

Parametric Feature Detection*

Simon Baker[†], Shree K. Nayar[†], 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. In the algorithm, each feature is represented as a densely sampled parametric manifold in a low dimensional subspace of \mathbb{R}^N . Detection is performed by projecting the brightness distribution around each image pixel into the subspace. If the projection lies sufficiently close to the feature manifold, the feature is detected and the location of the closest point on the manifold is used to estimate the feature parameters. By applying the algorithm to appropriate feature models, detectors have been constructed for five parametric features, namely, step edge, roof edge, line, corner, and circular disc.

1 Introduction

Many applications in computational vision rely upon robust detection of image features and accurate estimation of their parameters. Although the standard example of such a feature is the *step edge*, it is by no means the only feature of interest. A comprehensive list would also include *lines*, *corners*, *junctions*, and *roof edges*¹ as well as numerous others. In short, features may be too numerous to justify the process of deriving a new detector for each one. Our aim in this paper is to develop a single detection mechanism that can be applied to any parametric feature. Moreover, we wish to obtain precise estimates of feature parameters, which if recovered with precision can be of vital importance to higher levels of visual processing.

To obtain high performance in both feature 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 simplifying assumptions for analytic or efficiency reasons, and instead use realistic multi-parameter feature models. Further, we give careful consideration to the conversion of the continuous radiance function of the feature in the world to its discrete image.

Given a parametric model of a feature and a model of the imaging system, we can accurately predict the pixel brightness values in a window about an imaged feature. If we regard the pixel brightness values as real numbers, we can treat each feature as corresponding to a parametric manifold in \mathbb{R}^N , where N is the number of pixels in the window surrounding the feature. Feature detection is then posed as finding the closest point on the manifold to the point in \mathbb{R}^N corresponding to the pixel brightness values in a novel image window. If the closest manifold point is near enough to the novel point, we detect the feature and the exact location (parameters) of the closest manifold point may be used as estimates of the parameters of the feature. This statement of the feature detection problem was first introduced by Hueckel [1971] and was subsequently used by Hummel [1979] amongst others.

Hueckel and Hummel both argued that, in order 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 argue that the features we wish to detect are inherently complex visual entities and so give up 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, finding the closest sample point may seem inefficient to the point of impracticality. However, we will demonstrate that our approach is very practical through a combination of normalization, dimension reduction [Nayar *et al.*, 1996], efficient heuristic search [Baker *et al.*,

*This research was supported in parts by ARPA Contract DACA-76-92-C-007, DOD/ONR MURI Grant N00014-95-1-0601, an NSF National Young Investigator Award, and the NTT Basic Research Laboratory.

¹Given 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 primarily be interested in examples that use parametric feature models rather than those based upon differential invariants.

1998], and rejection techniques [Baker and Nayar, 1996b]. 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.

2 Parametric Feature Representation

2.1 Parametric Scene Features

By a scene feature we mean a geometric or photometric phenomenon that produces spatial radiance variations which 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.

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, 1965], the optical transfer function by the square of the first-order Bessel function of the first kind [Born and Wolf, 1965], and the blurring due to imperfections in the feature by a Gaussian function [Koenderink, 1984]. We combine all three effects into a single blurring factor that is assumed to be a 2-D Gaussian function:

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

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, 1980] [Norton, 1982], the averaging function is:

$$a(x, y) = \frac{1}{w_x w_y} \Pi\left(\frac{1}{w_x}x, \frac{1}{w_y}y\right) \quad (2)$$

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

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

where p_x and p_y are the spacings between samples. The final discrete image of a feature may then be written as: $F(x, y; \mathbf{q}) =$

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

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

2.3 Parametric Feature Manifolds

If the number of pixels (m, n) in the window S is N , each feature instance $F(m, n; \mathbf{q})$ may be regarded as a point in \mathfrak{R}^N . Suppose the feature has k parameters: $\dim(\mathbf{q})=k$. Then, as the parameters vary over their ranges, $F(m, n; \mathbf{q})$ traces out a k -parameter manifold. Feature detection is then posed as finding the closest point on the feature manifold to the point in \mathfrak{R}^N corresponding to each window in the image. If the manifold is near enough, we detect the feature and the location (parameters) of the closest manifold point provides an estimate of the feature parameters.

2.4 Parameter Normalization

For each feature instance $F(m, n; \mathbf{q})$ encountered, we compute its mean pixel value $\mu(\mathbf{q}) = \frac{1}{N} \sum_{(n,m) \in S} F(m, n; \mathbf{q})$, and its pixel variance $\nu(\mathbf{q}) = \| F(m, n; \mathbf{q}) - \mu(\mathbf{q}) \|^2$. We then apply the following brightness normalization:

$$\overline{F}(m, n; \mathbf{q}) = \frac{1}{\nu(\mathbf{q})} [F(m, n; \mathbf{q}) - \mu(\mathbf{q})] \quad (5)$$

For all of the features we have considered, the above normalization reduces the dimensionality of the feature manifold by two. This happens because $\overline{F}(m, n; \mathbf{q})$ is (approximately) independent of two of the parameters in \mathbf{q} . Once a feature has been detected, μ and ν can be used to recover the two normalized parameters [Baker *et al.*, 1998].

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 \mathfrak{R}^N without significant loss of information². If correlation between fea-

²This idea was first explored in [Hummel, 1979]. Whereas Hummel derived closed-form solutions based upon 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 in [Nandy *et al.*, 1996].

ture instances is the preferred measure of similarity, the Karhunen-Loéve (K-L) expansion [Fukunaga, 1990], yields the optimal subspace.

3 Example Features

For lack of space, we now illustrate the parametric manifold representations for only 1 of the 5 features which we constructed detectors for. The results for the other features are similar and may be found in [Baker *et al.*, 1998].

3.1 Step Edge

Figures 1(a) and 1(b) show isometric and plan views of our step edge model. It is a generalization of the models used in [Hueckel, 1971], [Hummel, 1979], and [Lenz, 1987]. It is particularly similar to the model of [Nalwa and Binford, 1986], differing only slightly in its treatment of smoothing effects.

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

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

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 angle θ to the x -axis, and lies at distance ρ from the origin. Then, the perpendicular distance of an arbitrary 2-D point (x, y) from the step is given by:

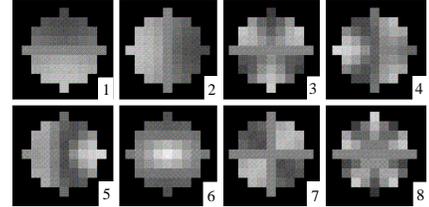
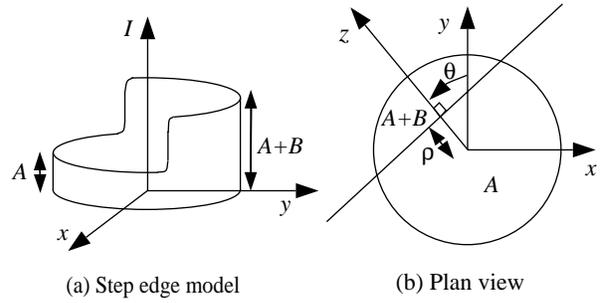
$$z = y \cdot \cos \theta - x \cdot \sin \theta - \rho \quad (7)$$

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: $F_{SE}(x, y; A, B, \theta, \rho, \sigma) =$

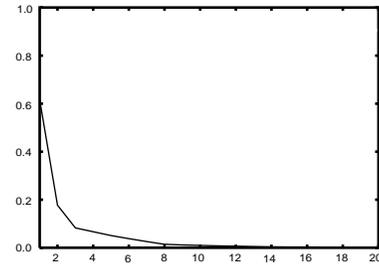
$$\{ (A+B \cdot u(z)) * g(x, y; \sigma) * a(x, y) \} \cdot s(x, y) \quad (8)$$

where z is given by Equation (7).

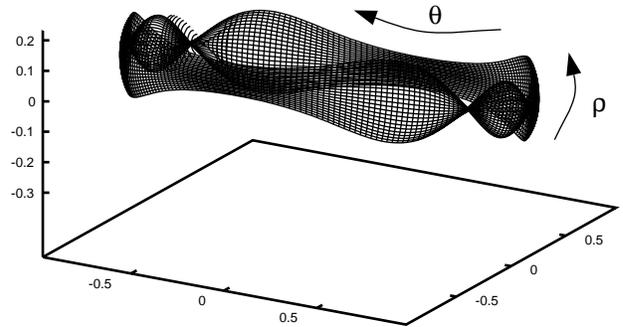
The step edge model has 5 parameters, namely, orientation θ , localization ρ , blurring or scaling σ , and the brightness values A and B . The orientation parameter θ is drawn from $[0^\circ, 360^\circ]$. We restrict the localization parameter ρ to lie in $[-1/\sqrt{2}, 1/\sqrt{2}]$, since any edge must pass closer than $1/\sqrt{2}$ pixels from the center of at least



(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. The step edge manifold is parameterized by orientation and intrapixel localization for a fixed blurring value and is displayed in a 3-D subspace.

one pixel in the image. The blurring parameter $\sigma \in [0.3, 1.5]$. 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 & B and is uniquely determined by the parameters θ , ρ , and σ . Further, the values of A and B may be recovered from the values of μ and ν calculated during normalization [Baker *et al.*, 1998].

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-Loève 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 [Hummel, 1979] 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 in Hummel’s edge model.

In Figure 1(d), the decay of the Karhunen-Loève 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 is 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 while varying θ and ρ .

4 Feature Detection

Given a point in \mathbb{R}^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 densely sampling the manifold and then per-

forming 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. We search using a heuristic coarse-to-fine search which takes advantage of the relatively smooth manifolds [Baker *et al.*, 1998].

As an example of the search complexity for the step edge model, if we sample θ every 1.6° , ρ every 0.088 pixel, and σ every 0.14 pixel, we have 46,368 sample points. Then, in a 10-D subspace, the complete time to perform normalization, projection, and search is around 1ms per image window on a DEC Alpha 3600. For a 512×480 image complete processing takes around 4 minutes. However, by applying rejection techniques such as [Baker and Nayar, 1996a] the overall time can be reduced to under 30secs.

5 Experimental Results

5.1 Feature Detection Rates

We statistically compare our step edge detector with the Canny [1986] and Nalwa-Binford [1986] detectors, following the approach in [Nalwa and Binford, 1986]. (See [Baker *et al.*, 1998] for more details.) 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 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, 1986], by taking a constant intensity window, and adding zero-mean Gaussian noise.

In Figure 2 we compare the detection performance of the three edge detectors. For each pair of S.N.R. and 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.

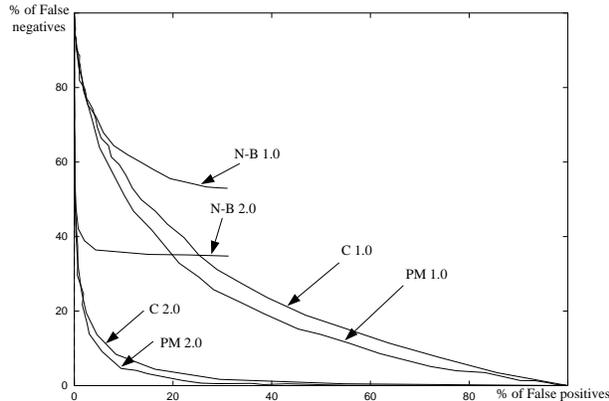


Figure 2: A comparison of edge detection rates. The Canny (C), Nalwa-Binford (N-B), and parametric manifold (PM) detectors are compared for S.N.R. = 1.0 and 2.0. We plot false positives against false negatives. For each detector and S.N.R., the result is a curve parameterized by the threshold inherent 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 comparably.

The closer a curve lies to the origin in Figure 2, 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 consistent³ with those described in [Nalwa and Binford, 1986]. Applied to real images, the Nalwa-Binford detector does not perform as poorly as Figure 2 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, 1986], 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 3, we compare the performance of our step edge detector with that of the Canny detector [1986] and the Nalwa-Binford [1986] 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 perfor-

³We did not use step 2)' of the Nalwa-Binford algorithm, however the inclusion of this step does not radically alter the performance [Nalwa and Binford, 1986].

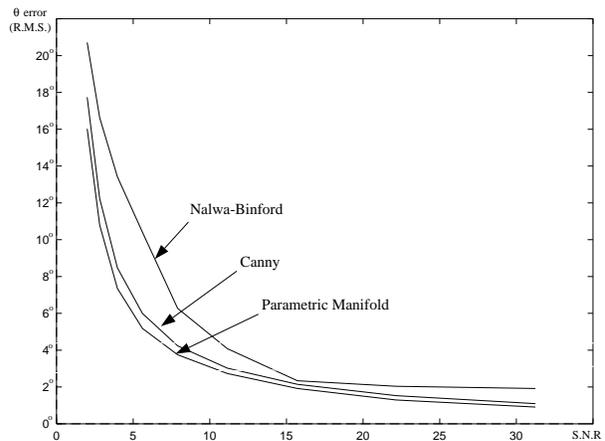


Figure 3: A comparison of the orientation estimation accuracy for the three step edge detectors. We took synthesized step edges, added noise to them, and then applied the edge detectors. We plot the R.M.S. error of the orientation estimate against the S.N.R. At all noise levels, the parametric manifold approach slightly outperforms both the Nalwa-Binford and Canny detectors.

mance of all detectors is limited by the noise. For lower noise levels, our detector marginally outperforms both of the other detectors.

5.3 Application to Images

In Figures 4(b) and (c) we present the results of applying our step edge and corner detectors to the image in Figure 4(a). The original image is taken from [MOMA, 1984] and was digitized using an Envisions 6600S scanner at 200dpi. We present the outputs of the detectors as grey-coded distance to the feature manifold (on a non-linear scale) so that the structure of the object can be seen clearly. It is immediate that the features detected are consistent with the original image. Thresholding on the distance to the feature manifold to finally detect features is straightforward as is demonstrated in [Baker *et al.*, 1998] where we superimpose thresholded feature maps on the original images.

Acknowledgements

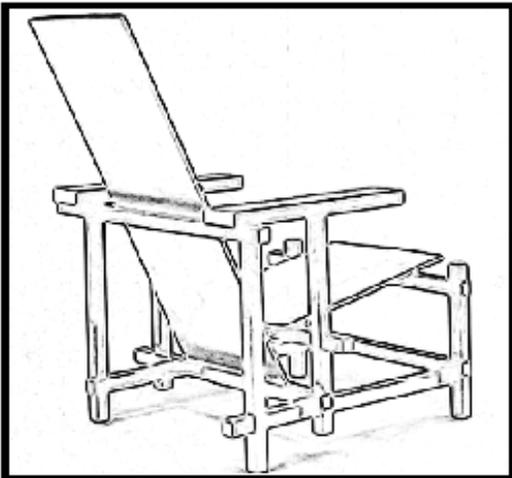
We wish to thank Vic Nalwa for providing an implementation of the Nalwa-Binford edge detector, and Sergio Cuniolo, Michael Heath, Tony Lindenber, Jose-Maria Montiel, and Geoff West for providing pointers to Canny edge detectors.

References

[Baker and Nayar, 1996a] S. Baker and S.K. Nayar. Algorithms for pattern rejection. In *Proceedings of the IAPR International Conference*



(a) Original image (711×661 pixels)



(b) Grey-coded distance to step edge manifold



(c) Grey-coded distance to corner manifold

Figure 4: Results of step edge and corner detection for a 711×661 image of “Red and Blue,” by Gerrit Rietveld, circa 1918. The raw (unthresholded) detector outputs in (b) and (c) reflect high accuracy in detection and localization.

on *Pattern Recognition*, pages 869–874, 1996.

[Baker and Nayar, 1996b] S. Baker and S.K. Nayar. Pattern rejection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 544–549, 1996.

[Baker et al., 1998] S. Baker, S.K. Nayar, and H. Murase. Parametric feature detection. *International Journal of Computer Vision*, 1998. (Accepted for publication).

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

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

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

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

[Hueckel, 1971] M.H. Hueckel. An operator which locates edges in digitized pictures. *Journal of the Association for Computing Machinery*, 18:113–125, 1971.

[Hummel, 1979] R.A. Hummel. Feature detection using basis functions. *Computer Graphics and Image Processing*, 9:40–55, 1979.

[Koenderink, 1984] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[Lenz, 1987] R. Lenz. Optimal filters for the detection of linear patterns in 2-D and higher dimensional images. *Pattern Recognition*, 20:163–172, 1987.

[MOMA, 1984] MOMA. *The Museum of Modern Art New York: The History and the Collection*. Harry N. Abrams, 1984.

[Nalwa and Binford, 1986] 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., 1996] D. Nandy, Z. Wang, J. Ben-Arie, K.R. Rao, and 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, 1996.

[Nayar et al., 1996] S.K. Nayar, S. Baker, and H. Murase. Parametric feature detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 471–477, 1996.

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