Parametric Feature Detection *

Shree K. Nayar[†], Simon Baker[†], and Hiroshi Murase[‡]

[†]Department of Computer Science, Columbia University, New York, USA

[‡]NTT Basic Research Laboratory, Atsugi-shi, Kanagawa, Japan

Abstract

We propose an algorithm to automatically construct feature detectors for arbitrary parametric features. To obtain a high level of robustness we advocate the use of realistic multi-parameter feature models and incorporate optical and sensing effects. Each feature is represented as a densely sampled parametric manifold in a low dimensional subspace of a Hilbert space. During detection, the brightness distribution around each image pixel is projected into the subspace. If the projection lies sufficiently close to the feature manifold, the feature is detected and the location of the closest manifold point yields the feature parameters. The concepts of parameter reduction by normalization, dimension reduction, pattern rejection, and heuristic search are all employed to achieve the required efficiency. By applying the algorithm to appropriate parametric feature models, detectors have been constructed for five features, namely, step edge, roof edge, line, corner, and circular disc. Detailed experiments are reported on the robustness of detection and the accuracy of parameter estimation.

1 Introduction

Most applications in image processing and computational vision rely on robust detection of image features and accurate estimation of their parameters. The standard example of a parametrized feature is the step edge [Nalwa 93]. The step edge, however, is by no means the only feature of interest in image understanding. A comprehensive list would also include *lines, corners, junctions,* and *roof edges*[1] as well as numerous others. Moreover, in any given application, the term feature may take on a meaning that is specific to that application. For instance, in the inspection or recognition of a manufactured part, a subpart such as bolt may be the parametric feature of interest. In short, features may be too numerous to justify the process of deriving a new detector for each one. Is it possible to develop a single detection mechanism that is applicable to *any* parametrized feature?

[1]Given the extent to which feature detection has been explored, a survey of the work in this area is well beyond the scope of this paper. In our discussion, we only use examples of previous detectors without attempting to mention all of them. Further, we will be primarily interested in examples that use parametric feature models rather than those based on differential invariants.

This is exactly the objective of our work. We seek a general methodology for detecting parametric features. In addition to feature detection, we also wish to obtain precise estimates of the feature parameters, which if recovered with precision, can be of vital importance to higher levels of visual processing.

To obtain high performance in both detection and parameter estimation, it is essential to accurately model the features as they appear in the physical world. Hence, we choose not to make any simplifications for analytic or efficiency reasons, and instead use realistic multiparameter feature models. Further, we give careful consideration to the conversion of the continuous radiance function of the feature to its discrete image produced by a sensor. Amongst other effects, we model the blurring caused by the optical transfer function of the imaging optics, and the spatial averaging which takes places over each sensor pixel.

A parametric model of the feature, together with knowledge of the imaging system, allow us to accurately predict the pixel intensity values in a window about the imaged feature. If we regard the pixel values as real numbers, we can treat each feature as corresponding to a parameterized manifold in $\Re^N$. Here, N is the number of pixels in a fixed window surrounding the feature. In this setting, feature detection can be posed as finding the closest point on the manifold to the point in $\Re^N$ corresponding to the pixel values in a novel image window. If the closest manifold point is near enough to the novel point, we register the presence of the feature. Then, the exact location (parameters) of the closest manifold point may be used to estimate the scene parameters of the feature. This statement of the feature detection problem was first introduced by Hueckel in [Hueckel 71], and was subsequently used by Hummel [Hummel 79] amongst others.

Hueckel and Hummel both argued that to achieve high efficiency, a closed form solution must be found for (the parameters of) the closest manifold point. To make their derivations possible they used simplified feature models. Our view of feature detection is radically different. We believe that the features we wish to detect are inherently complex visual entities. Hence, we willingly forego all hope of finding closed-form solutions for the best-fit parameters. Instead, we discretize the search problem by densely sampling the feature manifold.

At first glance this may seem inefficient to the point of impracticality. However, we will demonstrate that our approach is very practical, through a combination of normalization, dimension reduc-

tion [Murase and Nayar 95], efficient heuristic search, and rejection techniques [Baker and Nayar 96a]. Even in the present unoptimized implementation, feature detection and parameter estimation take only a few seconds on a standard single-processor workstation when applied to a 512×480 image. Given the enormous strides being made in memory and multi-processor technology, it is only a matter of time before real-time performance is achieved.

2 Parametric Feature Representation

2.1 Parametric Scene Features

By a scene feature we mean a geometric and/or photometric phenomenon that produces spatial radiance variations, which, if detectable, can aid in visual perception. The continuous radiance function of the scene feature can be written as $F^c(x, y; \mathbf{q})$ where $(x, y) \in S$ are points within a feature window, S, and $\mathbf{q}$ are the parameters of the feature. For example, in the case of a step edge, $\mathbf{q}$ would include edge orientation and the brightness values on the two sides of the edge.

2.2 Image Formation and Sensing

Previous work on feature detection has implicitly assumed that artifacts induced by the imaging system are negligible and can be ignored. We make our models as precise as possible by incorporating these effects. One such effect is defocus. Another is that the finite size of the lens aperture causes the optical transfer function to be spatially bandlimited. Also, the feature itself, even before imaging, may be somewhat smoothed or rounded. The defocus factor can be approximated as a pillbox function [Born and Wolf 65], the optical transfer function by the square of the first-order Bessel function of the first kind [Born and Wolf 65], and the blurring due to imperfections in the feature by a Gaussian function [Koenderink 84]. We combine all three effects into a single blurring factor that is assumed to be a 2-D Gaussian function:

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp(-\frac{1}{2} \cdot \frac{x^2 + y^2}{\sigma^2}) \qquad (1)$$

The continuous image on the sensor plane is converted to a discrete image through two processes. First, the light flux falling within each pixel is integrated. If the pixels are rectangular in structure [Barbe 80] [Norton 82], the averaging function is:

$$a(x, y) = \frac{1}{w_x w_y}{}^2\Pi(\frac{1}{w_x}x, \frac{1}{w_y}y) \qquad (2)$$

where, w_x and w_y are the dimensions of the pixel. Next, the pixels are sampled, which we model by the rectangular grid:

$$s(x, y) = {}^2\text{III}(\frac{1}{p_x}x, \frac{1}{p_y}y) \qquad (3)$$

where, p_x and p_y are the spacings between samples. The final discrete image of a feature may then be written as:

$$F(x, y; \mathbf{q}) = \{ F^c(x, y; \mathbf{q}) * g(x, y) * a(x, y) \} \cdot s(x, y) \qquad (4)$$

where $*$ is the 2-D convolution operator. Since the above is a weighted sum of Kronecker delta functions, it can be rewritten as $F(m, n; \mathbf{q})$, where $(m, n) \in S$ are the (integer valued) pixel coordinates.

2.3 Parametric Feature Manifolds

If the total number of pixels in the window is N, each feature instance, $F(m, n; \mathbf{q})$, may be regarded as a point in the N-dimensional Hilbert space, $\Re^N$. Suppose the feature has k parameters (dim($\mathbf{q}$)=k). Then, as the parameters vary over their ranges, the point traces out a k-parameter manifold. Feature detection can then be posed as finding the closest point on the feature manifold to each novel candidate window in the image.

2.4 Parameter Reduction

For each feature instance encountered, we compute its mean, $\mu(\mathbf{q}) = \frac{1}{N} \sum_{(n,m) \in S} F(m, n; \mathbf{q})$, and its magnitude, $\nu(\mathbf{q}) = \| F(m, n; \mathbf{q}) - \mu(\mathbf{q}) \|$. We then apply the following brightness normalization:

$$F'(m, n; \mathbf{q}) = \frac{1}{\nu(\mathbf{q})} (F(m, n; \mathbf{q}) - \mu(\mathbf{q})) \qquad (5)$$

In all the features we have considered, the above normalization reduces the dimensionality of the feature manifold by two. This happens because $F'(m, n; \mathbf{q})$ is (approximately) independent of two of the (brightness) parameters in $\mathbf{q}$. Once a feature has been detected, its mean, μ, and magnitude, ν, can be used to recover the two parameters eliminated during normalization. See [Baker and Nayar 96b] for more details.

2.5 Dimension Reduction

For several reasons, such as feature symmetries and high correlation between feature instances with similar parameter values, it is possible to represent the feature manifold in a low-dimensional subspace of $\Re^N$ without significant loss of information[2]. If correlation between feature instances is the preferred measure of similarity, the Karhunen-Loéve (K-L) expansion [Fukunaga 90], yields the optimal subspace. To give an idea of the data compression possible, a step edge manifold in a 49-D Hilbert space can be represented in a 3-D subspace with only 10% loss of information.

3 Example Features

For lack of space, we now illustrate the parametric manifold representations of only 3 of the 5 features which we constructed detectors for. The results for the roof edge and the circular disc are similar and may be found in [Nayar et al. 95].

[2]This idea was first explored by Hummel [Hummel 79]. Whereas Hummel derived closed-form solutions based on simplistic feature models, our approach is to use elaborate feature models and numerical methods. This results in higher precision and greater generality. A similar approach has been adopted by Nandy et al. [Nandy et al. 96].

3.1 Step Edge

Parametric models for edges date back to the work of Hueckel [Hueckel 71]. Since then, the edge has been studied in more detail than any other visual feature (see [Davis 75][Nalwa 93]). Figures 1(a) and 1(b) show isometric and plan views of our step edge model. This model is a generalization of those used in [Hueckel 71], [Hummel 79], and [Lenz 87]. It is closest to the one used by Nalwa and Binford [Nalwa and Binford 86], but differs slightly in its treatment of smoothing effects.

The basis for the 2-D step edge model is the 1-D unit step function:

$$u(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases} \tag{6}$$

A step with lower intensity level, A, and upper intensity level, $A+B$, can be written as $A+B\cdot u(t)$. To extend to 2-D, we assume that the step edge is of constant cross section, is oriented at an angle, θ, and lies at a distance, ρ, from the origin. Then, the orthogonal distance of an arbitrary 2-D point, (x, y), from the step is given by:

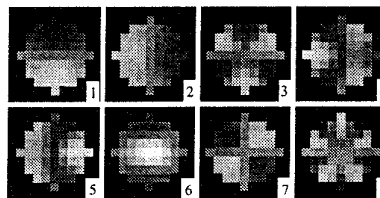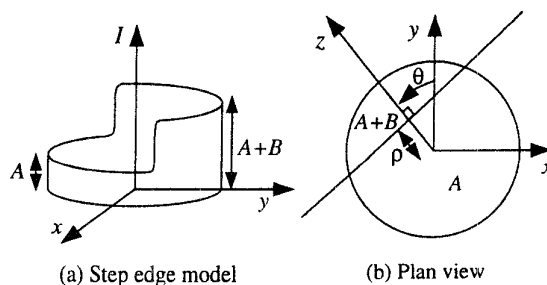$$z = y \cdot \cos\theta - x \cdot \sin\theta - \rho \tag{7}$$

Therefore, an ideal step edge of arbitrary orientation and displacement from the origin is given by the 2-D function, $A + B \cdot u(z)$. For the reasons given in section 2.2, we incorporate the Gaussian blurring function, the pixel averaging function, and the sampling function. Finally, the step edge model is: $F_{SE}(x, y; A, B, \theta, \rho, \sigma) =$

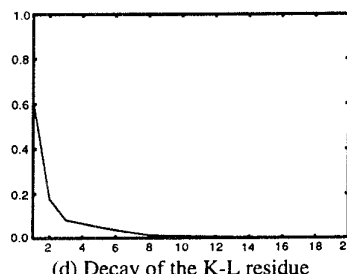$$\{ (A + B \cdot u(z)) * g(x, y; \sigma) * a(x, y) \} \cdot s(x, y) \tag{8}$$

where z is given by equation (7).

The step edge model has 5 parameters, namely, orientation, θ, localization, ρ, blurring or scaling, σ, and the brightness values, A and B. The orientation parameter, θ, is drawn from $[0^\circ, 360^\circ]$. We restrict the localization parameter, ρ, to lie in $[-1/\sqrt{2}, 1/\sqrt{2}]$, since any edge must pass closer than $1/\sqrt{2}$ pixels from the center of at least one pixel in the image. The blurring parameter, $\sigma \in [0.3, 1.5]$. As described in [Nalwa and Binford 86], substantially larger values of σ could be used, but really represent an edge at a much higher magnification. The intensity parameters, A and B, are free to take any value because of the normalization described in section 2.4. The structure of a normalized step edge is independent of A and B and is uniquely determined by the parameters, θ, ρ, and σ. Further, as described in [Baker and Nayar 96b], the values of A and B may be recovered from the mean, μ, and the magnitude, ν, calculated during normalization.
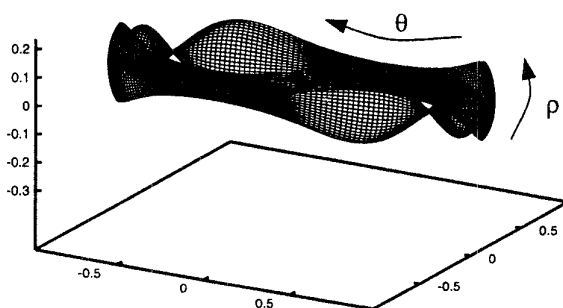
The window chosen for our edge model is a 49 pixel disc to avoid unnecessary non-linearities induced by a square window. The results of applying the Karhunen-Loéve expansion are displayed in Figures 1(c) and 1(d). In Figure 1(c) we display the 8 most important eigenvectors, ranked by their eigenvalues. The similarity between the first 4 eigenvectors and the ones derived in

(a) Step edge model　　　(b) Plan view

(c) First 8 eigenvectors

(d) Decay of the K-L residue

(e) Step edge parametric manifold

Figure 1: The step edge model includes two constant intensity regions of brightness, A and $A + B$. Its orientation and intrapixel displacement from the origin are given by the parameters, θ and ρ, respectively. The fifth parameter (not shown) is the blurring factor, σ. The K-L residue plot shows that 90% of the edge image content is preserved by the first 3 eigenvectors and 98% by the first 8 eigenvectors. The step edge manifold is parameterized by orientation and intrapixel localization for a fixed blurring value and is displayed in a 3-D subspace constructed using the first three eigenvectors.

473

[Hummel 79] is immediate. On closer inspection, however, we notice that while Hummel's eigenvectors are radially symmetric, the ones we computed are not. This is to be expected since the introduction of the parameters, ρ and σ, breaks the radial symmetry that Hummel's edge model assumes. While Hummel's eigenvectors are optimal for his continuous 3 parameter edge model, our numerically obtained results imply that they are not optimal for our discretely sampled 5 parameter model.

In Figure 1(d), the decay of the Karhunen-Loéve residue (sum of eigenvalues discarded) is plotted as a function of the number of eigenvectors. To reduce the residue to 10% we need to use 3 eigenvectors. To reduce it further to 2% we need 8 eigenvectors. Figure 1(d) illustrates a significant data compression factor of 5-15 times. As a result, feature detection and parameter estimation are made far more efficient.

The step edge manifold is displayed in Figure 1(e). Naturally, we are only able to display a projection of it into a 3-D subspace. This subspace is the one spanned by the 3 most important eigenvectors. For clarity, we only display a 2 parameter "slice" through the manifold, obtained by keeping σ constant and varying θ and ρ. As mentioned earlier, the first 3 eigenvectors capture more than 90% of the information. This is reflected in Figure 1(e), where most points on the manifold are seen to lie at unit distance from the origin. The four apparent "singularities" of the manifold are simply artifacts of the projection of the manifold into the 3-D subspace. If we were able to visualize a higher dimensional projection, these would disappear.
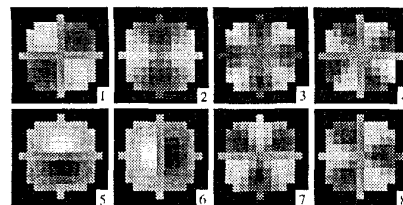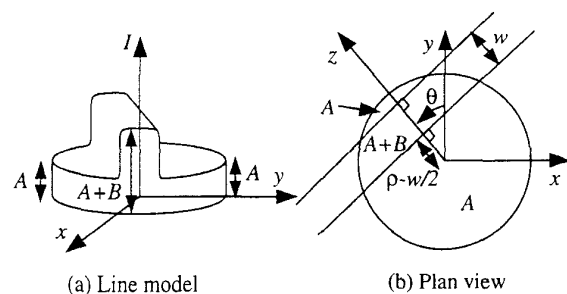
3.2 Line

The line consists of a pair of parallel step edges separated by a short distance, namely, the width w of the line [Hueckel 73] [Lenz 87]. The line is illustrated in Figure 2(a). In our definition, we assume that the intensity steps are both of the same magnitude. It is possible to generalize this model to lines with arbitrary brightness on either side with the addition of one extra parameter [Hueckel 73]. Our symmetric line model has 6 parameters and is given by: $F_L^c(x, y; A, B, \theta, \rho, w, \sigma) =$

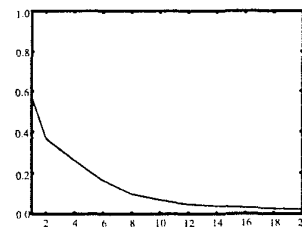$$A + B \cdot u(z + w/2) - B \cdot u(z - w/2) \qquad (9)$$

The discrete line model is then given by equation (4).

The ranges of the parameters, ρ and σ, are $\rho \in [-1/\sqrt{2}, 1/\sqrt{2}]$, and $\sigma \in [0.4, 1.0]$. Given the brightness symmetry in our line model, the orientation range can be halved to $\theta \in [0^{\circ}, 180^{\circ}]$. We restrict the line width to $w \in [1.0, 3.5]$. Just as for the step edge, the brightness parameters, A and B, are free and can be eliminated by the normalization procedure in section 2.4. Again, during detection, A and B can be recovered from the normalization coefficients, μ and ν.
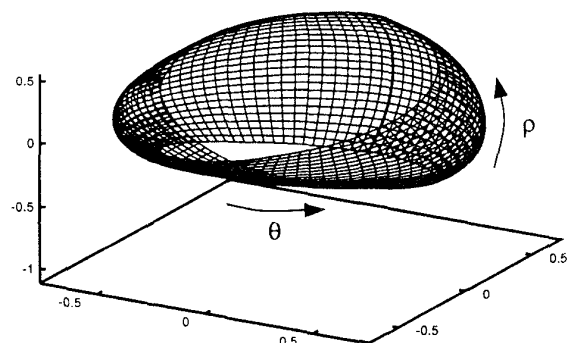
The window chosen for our line model is an 81 pixel disc. The result of applying the Karhunen-Loéve expansion is somewhat different from that for the step edge. Most significant is the lower rate of decay in the residue,

(a) Line model   (b) Plan view

(c) First 8 eigenvectors

(d) Decay of the K-L residue

(e) Line parametric manifold

Figure 2: The line is of width, w, brightness, $A + B$, and has regions of brightness, A, on either side. In addition, it has the orientation parameter, θ, the localization parameter, ρ, and the blur parameter, σ. 8 eigenvectors are need to capture 90% of the feature content and 22 eigenvectors for 98%. By this measure, the line is a considerably more complex feature than an edge. The symmetric line manifold has the structure of a Möbius band.

474

as seen in Figure 2(d). To reduce the residue to 10% we require 8 eigenvectors, and to reduce it to 2% we need 22. By this measure the line is a considerably more complex feature than an edge. However, the data compression factor is still large, and in the range of 3-5 for the 81 pixel disc. The symmetric line manifold in Figure 2(e) has the structure of a Möbius band. This results from the following symmetry in the line model:

$$F_L(; A, B, \theta + 180^o, \rho, w, \sigma) = F_L(; A, B, \theta, -\rho, w, \sigma) \quad (10)$$

3.3 Corner

The corner is a common and hence important image feature [Nobel 88]. In our corner model, shown in Figure 3(a), θ_1 is the angle of one of the edges which comprise the corner, and θ_2 is the angle subtended by the corner itself, as illustrated in Figure 3(b). Mathematically, the corner can be expressed as the product of two unit step functions: $F^c_C(x, y; A, B, \theta_1, \theta_2, \sigma) =$

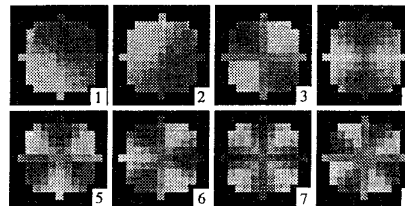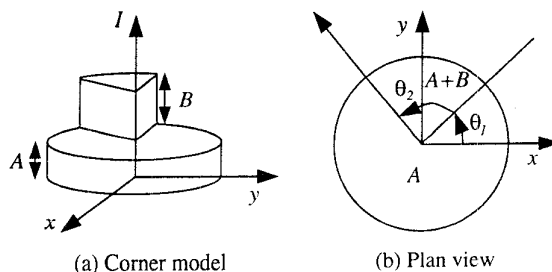$$A + B \cdot u(z(\theta_1)) \cdot u(z(180^o + \theta_1 + \theta_2)) \quad (11)$$

where, $z(\theta) = y \cdot \cos\theta - x \cdot \sin\theta$. Then, F_C is given by equation (4). The parameter ranges used are, $\theta_1 \in [0^o, 360^o]$, $\theta_2 \in [30^o, 120^o]$, and $\sigma \in [0.4, 1.0]$. As above, brightness normalization eliminates the parameters, A and B. The window chosen for the corner model is an 81 pixel disc. The decay of the K-L residue is displayed in Figure 3(d) and is similar to that of the line. Here, 7 eigenvectors reduce the residue to 10%, and 15 eigenvectors are needed to reduce it to 2%. The corner manifold has a rather interesting shape, as can be seen in Figure 3(e).
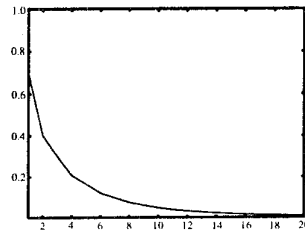
4 Feature Detection

Given a point in $\Re^N$ corresponding to the pixel intensity values in a novel feature window, feature detection requires finding the closest point on the parametric manifold. If the distance between the novel point and the closest manifold point is sufficiently small, we declare the presence of the feature. The parameters of the closest manifold point are then used as estimates of the scene feature's parameters. If the distance between the novel point and the manifold is too large, we assert the absence of the feature.

We approximate the closest manifold point by first densely sampling the manifold, and then performing a search for the closest sample point. So long as we sample densely enough, this yields a sufficiently good estimate of the closest manifold point. The search technique used is a heuristic coarse-to-fine search which takes advantage of the relatively smooth manifolds. The details of the sampling and search procedures can be found in [Nayar et al. 95].
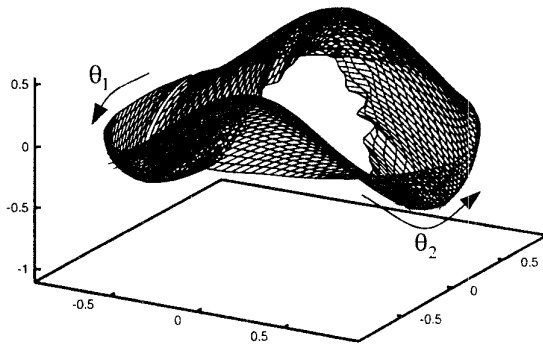
As an example of the search complexity for the step edge model, if we sample θ every 1.6^o, ρ every 0.088, and σ every 0.14, we end up with 46,368 sample points. If the manifold is sampled in a 10-D space, the complete time

(a) Corner model   (b) Plan view

(c) First 8 eigenvectors

(d) Decay of the K-L residue

(e) Corner parametric manifold

Figure 3: The corner is described by the brightness values, A and $A + B$, inside and outside the corner, the angles, θ_1 and θ_2, made by its edges, and the blur parameter, σ. Here, 7 eigenvectors are needed to preserve 90% of the information and 15 eigenvectors for 98%. The corner manifold is shown for a particular value of σ.
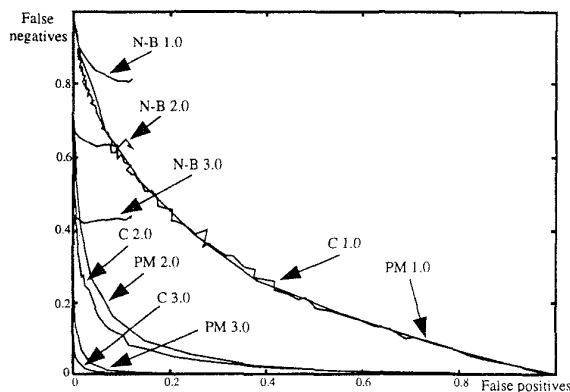
475

Figure 4: A comparison of edge detection rates. The Canny (C), Nalwa-Binford (N-B), and parametric manifold (PM) detectors are compared for S.N.R. = 1.0, 2.0, and 3.0. We plot false positives against false negatives. For each detector and S.N.R., the result is a curve parameterized by the threshold inherent in that detector. The closer a curve lies to the origin, the better the performance. We see that the Canny detector and the parametric manifold technique perform similarly, with the Canny detector doing marginally better for low levels of noise. The results for the Nalwa-Binford detector (which are consistent with the results presented in [Nalwa and Binford 86]) are completely different.

to perform normalization, projection and search takes around 1ms on a DEC Alpha 3600. So, for a 512 × 480 image complete processing would take around 4 minutes. However, by applying rejection techniques such as [Baker and Nayar 96a] the overall time was reduced to under 30secs.

5 Experimental Results

5.1 Feature Detection Rates

We statistically compare our step edge detector with the Canny [Canny 86] and Nalwa-Binford [Nalwa and Binford 86] detectors, following the approach in [Nalwa and Binford 86]. Since we took great care modeling both the features and the imaging system, we used our step edge model to generate ideal step edges for the comparison. For fairness, however, we changed the details slightly. Both the Canny and Nalwa-Binford detectors assume a constant blur/scale, so we fixed the value of σ in the step edge model to be 0.6 pixels. Secondly, the Nalwa-Binford detector is based on a square 5 × 5 window, as is Canny in the implementation that we used. Hence, we changed the window of our detector to be a square window containing 25 pixels, rather than the 49 pixel disc window used earlier. We generate "not edges" exactly as in [Nalwa and Binford 86], by taking a constant intensity window, and adding white zero-mean Guassian noise.

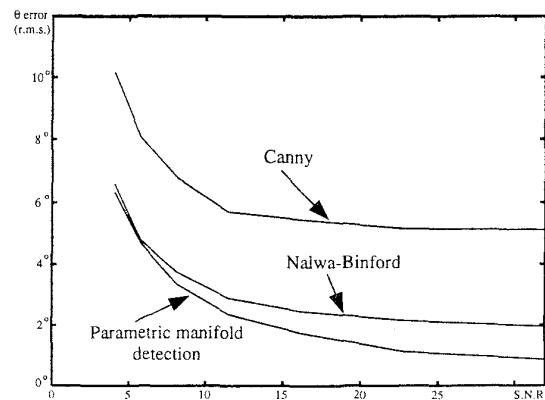In Figure 4 we compare the detection performance of the three edge detectors. For each pair of S.N.R. and

Figure 5: A comparison of edge detector orientation estimation accuracy. We took a synthesized step edge, added noise to it, and then applied the edge detectors. We plot the r.m.s. error of the orientation estimate against the S.N.R. We can see that for high levels of noise (low S.N.R.) the accuracy is inherently limited by the noise. As the noise level decreases, the parametric manifold approach outperforms both the Nalwa-Binford and Canny detectors.

detector, we plot a curve of false positives against false negatives obtained by varying the threshold inherent in each detection algorithm. The Canny operator thresholds on the gradient magnitude, the Nalwa-Binford detector thresholds on the estimated step size, and our approach thresholds on the distance from the parametric manifold. The rate of false positives was estimated by applying each detector to a constant intensity window with noise added. The rate of false negatives is obtained by applying the detectors to noisy ideal step edges.

The closer a curve lies to the origin in Figure 4, the better the performance. Hence, we can see that both the Canny detector and our detector do increasingly well as the S.N.R. increases. The results for the Nalwa-Binford detector are consistent[3] with those described in [Nalwa and Binford 86]. Applied to real images, the Nalwa-Binford detector does not perform as poorly as Figure 4 might indicate. The poor Nalwa-Binford results are probably due to thresholding on the step-size and may well be completely different if we fix the step-size threshold, and vary the tanh-fit threshold.

5.2 Parameter Estimation Accuracy

Again following [Nalwa and Binford 86], we analyze parameter estimation accuracy by randomly generating a set of feature parameters, synthesizing a feature with these parameters, adding noise, applying the detector, and then measuring the accuracy of the estimated parameters. In Figure 5, we compare the performance of our step edge detector with that of the Canny detector [Canny 86] and the Nalwa-Binford

---

[3]We did use step 2)' of the Nalwa-Binford algorithm, however the inclusion of this step does not radically alter the performance. See [Nalwa and Binford 86] for more details.

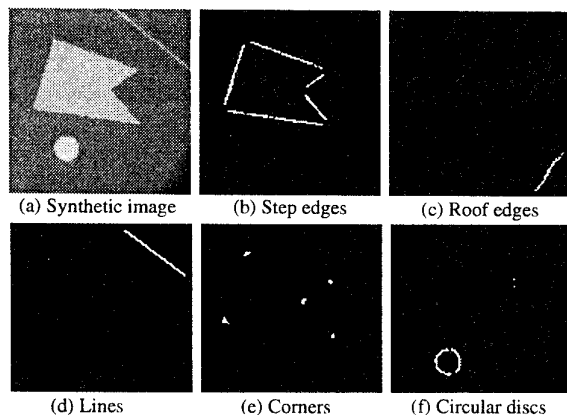| (a) Synthetic image | (b) Step edges | (c) Roof edges |
| (d) Lines | (e) Corners | (f) Circular discs |

Figure 6: The application of our 5 feature detectors to a synthetic image with noise. It is possible to completely detect and discriminate all 5 example features in the same image using the same technique.

[Nalwa and Binford 86] detector. In the figure, we plot the R.M.S. error in the estimate of the orientation, θ, against the S.N.R. We see that for low S.N.R. the performance of all detectors is limited by the noise. For lower noise levels, our detector out performs both of the other detectors.

5.3 Application to Images

In Figures 6(b)–(f) we display the results of applying the 5 example feature detectors to the synthetic image in Figure 6(a). The synthetic image is of size 128×128 pixels and contains a pentagonal region (intensity 175), a circular disc (radius 8.5 pixels, intensity 206), a line (width 2.3 pixels, intensity 153), and a roof edge (slope 4 intensity levels per pixel). The background intensity is 110. The image was first blurred with Guassian smoothing ($\sigma = 0.6$ pixels) and then white zero-mean Gaussian ($\sigma = 4.0$ grey-levels) noise was added. At pixels where two feature detectors both register the presence of a feature, we choose the one with the closer distance to the manifold. To our knowledge, this is first time these 5 different features have been detected and discriminated in the same image. Further, the proposed technique can easily be generalized to other user-defined parametric features [Nayar et al. 95].

Acknowledgements

References

[Baker and Nayar 96a] S. Baker and S.K. Nayar, "Pattern Rejection," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, 1996.

[Baker and Nayar 96b] S. Baker and S.K. Nayar, "The Design and Implementation of Parametric Feature Detectors," *Columbia University Technical Report*, CUCS-008-96.

[Barbe 80] D.F. Barbe, editor, *Charge-Coupled Devices*, Springer-Verlag, 1980.

[Born and Wolf 65] M. Born and E. Wolf, *Principles of Optics*, Permagon Press, 1965.

[Bracewell 78] R.N. Bracewell, *The Fourier Transform and Its Applications*, Second Edition, McGraw-Hill Book Co., 1978.

[Canny 86] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[Davis 75] L.S. Davis, "A Survey of Edge Detection Techniques," *Computer Graphics and Image Processing,* 4:349–376, 1975.

[Fukunaga 90] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.

[Hueckel 71] M.H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," *Journal of the Association for Computing Machinery*, 18:113–125, 1971.

[Hueckel 73] M.H. Hueckel, "A Local Visual Operator Which Recognizes Edges and Lines," *Journal of the Association for Computing Machinery*, 20:634–647, 1973.

[Hummel 79] R.A. Hummel, "Feature Detection Using Basis Functions," *Computer Graphics and Image Processing*, 9:40–55, 1979.

[Koenderink 84] J.J. Koenderink, "The Structure of Images," *Biological Cybernetics,* 50:363–370, 1984.

[Lenz 87] R. Lenz, "Optimal Filters for the Detection of Linear Patterns in 2-D and Higher Dimensional Images," *Pattern Recognition*, 20:163–172, 1987.

[Murase and Nayar 95] H. Murase and S.K. Nayar, Visual Learning and Recognition of 3-D Objects from Appearance," *International Journal of Computer Vision*, 14:5–24, 1995.

[Nalwa 93] V.S. Nalwa, *A Guided Tour of Computer Vision*, Addison-Wesley, 1993.

[Nalwa and Binford 86] V.S. Nalwa and T.O. Binford, "On detecting edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:699–714, 1986.

[Nandy et al. 96] D. Nandy, Z. Wang, J. Ben-Arie, K.R. Rao, N. Jojic, "A Generalized Feature Extractor using Expansion Matching and the Karhunen-Loeve Transform," In *Proceedings of the ARPA Image Understanding Workshop*, pages 969–972, Palm Springs, February 1996.

[Nayar et al. 95] S.K. Nayar, S. Baker, and H. Murase, "Parametric Feature Detection," *Columbia University Technical Report*, CUCS-028-95, 1995.

[Nobel 88] J.A. Nobel, "Finding corners," *Image and Vision Computing*, 6:121–127, 1988.

[Norton 82] H.N. Norton, *Sensor and Analyzer Handbook*, Prentice-Hall, 1982.