A Robust Method For Detecting Image Features With Application to Face Recognition and Motion Correspondence

B. S. Manjunath
Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560

Chandra Shekhar
Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2564

R. Chellappa
Dept. of Electrical Engineering
University of Maryland
College Park, MD 20742

C. von der Malsburg
Ruhr-Universität Bochum
Institut für Neuroinformatik
Federal Republic of Germany

Abstract

Feature detection is a fundamental issue in many intermediate level vision problems such as stereo, motion correspondence, image registration, etc. In this paper a new approach to feature detection is presented. It is based on a scale-interaction model of the end-inhibition property exhibited by certain cells in the visual cortex of mammals. These feature detector cells are responsive to short lines, line endings, corners and other such sharp changes in curvature. In addition, this method also provides a compact representation of feature information which is useful in shape recognition problems. Application to face recognition and motion correspondence are illustrated.

1 Introduction

We present a novel approach to feature detection and illustrate its usefulness in applications such as motion correspondence, and face recognition (see also [1] in this proceedings for application to image registration). Feature detection is an important early vision problem. Previous work on feature detection include using grey level statistics (eg: Moravec's operator [2]), and detecting edges and corners. Strong edges and corners are particularly useful in applications such as analysing aerial images of urban scenes, airport facilities etc. Algorithms based on grey level statistics are applicable to a wider variety of images such as desert scenes, surfaces of Moon and Mars, which do not contain any man made structures. Given that the nature of salient features vary from application to application, it is desirable that a feature selection algorithm be as general as possible. Features, by definition, are "perceptually interesting". In case of structured objects such features could be corners and locations with significant curvature changes. When analyzing human faces, features of interest could be the eyes, nose, mouth, etc. The generality criterion addresses the issue of whether a given feature detection algorithm can be used in a wide variety of applications. A second criterion, that of robustness, is equally important in applications such as correspondence. A feature detection algorithm can be considered robust if it identifies the same feature locations independent of rotation, translation, and minor scaling and perspective distortions. Most feature detection schemes which obtain a symbolic representation in terms of edges and corners are not quite general, where as it has been observed that general purpose feature detection algorithms such as Moravec's or its variants are not robust [3]. The method that we are going to describe below appears to be both robust and of general utility, and has been tested successfully on several wide ranging applications. A third attribute of the scheme is that it provides a simple representation mechanism as well, and this could be useful in applications such as recognition.

2 Feature Detection Model

The development of our feature detection scheme is motivated in part by the early processing stages in

0-8186-2915-0/92 $3.00 © 1992 IEEE

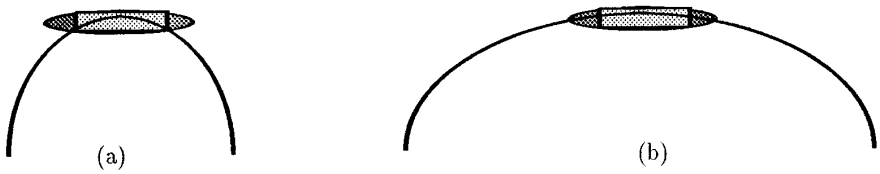

Figure 1: Illustrating the selectivity of end-inhibited cells to curvature changes. In (a) the inhibitory end zones of such a cell are not activated, and the cell in turn responds strongly to the contour. In (b) the same cell is not activated as the inhibitory end zones suppress its activity. In our model these inhibitory end zones are simulated through the interactions between simple cells at different scales.

the visual cortex. The feature detection model is an extension of our earlier work on boundary detection [4]. Cells in the visual cortex can be broadly classified into simple, complex and hypercomplex [5]. Of interest here is the end-inhibition property of the hypercomplex cells, which we try to model at a functional level. This property refers to the receptive fields of cells which respond to short lines and line endings, and whose response decrease as the line lengths are increased. In [4] we discuss the role of end-inhibition in texture boundary localization and perception of illusory contours. Here we use end-inhibition to localize perceptually significant features in the image, and to provide a simple yet powerful representation mechanism for pattern recognition.

End-inhibition is modeled using scale interactions between simple features at different spatial frequencies. The first stage in our model consists of obtaining a wavelet decomposition of the image. The basic wavelet function used is a Gabor function of the form

$$g_\lambda(x,y,\theta) = e^{-(\lambda^2 x'^2 + y'^2) + i\pi x'} \quad (1)$$
$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta$$

where λ is the spatial aspect ratio, θ is the preferred orientation. To simplify the notation, we drop the subscript λ and unless otherwise stated assume that $\lambda = 1$. For practical applications, discretization of the parameters is necessary. The discretized parameters must cover the entire frequency spectrum of interest. Let the orientation range $[0, \pi]$ be discretized into N intervals and the scale parameter α be sampled exponentially as $\alpha^j, j \in \mathbf{Z}$. This results in the wavelet family

$$(g(\alpha^j(x - x_0, y - y_0), \theta_k)), \alpha \in \mathbf{R},\ j = \{0, -1, -2, ..\} \quad (2)$$

where $\theta_k = k\pi/N$. The Gabor wavelet transform is then defined by

$$W_j(x,y,\theta) = \int f(x_1,y_1)\, g^*(\alpha^j(x_1 - x, y_1 - y), \theta)\, dx_1 dy_1 \quad (3)$$

The next stage in our model involves local scale interactions to generate end-inhibition. If $Q_{ij}(x,y,\theta)$ represents the output after interactions between features at two scales i and j ($\alpha^i < \alpha^j$) with preferred orientation θ, then

$$Q_{ij}(x,y,\theta) = g(||W_i(x,y,\theta) - \gamma W_j(x,y,\theta)||) \quad (4)$$

where $\gamma = \alpha^{-2(i-j)}$ is the normalizing factor. It is easy to visualize the inhibitory end-zones of the resulting transformation by considering (4) as a non-linear transformation of a difference-of-Gaussians along the preferred orientation.

The final step is to localize these features. Locations (x, y) in the image are identified as features if

$$Q_{ij}(x,y) = \max_{(x',y') \in N_{xy}} Q_{ij}(x',y') \quad (5)$$

where

$$Q_{ij}(x',y') = \max_\theta Q_{ij}(x',y',\theta)$$

and $Q_{ij}(x',y',\theta)$ is given by (4). N_{xy} represents a local neighborhood of (x, y) within which the search is conducted. In our experiments we have set this neighborhood to a circle with radius equal to the standard deviation of the Gaussian of the coarser of the two scales used in the interaction. Figures 1 and 2 illustrate the response of these features detectors to changes in curvature. Applications to face recognition and motion correspondence are now discussed.

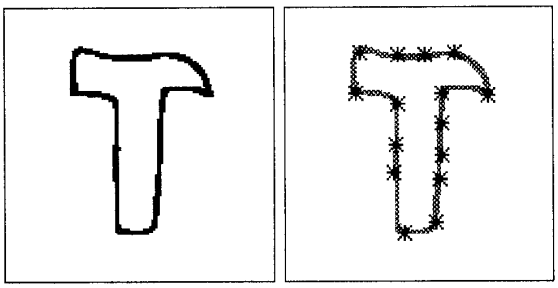

Figure 2: Salient feature locations detected by the system for the hand drawn hammer image. The particular scale-pair used in this example is $i = 0$, $j = -6$, with $\alpha = \sqrt{2}$.

3 Application to Face Recognition

Here we illustrate an useful aspect of this feature detection scheme - that of representing shape information. As we mentioned earlier, the feature locations correspond to, in some sense, locations with significant change in the local curvature. This information can be quantitatively represented as a feature vector (whose dimensions correspond to the number of discretized orientations). A graph is then constructed, with the nodes in the graph representing features, and the links representing relationship between the features. As an example, the nodes contain information about the feature values and their locations, while the links represent distances between the feature nodes. The i-th feature vector $\mathbf{q}_i$ with spatial coordinates (x,y) in our representation corresponds to

$$\mathbf{q}_i = [Q_i(x, y, \theta_1), \ldots, Q_i(x, y, \theta_N)]' \qquad (6)$$

The problem of recognition can then be formulated as an "inexact" graph matching, involving matching an input graph to one of the stored model graphs. This in turn can be formulated as an optimization problem involving minimization of an appropriate cost function. The cost function has two parts: the similarity measure between the set of matched features, and a topolgy measure which preserves the spatial relationship between matched features. We use a local deterministic search algorithm to minimize the cost function. For details we refer to [6]. To summarize, the face recognition scheme involves three steps: (a) detecting the feature locations, (b) constructing a graph representation, and (c) matching with a stored database.

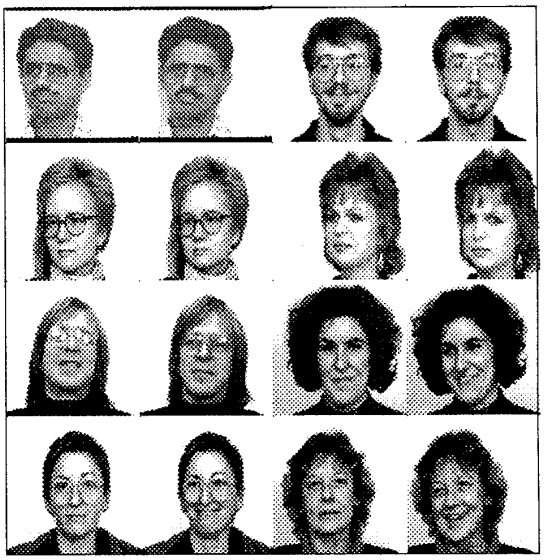

Figure 3: Examples of successful matches. The left image of each pair is the input image to the system, and the right image is the best match found.

In our current implementation, we have used four discrete orientations (N = 4, $\theta = \{0, 45, 90, 135\}$), and each feature vector needs four bytes of memory to store the information. Features are detected at one scale corresponding to the following parameters: $\alpha = \sqrt{2}, i = -2, j = -5$ in (4). The neighborhood set N_i of a feature node i consists of its five nearest neighbors. Note that this set is not necessarily symmetric. Typical number of feature points per face image is about 40, and thus we need about 150-200 bytes to store information about each face. This should be compared to the original 128×128 intensity image which occupies about 16K bytes of memory. The recognition statistics on a fairly large database of over 300 face images of 86 persons shows that the system is able to correctly identify the person 85% of the time. Figures 3 and 4 show some results of recognition. These results indicate that the system tolerates a fair amount of distortion and/or changes in facial orientation.

4 Application to Motion Correspondence

The goal here is to extract salient points from a sequence of images, and to obtain the image plane

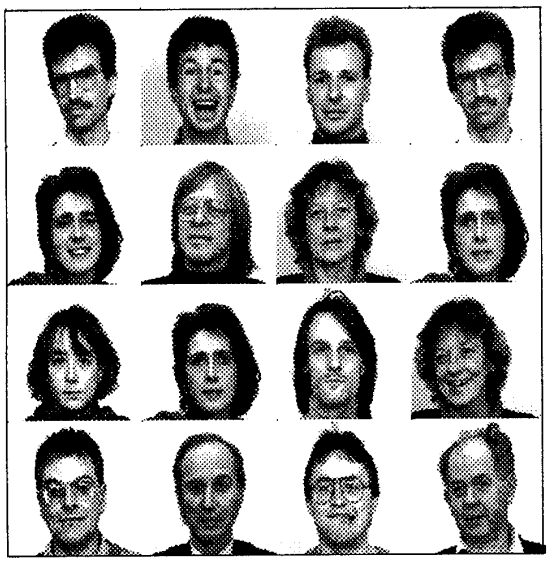

Figure 4: Examples of failures. The first image in each row is the input image, and the following three images are the top three matches found. Note that in the first two rows the correct match is among the three best matches.

trajectories of these points. This is formulated as a recursive tracking problem, with the dual objective of estimating the motion of the camera, and tracking feature points in the image sequence.

Feature points are extracted using the method described in Section 2. Feature point correspondence between two successive image frames in the sequence is posed as a labelled graph matching problem, where the feature points are treated as nodes of a labelled graph. Points within a certain minimum distance of each other are connected by edges of the graph. The problem of motion correspondence is somewhat similar to the face recognition problem in the sense that both require a correspondence between distinct features in two or more images, or between stored patterns and a test pattern. In both cases, labelled graph matching provides the required invariance to limited amounts of distortion, unlike correlation-based methods which are known to be sensitive to distortion.

The difference betweeen the two applications arises from the fact that in motion tracking, it is possible to interleave feature point matching with the recursive estimation of motion parameters. Current 3-D motion information is used to predict the positions of feature points in the incoming image, thereby reducing the search time for finding match points. Feature points are not assumed to have already been extracted in all the images in the sequence; instead, feature points extracted from the first image are "tracked" over successive images in the sequence by graph matching between consecutive image frames.

Experimental results on a real image sequence, called the UMASS Rocket sequence, are shown in Fig. 5. Details of the motion models and estimation methods used can be found in [7]. The 1st, 8th and 16th frames from this sequence are shown in Fig. 5(a), (b) and (c). Feature points extracted using scale interactions are shown in Fig. 5(d), and trajectories of selected points are shown in Fig. 5(e) and (f), superimposed on the 1st and 16th frames, respectively.

References

[1] Q. Zheng and R. Chellappa, "A computational vision approach to image registration," August 1992. in this proceedings.

[2] H. P. Moravec, "Towards automatic visual obstacle avoidance," in *Proc. 5th Int. Joint Conf. Artificial Intell.*, (Cambridge, MA), p. 584, August 1977.

[3] H. Li and S. K. Mitra, "Automatic selection of control points for image registration," Tech. Rep. CIPR 91-16, Center for Information Processing Research, University of California, Santa Barbara, September 1991.

[4] B. S. Manjunath and R. Chellappa, "A computational model for boundary detection," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Maui, Hawaii), pp. 358–363, June 1991.

[5] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, vol. 160, pp. 106–154, January 1962.

[6] B. S. Manjunath and R. Chellappa, "A feature based approach to face recognition," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Champaign, Illinois, June 1992), June 1992. (to appear).

[7] S. Chandrashekhar and R. Chellappa, "Passive navigation in a partially known environment," in *IEEE Workshop on Visual Motion*, (Princeton, NJ), pp. 2–7, October 1991.

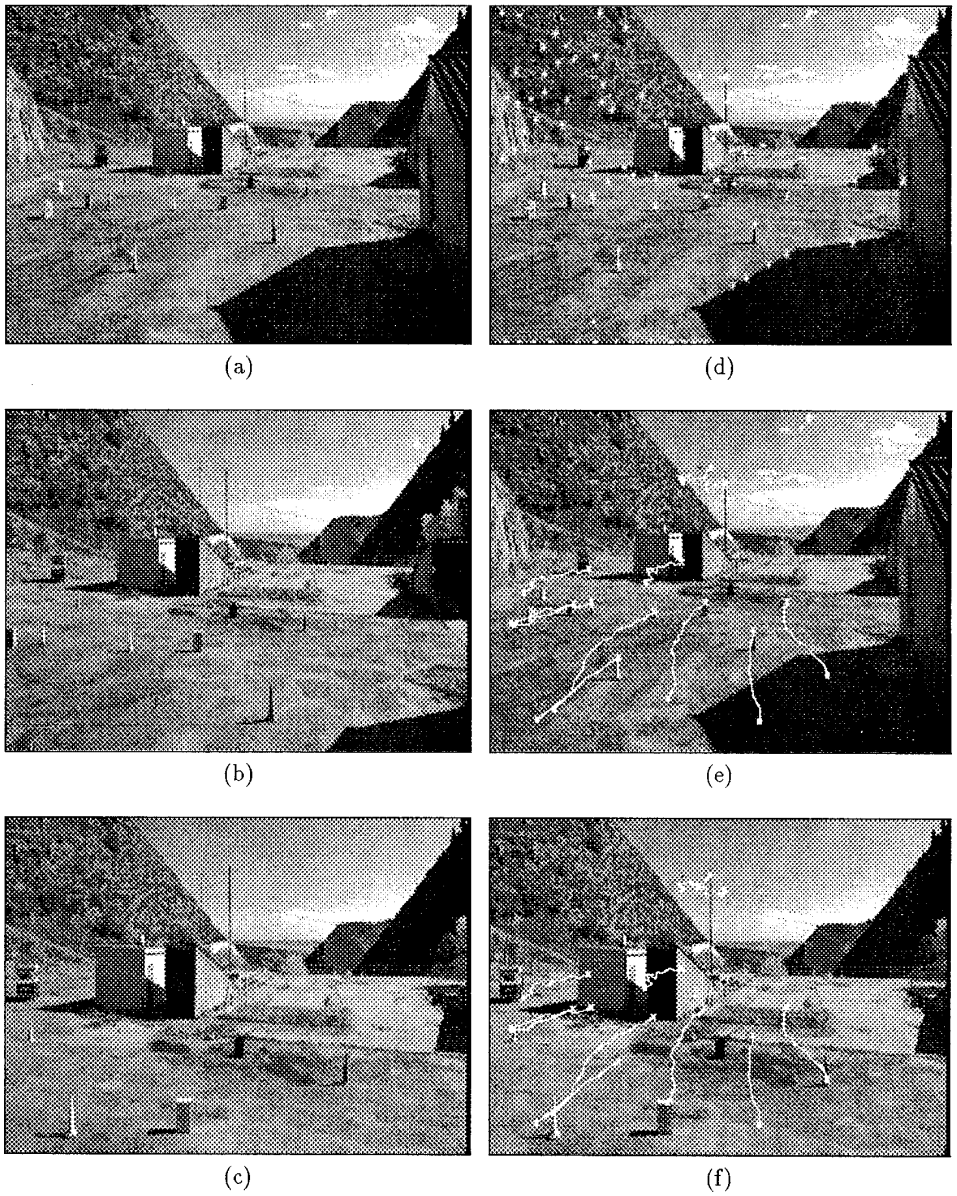

Figure 5: Tracking results for the Rocket sequence. (Courtesy: R. Dutta and R. Manmatha, University of Massachusetts at Amherst.)