match.

The similarity of two triangles is defined as follows. Let l_i and l'_i be the side length of two corresponding triangles, $i = 1, 2, 3$. Then the side length change is defined as

$$\eta_i = \frac{|l_i - l'_i|}{max(l_i, l'_i)}, \quad i = 1, 2, 3. \tag{5.9}$$

Let

$$\eta_M = \max_i(\eta_i), \quad \eta_m = \min_i(\eta_i). \tag{5.10}$$

Then, the similarity measure is

$$S = (\eta_M - \eta_m) \cdot \eta_M. \tag{5.11}$$

If S is smaller than threshold γ, then the two triangles are considered as similar and accepted. The threshold γ is adaptive in the sense that it can be changed dynamically. If no correspondences are found for a given γ, then a larger γ is selected and the algorithm restarts. In our experiments, a fixed γ was used for all images.

Test B (Geometric Test). To overcome the bad neighborhood effect in Test A so that more matches can be found, we can use good neighbors only. In this test, a point is grouped with its nearest two points that have passed Test A to form a triangle. Therefore, the bad neighborhood effect in Test A can be eliminated. Two sets of input data are required for Test B. Let these be denoted by S_W and S_R, where S_W is the set of points that are to be checked (*working set*), and S_R is the set of points that have passed Test A, or the *rigid set.* For each point $\mathbf{p}$ in S_W, we find its two closest neighbors in S_R. We now determine if the triangles constructed by the three points and their correspondences are similar. If so, $\mathbf{p}$ is retained for further use; otherwise, $\mathbf{p}$ is discarded. ■

Test C (Geometric Test). In the above tests, the nearest neighbors are defined in terms of image plane distances. Perspective distortion could be a serious problem when the neighboring points in the image plane have large depth differences. This is called the *closest neighborhood effect.* To reduce this effect, the closest neighboring points could be required to have nearly the same disparities. That is, given a point $\mathbf{p}$ of disparity $\mathbf{d}$, the nearest neighbors of $\mathbf{p}$ are chosen as two points closest to $\mathbf{p}$ whose disparities are within $(\mathbf{d}-\mathbf{e}, \mathbf{d}+\mathbf{e})$, where $\mathbf{d}$ and $\mathbf{e}$ are vectors in the image plane. The choice of the value of $\mathbf{e}$ is discussed later in the experiment section. Test C differs from Test B only in the choice of closest neighbors. ■

Rigidity Constraint

The above geometric tests are based on 2D similarity and may hence result in error. A more profound constraint is the 3D *rigidity constraint* which requires that the object keeps the same size and shape during motion. Equity of the three-dimensional shapes and sizes of an object viewed at different times can be enforced to ensure that the matches are correct. This is a very powerful constraint. To apply this constraint, 3D motion parameters that relate objects in different images or 3D coordinates of objects in different images are needed. However, to compute depths, motion parameters are needed, which cannot be estimated without an initial set of matched points. The geometric tests serve to provide the correspondences needed for motion estimation. After six or more matches are obtained, the motion can be computed using the method, e.g. in Ref. 19, and then the 3D rigidity of the points can be examined. Points that preserve 3D rigidity may not preserve 2D geometric similarity and vice versa. But only in rare situations does it happen that points preserving 2D geometric similarity do not preserve 3D rigidity.

Test D (Rigidity Test). This test is essentially the same as Test A (or B if a rigid set has been already obtained) except that the depth data are used and the

equality of a pair of 3D triangles instead of the similarity of a pair of planar triangles is examined each time. When a pair is mismatched, a consistent depth cannot be found for the pair. In this case, the distance of the correspondence of a point p to the motion epipolar line defined by p is used to determine whether the pair is acceptable. If the distance is small and the resulting depth for the pair is positive, then the pair is retained; otherwise, discarded. ■

For two views, Test D fails for mismatches that satisfy the motion epipolar line equation and yields positive depths. This is called the *epipolar effect.* However, when a sequence of images are available, Test D is stronger than the epipolar constraint since Test D requires that the 3D structure defined by the points constructed preserve shape and size from frame to frame while the epipolar constraint applies only to pairs of images.

Disparity Constraint

For two views or stereo matching, mismatches lying on the motion epipolar line and yielding positive depths cannot be removed by the rigidity constraint. This occurs when there are occlusions or periodic patterns in the image. To remove mismatches that have escaped rigidity constraint, another constraint called *disparity constraint* is developed, which aims at identifying those matched points that can be rematched if assigned with a disparity value. The idea of disparity constraint is illustrated by Fig. 8. The disparity constraint states that *an unambiguous match is a pair of points both of which cannot be rematched with another legal disparity.* If motion is known, the space of legal disparities for a given point consists of the disparities that lead to matches on the epipolar line and yield positive depths. If motion is unknown, the space of legal disparities is unrestricted and intractable. In this case, disparities of existing matches can be used to approximate the space of legal disparities. This method works under the assumption that the disparities possessed by the already matched pairs contain those possessed by other correct, yet unknown matches. In any case, the disparity test will not worsen the results.

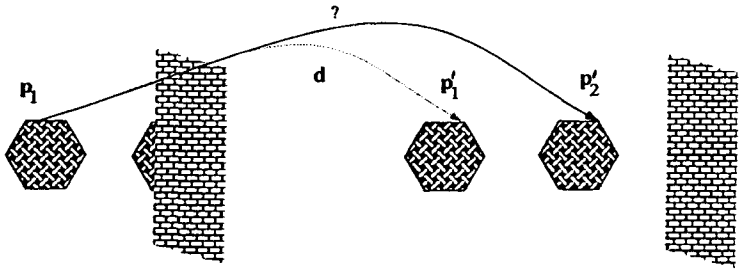


Fig. 8. Disparity constraint: Assume point $\mathbf{p}_1$ is wrongly matched to point $\mathbf{p}'_2$. When another disparity d is assigned to $\mathbf{p}_1$ such that $\mathbf{p}_1$ matches $\mathbf{p}_1 + \mathbf{d} \equiv \mathbf{p}'_1$, ambiguity arises: either $(\mathbf{p}_1, \mathbf{p}'_1)$ or $(\mathbf{p}_1, \mathbf{p}'_2)$ is the correct match. To be safe, both matches are removed.

Test E (Disparity Test). Given a set of pairs of matches $\mathbf{p}_i = (x_i, y_i)$ and $\mathbf{p}'_i = (x'_i, y'_i)$, Test E examines them for uniqueness and eliminates any nonunique ones as spurious matches. A set of disparity vectors D is formed by collecting the disparity vectors $\mathbf{d}_i = \mathbf{p}'_i - \mathbf{p}_i$ for all i (if motion is available, D is formed by the disparity vectors of all possible matches on the epipolar line that yield positive depths). When the number of matches is small (e.g. when less than 100 matches are available in a 512×512 image), D is enlarged by adding neighboring vectors (e.g. in a 3×3 window) of each disparity vector corresponding to the matches. A matched pair $\mathbf{p}_i$ and $\mathbf{p}'_i$ is accepted if for each $\mathbf{d} \in D$ such that $\|\mathbf{p}_i + \mathbf{d} - \mathbf{p}'_i\| > \epsilon$, it is true that

$$Q(\mathbf{p}_i, \mathbf{p}'_i) - Q(\mathbf{p}_i, \mathbf{p}_i + \mathbf{d}) > \delta_2 \,, \tag{5.12}$$

where ϵ is a constant denoting the radius of a *forbidden region* surrounding $\mathbf{p}'_i$. The reason for excluding disparity vectors falling within the forbidden region is that when $\mathbf{p}_i$ matches $\mathbf{p}'_i$, it is very likely that $\mathbf{p}_i$ also matches points near $\mathbf{p}'_i$. This method is shown in Fig. 9. Again, this test is done in both directions: from the first image to the second and vice versa. ■

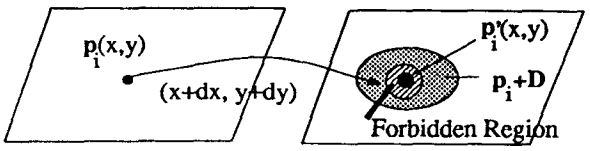


Fig. 9. Disparity test: if $\mathbf{p}_i(x, y)$ and $\mathbf{p}'_i(x', y')$ is a candidate matched pair, then for each disparity vector $\mathbf{d} = (dx, dy)$ in the disparity vector set D such that $\mathbf{p}_i + \mathbf{d}$ falls out of the forbidden region of $\mathbf{p}'_i(x', y')$, we determine if $\mathbf{p}_i$ has a good intensity-based or feature-based match with $\mathbf{p}_i + \mathbf{d}$. If so, $\mathbf{p}_i$ has more than one match and the pair $(\mathbf{p}_i, \mathbf{p}'_i)$ is removed as a spurious match; if not, pair $(\mathbf{p}_i, \mathbf{p}'_i)$ is retained.

Applying the disparity constraint is the last effort to remove possible mismatches. However, correct matches of periodic patterns could also be removed. There are only two solutions if ambiguity arises: either remove all the matches temporarily or resolve the ambiguity by examining the local geometrical situations. When insufficient information is present, the safest way is to remove all ambiguous matches. Test E can be used twice, once before the motion is solved for and once after the motion is solved for. At each time, the disparity space D has different contents as described above.

The Effect of the Constraints

After the geometric, rigidity, and disparity tests have been applied, only mismatches in rare situations described by Fig. 10 may survive. Because the method for selecting potential matches and the constraints complement each other, the probability for mismatch to occur is essentially zero. If more than three points are tested at a time in the geometric tests, mismatches may be completely eliminated (along with more good matches).

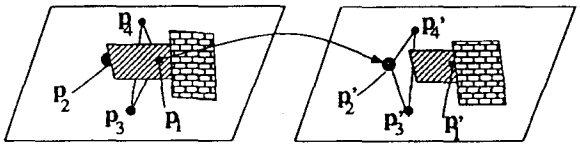


Fig. 10. Only when all of the following five conditions are satisfied, can a mismatched pair $\mathbf{p}_1$ and $\mathbf{p}'_2$ survive the geometric, rigidity, and disparity tests: (a) the correct match $\mathbf{p}'_1$ of $\mathbf{p}_1$ is undetected in the second image (b) the correct match $\mathbf{p}_2$ of $\mathbf{p}'_2$ is undetected in the first image (c) the local intensities of $\mathbf{p}_1$ matches the local intensities of $\mathbf{p}'_2$ (d) the triangle constructed by $\mathbf{p}_1$ and its two closest neighbors $\mathbf{p}_3$ and $\mathbf{p}_4$ pass the geometric and rigidity tests with respect to the triangle constructed by $\mathbf{p}'_2$, $\mathbf{p}'_3$, and $\mathbf{p}'_4$ (e) the disparity $\mathbf{p}'_1 - \mathbf{p}_1$ is not close to any disparity possessed by any other matched pair.

5.4. An Integrated Matching Algorithm

The algorithm is developed based on the five principles. However, the essence of the algorithm consists of the tests which enforce the unique response principle since other principles, as discussed in Sec. 3.2, are very easy to implement. Here we show how these principles are integrated to yield a robust matching algorithm.

Before we present the algorithm, let us consider the relationships of the tests. Test A can be used as an initial test since it does not require valid matches as references. Tests B and C need good matches as references. Test D needs knowledge of motion parameters. When Tests B and C are used, generally speaking, more point correspondences will be found. However, it is possible that some newly obtained correspondences will not satisfy the local geometrical constraints and are hence wrong matches. This situation occurs very often near the occlusion boundary, especially for views involving multiple objects of different motions. Therefore, it is necessary to apply Tests A or D again to the correspondence data obtained by Test B and/or Test C.

It should be noted that the acceptance of a point is determined by the neighbors of the point, which may be different from view to view due to perspective distortion, mismatching, or occlusion. Therefore, the result of applying a test to the points in View 1 may not be the same as when the test is applied to the points in View 2. Also, the decision about whether a point passes the test may change as the working set or the rigid set is updated.

In the stepwise algorithm below, W_1 and W_2 are the two working sets which are the input for each test, and R_1 and R_2 are the two rigid sets containing points that are the outcome of each test.

Step 1: Cross-match the detected feature points first from View 1 to View 2 and then from View 2 to View 1 using the waveform and/or chamfer matching method. If a pair of correspondences $(\mathbf{p}, \mathbf{q})$ is obtained in both matching processes, then $(\mathbf{p}, \mathbf{q})$ is retained as a unique match for further use. Assume W_1 and W_2 are the sets of points of unique matches in View 1 and View 2, respectively.

Step 2: Now apply Test A to the matches given by W_1 and W_2 to select those which satisfy the local geometric constraint. A pair $(\mathbf{p}, \mathbf{q})$ is considered a good match if either $\mathbf{p}$ passes the test in W_1 or $\mathbf{q}$ passes the test in W_2. Matched points in W_1 and W_2 are constantly moved into R_1 and R_2, respectively. Test A is repeatedly applied to points in W_1 and W_2 until no more good points can be obtained.

Step 3: Test the points in W_1 with reference to the points in R_1 and points in W_2 with reference to the points in R_2 using Test B. A pair $(\mathbf{p}, \mathbf{q})$ ($\mathbf{p} \in W_1$, $\mathbf{q} \in W_2$) will be considered a good match if $\mathbf{p}$ passes the test in W_1 or $\mathbf{q}$ passes the test in W_2. Again, good matches in W_1 and W_2 are moved into R_1 and R_2, respectively.

Step 4: Repeat Step 3 using Test C.

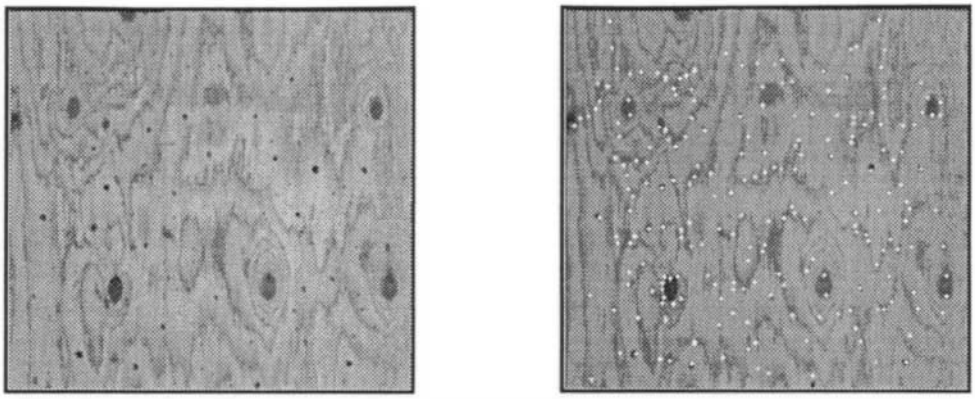
Step 5: Now apply Test D (if the images involve single rigid motion) or Test A (if the images involve multiple rigid motions) to R_1 and R_2. A pair $(\mathbf{p}, \mathbf{q})$ is removed from (R_1, R_2) if either $\mathbf{p}$ fails the test in R_1 or $\mathbf{q}$ fails in R_2.

Step 6: Apply the (Disparity) Test E to R_1 and R_2. If motion is unknown, the set of disparity vectors D is obtained from the correspondence data in R_1 and R_2. If motion is known, for each point $\mathbf{p}$ of concern, D consists of all possible disparities defined by the pixels on the epipolar line of $\mathbf{p}$ that yield positive depths. Then a pair $(\mathbf{p}, \mathbf{q})$ is removed from (R_1, R_2) if one or both of the following conditions hold: (1) for any disparity vector $\mathbf{d}$ in D such that $\mathbf{p} + \mathbf{d}$ falls out of the forbidden region of $\mathbf{q}, \mathbf{p}$ matches point $\mathbf{p} + \mathbf{d}$; (2) for any disparity vector $\mathbf{d}$ in D such that $\mathbf{q} - \mathbf{d}$ lies on the epipolar line defined by $\mathbf{q}$ and falls out of the forbidden region of $\mathbf{p}, \mathbf{q}$ matches point $\mathbf{q} - \mathbf{d}$. The remaining pairs in (R_1, R_2) comprise the results of the algorithm.

6. EXPERIMENTAL RESULTS

The feature extraction and matching methods in this paper have been implemented as part of a system for automatic understanding of shape and motion from image sequences. The methods have been applied to over a hundred image pairs and sequences and produced no mismatches of an error above three pixels for feature matching and only a few mismatches for edge matching. In this section we present a few of the experiments related to feature extraction and matching. More examples can be found in Ref. 18.

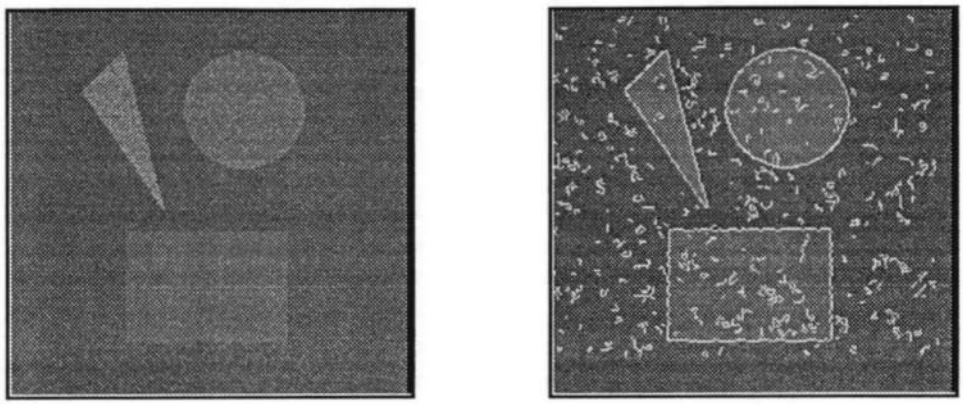
Example 1: The first example shows the performance of the spot detector. Figure 11(a) is the original image of a wooden texture. Figure 11(b) shows the result of the spot detector with $\sigma = 2.0$, from which we observe that almost all spots have been successfully detected. Therefore, the spot detector works well for well-defined spots. However, since spots are less abundant than corners and inflection points of curves, in the matching examples provided below the point features are detected by other detectors.[18]



(a) (b)

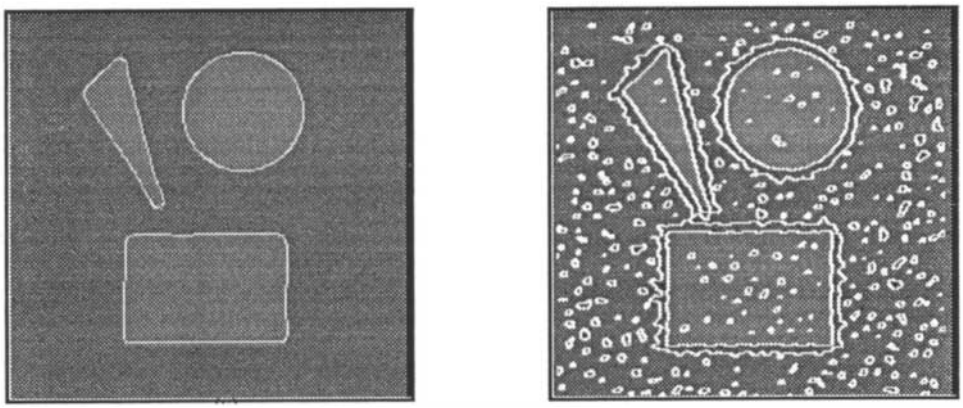
Fig. 11. Example 1. (a) A wooden texture image with artificial dots scattered in the image. (b) The result of the spot detector showing that most spots are detected.

Example 2: The second example uses synthetic images to show the performance of the edge detector. Results on real images appear in the matching examples. Figure 12(a) is a synthetic image containing a square, a disc, and a triangle lying on the background. The image is corrupted with a Gaussian noise with the variance $\sigma = 5$. The resulting signal-to-noise ratios (SNRs) for edges computed from Eq. (2.6) (with S being a step edge and n the noise) are as follows: 9 dB for the edges of the square, 12.6 dB for the edges of the circle, and 15 dB for the edges of the triangle. Figure 12(b) shows the edges detected with $\sigma = 1$. The edges are



(a) (b)

Fig. 12. Example 2. (a) A synthetic image corrupted with Gaussian noise. (b) The edges detected with $\sigma = 1.0$. The threshold is set as the strength of the edges of the square. (c) The edges detected with $\sigma = 4.0$. Edge/maximum ratio is set at 0.25. (d) Edges detected with $\nabla^2 G(x, y)$ at $\sigma = 2.0$.



(c) (d)

Fig. 12. (*Continued*).

overlapped on the original image to show the location errors. The threshold is set as the edge strength of the square edges. This shows that the noise edges cannot be removed by thresholding. When the SNR is low, the locations of the edges may be shifted one or two pixels away from the original positions, as is the case with the edges of the square (9 dB) and disc (12.6 dB). However, when the signal noise ratio increases to 15 dB, the locations of the edges are almost intact, even for sharp corners, as is the case with the edges of the triangle. Figure 12(c) shows that the noise edges can be removed by using an operator of a larger size. However, the locations of the edges are shifted for edges near the corners. The edge/maximum ratio is set to 0.25. Figure 12(d) shows the result of the zero-crossing method with the Laplacian–Gaussian operator $\nabla^2 G(x, y)$ at size $\sigma = 2.0$. There we see several problems with the zero-crossing method. First, the locations of the edges at corners are severely shifted from their original positions. Second, artificial zero-crossings are added to the edge image. Third, the quantization effect causes double edges all of which are zero-crossings.

Example 3: The third example shows the performance of the matching algorithm for images containing several objects undergoing different motions. Figure 13(a) shows the two images of an indoor scene and the detected feature points. The left image is significantly brighter since more lamps were turned on when this image was taken. Three different motions are involved in the images: the human moves nonrigidly, the chair turns around, the camera tilts and translates along the optical axis, making the background nonstationary. The feature points are detected by an improved Moravec interest operator.[18] Figure 13(b) shows the points matched using the waveform matching method after Step 1 is executed. Figure 13(c) shows the final results. The points on the background are matched very well, but no points on the human body and the chair are matched. Two reasons account for this: (1) most feature points on the chair and the human body are near the occlusion

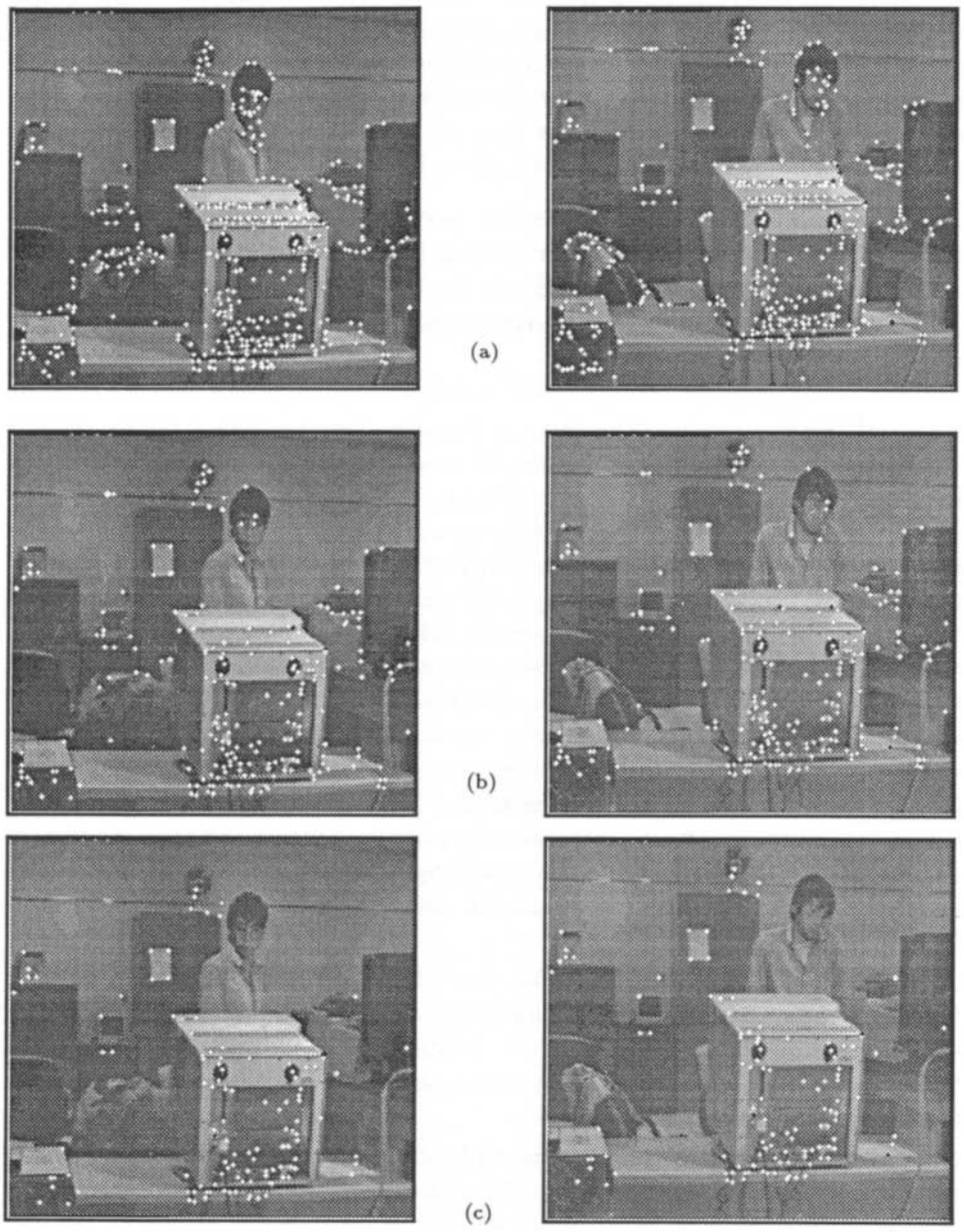


Fig. 13. Example 3. (a) Point features detected. (b) The result of intensity-based matching. (c) The final result after applying the tests.

boundary so that the neighborhood triangle rigidity is not preserved for them; (2) the feature points detected within the chair and the human body change drastically from image to image due to a large image distortion. Therefore, an insufficient number of common feature points is detected in both images. However, this example shows that even if the scene contains multiple objects, the matching algorithm still yields matches for points in the areas in which image distortion is small, no occlusion occurs, and there are enough points from within a single surface.

Example 4: The last example serves four purposes: first, it presents intermediate results of each step of the algorithm; second, it illustrates the use of 3-D rigidity, Test D or motion epipolar line constraint; third, it shows the effect of the disparity Test E; and fourth, it shows the performance when the method is extended for edge matching and surface reconstruction.

Figure 14(a) shows about 600 points detected by the improved Moravec interest operator[18] for two images. The boxes in the scene are at different depths, around which significant occlusions occur. The left image is related to the right image by a single motion involving a tilting rotation and a horizontal translation. Figure 14(b) shows the matched pairs obtained using intensity which contain some mismatches in the background and near the boundaries of the boxes. However, after applying the geometric tests, only a few very good correspondences survive, as shown in Fig. 14(c). There are two mismatches in Fig. 14(c), one in the upper left corner, the other between the two boxes. These two mismatches escape the geometric tests since they preserve the geometrical relations with their nearest neighboring feature points so well. However, they are eliminated by the disparity test, as shown in Fig. 14(d), along with some other matches. We then estimate the motion parameters between the two views (cf. Ref. 19). After the motion is estimated, the rigidity constraint is first applied to the existing matches to identify mismatches and then the motion epipolar constraint is applied to the remaining unmatched points to obtain more matches. The final results of all matched points (200 out of 600) are shown in Fig. 14(e).

As an example of chamfer matching, Fig. 14(f) shows the edge images detected by the edge detector described in this paper and the point features detected by a corner and end point detector[18]; Fig. 14(g) shows the initial matches of point features obtained by the chamfer matching algorithm and Fig. 14(h) is the final matched points of applying the tests.

Finally, we show some results obtained by extending the same methods to edge matching and surface reconstruction. The discussion of the details of these extensions is beyond the scope of this paper and the reader is hence referred to Ref. 18. Figure 14(i) shows the matched edges using the extended methods for edge matching. Almost all edges are correctly matched except a small block of edges inbetween two boxes are mismatched. The mismatches correspond to the rare situations discussed in Sec. 4 and can be corrected with additional information or constraints.[18] Figure 14(j) shows the reconstructed surface based on the matched edges, where we see that the depth discontinuities and surface shapes are recovered very accurately.

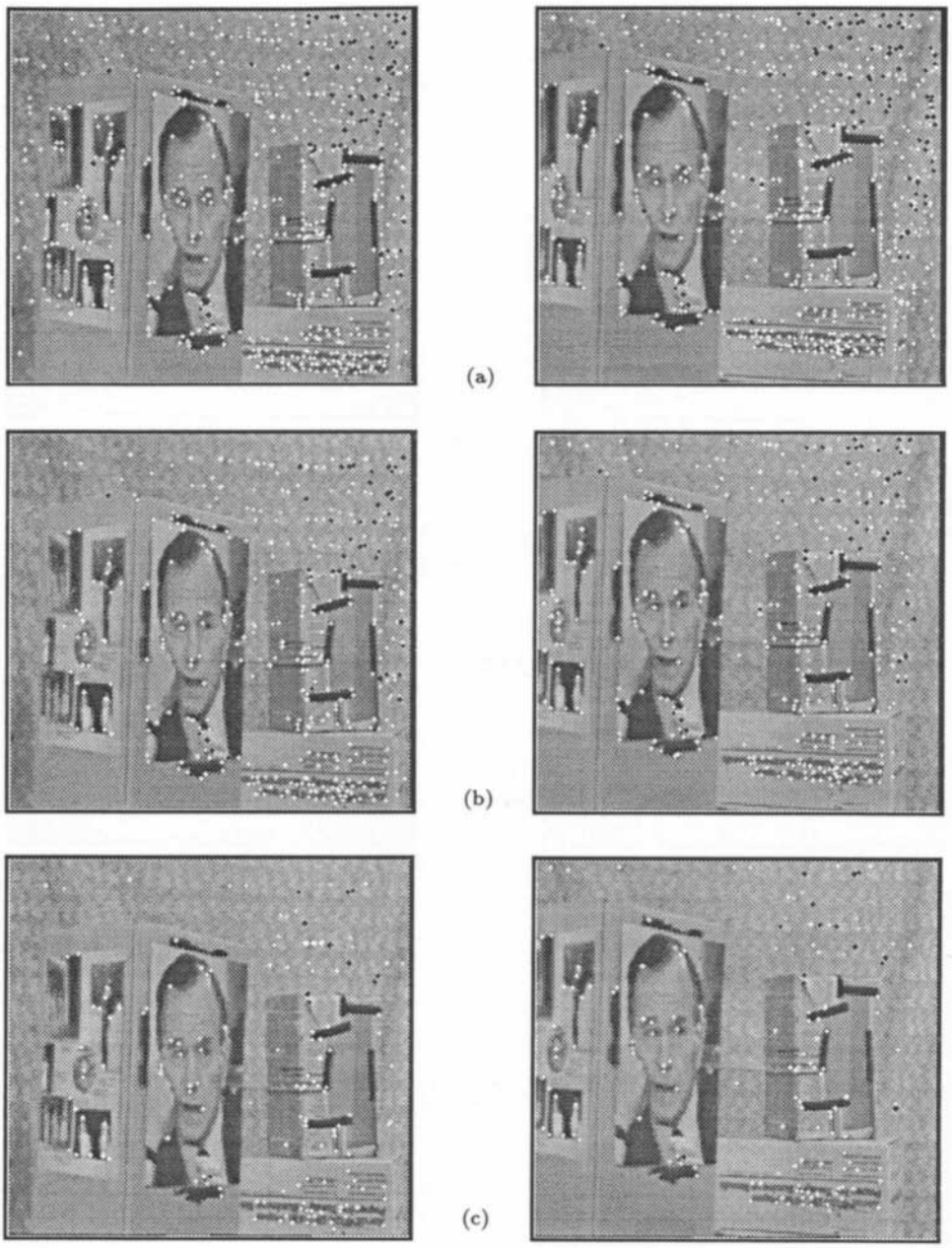


Fig. 14. Example 4. (a) Point features detected. (b) The feature points matched based on intensities after Step 1 of the matching algorithm. (c) The result after Step 5. (d) The result after Test E in Step 6. (e) The result after using the motion epipolar line constraint. (f) The edges and points used for chamfer matching. (g) The candidate matches obtained by chamfer matching. (h) The final matches of point features passing geometric tests. (i) The edges matched using the edge matching algorithm resulting from the formulation. (j) The reconstructed depth map as intensity based on the matched edges. The bright areas are closer to the viewer.

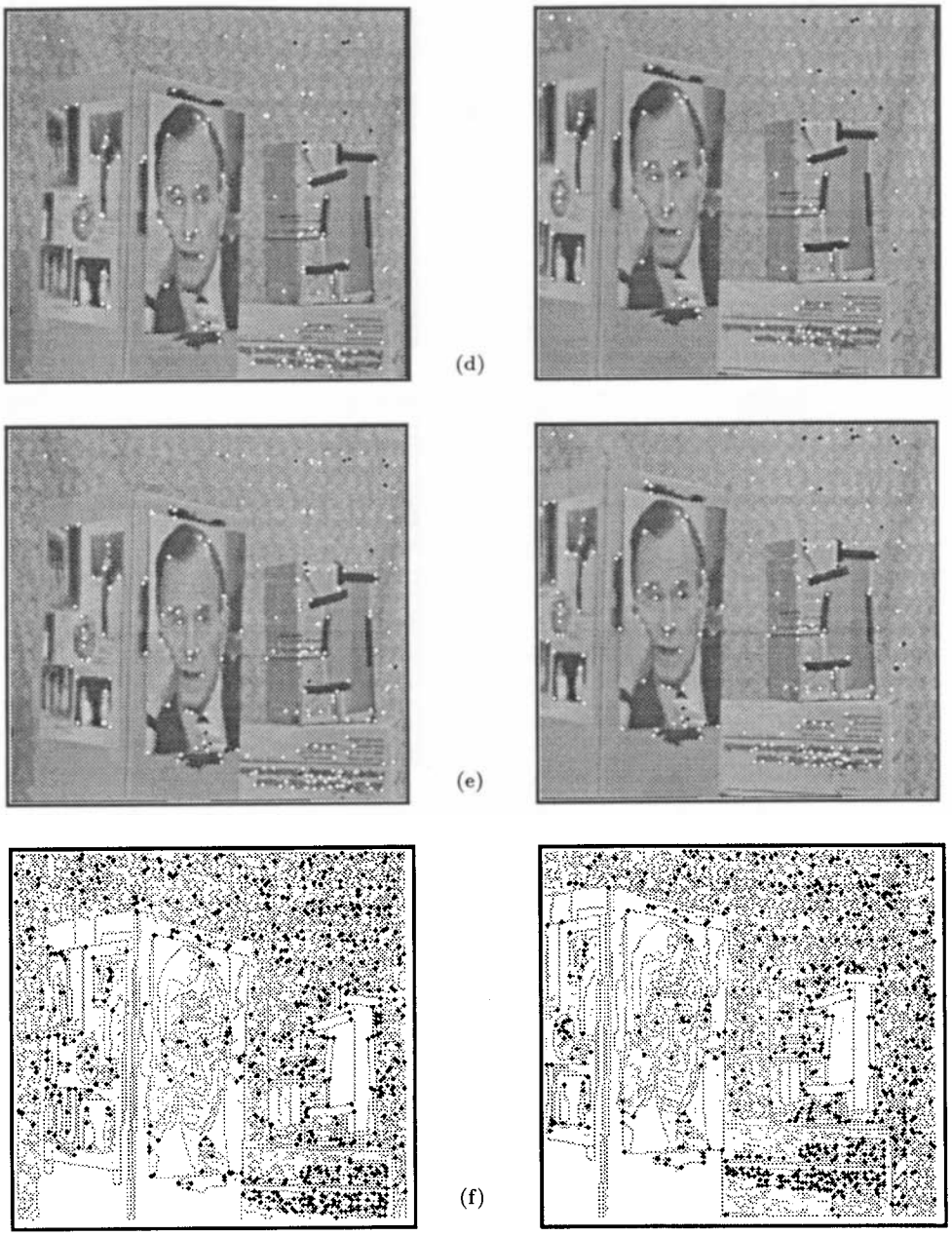


Fig. 14. (*Continued*).

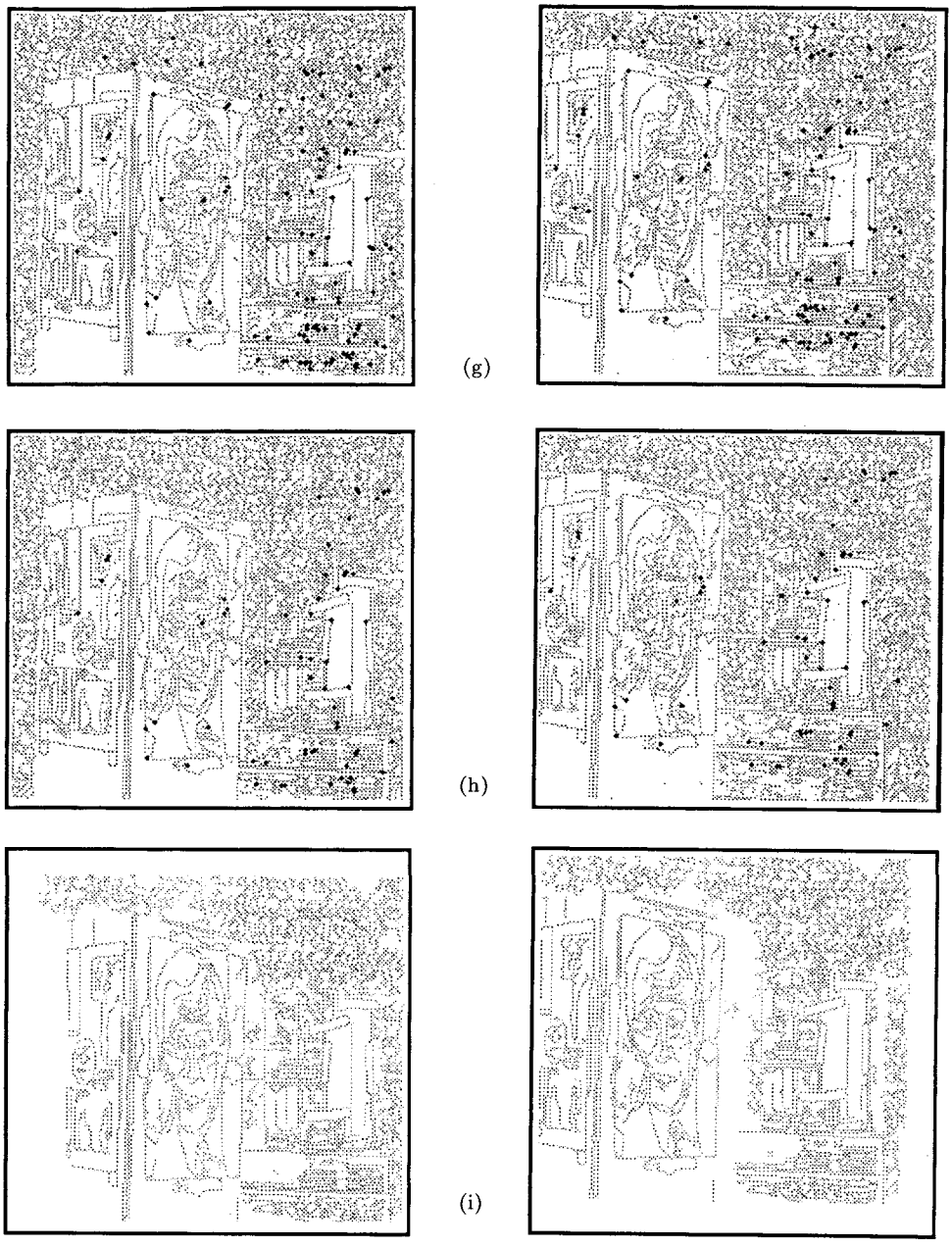
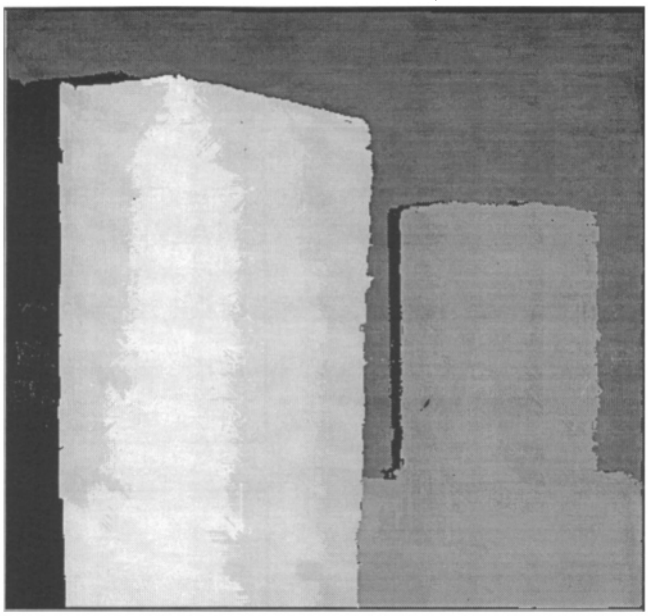


Fig. 14. (*Continued*).



(j)

Fig. 14. (*Continued*).

Typically, on a SUN Sparc 10 station, the intensity-based matching algorithm takes a few minutes to 20 minutes to match two 512×512 images containing 1,000 points, while the chamfer matching algorithm takes less than a minute. The rigidity tests take negligible time and the disparity test takes a few minutes. Therefore, most of the computation is spent on intensity-based matching. The computational complexity is proportional to the number of feature points.

Other parameters used in the algorithm are as follows. In the matching examples, M_1 is used as the similarity measure. The window size for evaluating the similarity measure is arbitrarily chosen in a range from 15×15 to 21×21 and the performance of the algorithm does not change significantly. The vector $\mathbf{e}$ in Test C is chosen as $(5, 5)$ (pixels). The thresholds δ_1 and δ_2 in the intensity-based matching part are set to 20 and 1, respectively. These choices are based on the assumption that the average intensity change per pixel between the two images is below 20 (gray levels) and that for most points, the similarity measure of the true match is 1 (gray level) better than a spurious match (otherwise, the true match is indistinguishable from the spurious one). In the geometric and rigidity tests, a neighbor is required to be at a minimum distance of 5 pixels in order to be considered as a closest neighbor. This allows up to a 20% relative error in side length distortion of a triangle when an absolute error of 1 pixel is present in one side of the triangle. The threshold γ is chosen as 0.33. In the disparity test, the forbidden region is chosen as a circle of radius 5 pixels.

7. SUMMARY

In this paper, a general formulation of feature extraction and matching as signal detection has been presented. General principles and methods have been proposed and sample algorithms resulting from these methods have been presented. Good experimental results with real images have been obtained using these methods which demonstrate the validity and usefulness of the formulation. The formulation has several advantages. First, it provides a systematic method for solving feature extraction and matching problems. Second, it unifies feature extraction and matching into a more general framework so that the two can be combined more closely in constructing an automatic vision system. Third, it builds a bridge between low level vision processing and signal detection theory. The methods of this paper apply to detection and matching of point features as well as edge matching.[18]

REFERENCES

1. G. Adiv. "Determining three-dimensional motion and structure from optical flow generated by several moving objects", *IEEE Trans. Pattern Anal. Mach. Intell.* **7**, 4 (1985) 384–401.
2. H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching", *Proc. Int. Joint Conf. on Artificial Intelligence*, vol. 2, 1977, pp. 659–663.
3. J. Ben-Arie and K. Rao, "Image expansion by non-orthogonal wavelets for optimal template matching", *Proc. 11th Int. Conf. on Pattern Recognition*, 1992, 650–654.
4. J. R. Beveridge, R. Weiss, and E. M. Riseman, "Combinatorial optimization applied to variable scale 2D model matching", *Proc. Int. Joint Conf. on Pattern Recognition*, 1990, pp. 18–22.
5. D. Blostein and N.Ahuja, "Recovering the orientation of textured surfaces in natural scenes", Tech. Rep. UILU-ENG-87–2219, Coord. Sci. Lab., University of Illinois at Urbana-Champaign, 1987.
6. D. Blostein and N. Ahuja, "Shape from texture: Integrating texture-element extraction and surface estimation", *IEEE Trans. Pattern Anal. Mach Intell.* **11**, 12 (1989) 1233–1251.
7. G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm", *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 6 (1988) 849–865.
8. J. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 6 (1986) 679–698.
9. C.-L. Cheng and J. K. Aggarwal, "A two-stage hybrid approach to the correspondence problem via forward-searching and backward-correcting", *Proc. Int. Joint Conf. on Pattern Recognition*, 1990, 173–179.
10. G. R. Cooper and C. D. McGillem, *Modern Communications and Spread Spectrum*, McGraw-Hill, New York, 1986.
11. M. S. Costa, R. M. Haralick, and L. Shapiro, "Optimal affine-invariant point matching", *Proc. Int. Joint Conf. on Pattern Recognition*, 1990, pp. 233–236.
12. J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles", *Vis. Res.* **20** (1980) 847–856.
13. J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters", *J. Opt. Soc. Amer.* **2**, 7 (1985) 1160–1169.

14. J. Daugman, "Complete discrete 2D Gabor transforms by neural networks for image analysis and compression", *IEEE Trans. Acoust. Speech Signal Process* **36**, 7 (1988) 1169–1179.
15. B. Dreher and K. Sanderson, "Receptive field analysis: Responses to moving visual contours by single lateral geniculate neurons in the cat", *J. Physiol.* **234** (1973) 95–118.
16. R. G. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, New York, 1977.
17. B. K. P. Horn and B. G. Schunk, "Determining optical flow", *Artif. Intell.* **17** (1981) 185–203.
18. X. Hu, "Perception of shape and motion", Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL, 1993.
19. X. Hu and N. Ahuja, "Estimating motion of constant acceleration from image sequences", *Proc. 11th Int. Conf. on Pattern Recognition*, Hague, The Netherlands, 1992.
20. D. Hubel and T. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex", *J. Physiol.* **160** (1962) 106–154.
21. D. Hubel and T. Wiesel, "Receptive fields and functional architecture of monkey striate cortex", *J. Physiol.* **195** (1968) 215–243.
22. R. Hummel, B. Kimia, and S. Zucker, "Gaussian blur and the heat equation: Forward and inverse solutions", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1985, pp. 668–671.
23. J. Jones and L. Palmer, "An evaluation of the two-dimensional Gabor filter model of simpler receptive fields in cat striate cortex", *J. Neurophys.* **58** (1987) 1233–1258.
24. Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Object recognition by affine invariant matching", *Proc. Int. Joint Conf. on Pattern Recognition*, 1988, pp. 335–344.
25. R. Leipnik, "The extended entropy uncertainty principle", *Inf. Control* **3** (1960) 18–25.
26. D. Marr, *Vision*, W. H. Freeman and Company, New York, 1982.
27. D. Marr and E. Hildreth, "Theory of edge detection", *Proc. R. Soc. London* **B**, 207 (1980) 187–217.
28. D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing", *Phil. Trans. R. Soc.* **B**, 275 (1979) 483–524.
29. T. Nguyen and T. Huang, "Image blurring effects due to depth discontinuities: Blurring that creates emergent image details", *Proc. 2nd European Conf. on Computer Vision*, 1992, pp. 347–362.
30. K. Prazdny, "Egomotion and relative depth map from optical flow", *Biol. Cyber.* **36** (1980) 87–102.
31. K. Rao and J. Ben-Arie, "Restoration wit equivalence to nonorthogonal image expansion for feature extraction and edge detection", *Proc. SPIE Conf. on Visual Communication and Image Processing*, 1992, pp. 187–197.
32. A. Rosenfeld and J. L. Pflatz, "Distance functions on digital pictures", *Pattern Recogn.* **1**, 1 (1968) 33–62.
33. V. Salari and I. K. Sethi, "Feature point correspondence in the presence of occlusion", *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, 1 (1990) 87–91.
34. G. Scott and H. Longue-Higgins, "An algorithm for associating the features of two patterns", *Proc. Roy. Soc. London* **B244** (1991) 21–26.
35. G. Scott and H. Longue-Higgins, "Feature grouping by "relocalization of eigenvectors of the proximity matrix", *Proc. 1st British Machine Vision Conf.*, Oxford, UK, 1990, pp. 103–108.
36. I. K. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence", *Pattern Anal. Mach. Intell.* **9**, 1 (1987) 56–73.

37. L. Shapiro and J. Brady, "Feature-based correspondence: An eigenvector approach", *J. Image Vision Comput.* **10**, 5 (1992) 283–288.
38. J. Shen and S. Castan, "An optimal linear operator for step edge detection", *CVGIP: Graph. Models Image Process.* **54**, 2 (1992) 112–133.
39. H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, Wiley, New York, pp. 1968–1971.
40. J. Weng, "A theory of image matching", *Proc. Int. Joint Conf. on Computer Vision*, 1990, pp. 200–209.
41. J. Weng, N. Ahuja, and T. S. Huang, "Two-view matching", *Proc. Int. Joint Conf. on Computer Vision*, 1988, pp. 64–73.

Received 18 January 1993; revised 31 July 1993.

Xiaoping Hu received his Ph.D. in Electrical Engineering from the University of Illinois at Urbana-Champaign in 1993. His research interests are in image processing, computer vision and video applications. He is now with the graphics group of Sun Microsystems, Inc.

Narendra Ahuja 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, USA, in 1979.

Since 1979 he has been with the Universty of Illinois at Urban-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, image synthesis, and parallel algorithms. His current research emphasizes integrated use of multiple image sources of scene information to construct three-dimensional descriptions of scenes, the use of integrated image analysis for realistic image synthesis, the use of the acquired three-dimensional information for navigation, and multiprocessor architectures for computer vision.