Appeared in the Proceedings of the 1998 Image Understanding Workshop.

Optimal Weighting Functions for Feature Detection *

Simon Baker and Shree K. Nayar
Department of Computer Science
Columbia University
New York, NY 10027

Abstract

One approach to feature detection is to match a parametric model of the feature to the image data. Naturally, the performance of such detectors is highly dependent upon the function used to measure the degree of fit between the feature model and the image data. In this paper, we first show how an existing detector can be extended to use a weighted L^2 norm as the matching function with minimal extra computation. Next, we propose optimality criteria for the two fundamental aspects of feature detection performance: feature detection robustness and parameter estimation accuracy. We also show how to combine these criteria in various ways. We analyze the optimality criterion for parameter estimation under the approximating assumption that the feature manifold is locally linear. We also present a numerical algorithm that can be used to estimate the optimal weighting functions for the other optimality criteria. We include the results of applying this algorithm for step edge, line, and corner features.

1 Introduction

In [Baker *et al.*, 1998] we proposed a general purpose feature detection algorithm that uses the model matching approach developed by [Hueckel, 1971], [Nalwa and Binford, 1986], and [Rohr, 1992]. A feature is detected by a model matching detector if there exist parameter values such that the image data and the feature with those parameters are sufficiently "similar." To measure the degree of similarity, a matching function is required. For simplicity, we used the Euclidean L^2 norm as the matching function in [Baker *et al.*, 1998]. In this paper, we show how the matching function can be selected to maximize the performance of the detector.

The selection of the matching function for a feature detector has never before been studied in a systematic manner. In fact, most detectors also use the Euclidean L^2 norm. Examples include [O'Gorman, 1978], [Hummel, 1979], [Morgenthaler, 1981], [Zucker and Hummel, 1981], [Nalwa and Binford, 1986], and [Rohr, 1992]. Other detectors have used weighted L^2 norms, but in all cases the weighting function was chosen in an ad-hoc manner. See, for example, [Hueckel, 1971], [Hueckel, 1973], and [Hartley, 1985]. Note that several other authors have briefly mentioned the possibility of using a weighted L^2 norm without actually doing so themselves [Paton, 1975] [Abdou and Pratt, 1979] [Lenz, 1987].

*This work was supported in part by the ONR/DARPA MURI program under ONR Contract No. N00014-95-1-0601. Several other agencies and companies have also supported aspects of this research.

As with any other matching problem, there is a trade off between the complexity of the matching function and its performance. We begin this paper in Section 2 by showing how the feature detection algorithm of [Baker *et al.*, 1998] can be extended to use an arbitrary weighted L^2 norm with essentially *no extra computation.* Hence, we can consider the class of weighted L^2 norms as possible matching functions, without sacrificing efficiency. In Section 3, we propose optimality criteria for the two key aspects of feature detection performance: *feature detection robustness* and *parameter estimation accuracy.* We also show how to combine these criteria in various ways.

In Section 4.1 we analyze the optimality criterion for parameter estimation under the approximating assumption that the feature manifold is locally linear. We show that the optimal weighting function assigns a weight to each pixel that is inversely proportional to the variance of the noise in that pixel. Unfortunately, extending this analysis to the general case proves to be impossible. Therefore, in Section 4.2 we propose a numerical algorithm that can be used to find the optimal weighted L^2 norm for arbitrary features and any of our optimality criteria. In Section 4.3 we present the results of applying this algorithm for three of the features studied in [Baker *et al.*, 1998]: the *step edge,* the *corner,* and the *symmetric line.* The results demonstrate the major contribution of this paper which is a method of automatically deriving optimal weighting functions for parametric feature detectors. These optimal weighting functions improve the performance of the general purpose feature detector proposed in [Baker *et al.*, 1998] with negligible computational cost.

2 Feature Detection & Weighted L^2 Norms

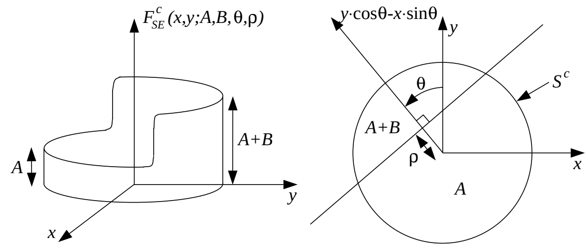
2.1 Background

We follow [Baker *et al.*, 1998] and start with a parametric feature model $F^c(x,y;\mathbf{q}^c)$, where $(x,y) \in S^c$ are points within a continuous feature window $S^c \subseteq \mathbf{R}^2$ and $\mathbf{q}^c$ is a vector containing the feature parameters. The feature model for the step edge is illustrated in Figure 1 and is given by: $F^c_{SE}(x,y;A,B,\theta,\rho) =$

$$\begin{cases} A+B & \text{if } y\cdot\cos\theta - x\cdot\sin\theta \geq \rho \\ A & \text{if } y\cdot\cos\theta - x\cdot\sin\theta < \rho \end{cases}. \quad (1)$$

Because we are trying to detect features in a discrete image $I(n,m)$, where $(n,m) \in \mathbf{Z}^2$, we discretize Equation (1) by modeling the image formation process:

$$F(n,m;\mathbf{q}) = F^c(x,y;\mathbf{q}^c) * g(x,y;\sigma) * a(x,y)|_{x=n,y=m} \quad (2)$$



(a) Step edge model (b) Plan view

Figure 1: The step edge consists of two constant intensity regions of brightness A and $A+B$. The other parameters are the orientation of the edge θ and the sub-pixel localization ρ. An additional blur parameter σ is added during discretization.

where $\mathbf{q} = \mathbf{q}^c \cup \{\sigma\}$ are the discrete model parameters, $(n, m) \in S = S^c \cap \mathbf{Z}^2$ are the pixel coordinates, and $*$ is the 2-D convolution operator. The 2-D Gaussian $g(x, y; \sigma)$ models the blurring of the feature by the optical system and the rectangular averaging function $a(x, y)$ accounts for the integration performed by the sensor. See [Baker *et al.*, 1998] for more details.

If N is the number of pixels in S, then each feature instance $F(n, m; \mathbf{q})$ can be regarded as a vector in $\mathbf{R}^N$. As the parameters vary over their ranges, $F(n, m; \mathbf{q})$ traces out a k-parameter manifold in $\mathbf{R}^N$, where $k = \dim(\mathbf{q})$. Feature detection is then posed as follows. If $(a, b) \in \mathbf{Z}^2$ is a pixel in the input image, find the closest point on the manifold to the vector $I(a+n, b+m) \in \mathbf{R}^N$. If the closest point is sufficiently near, the feature is detected and the parameters of the closest point on the manifold are used as estimates of the feature parameters.

2.2 Feature Detection with Weighted $\mathbf{L^2}$ Norms

Every measure w on S defines a different L^2 norm, denoted by $\mathrm{L}^2(w)$ or $\|\cdot\|_w$. A measure is defined by the weight $w(n, m) \geq 0$ that it assigns to each of the pixels $(n, m) \in S$. If $\mathbf{v}_1 = (v_1(n, m))$ and $\mathbf{v}_2 = (v_2(n, m))$ are two vectors, their weighted inner product[1] is:

$$\langle \mathbf{v}_1, \mathbf{v_2} \rangle_w \;=\; \sum_{(n,m)\in S} w(n, m) \cdot v_1(n, m) \cdot v_2(n, m) \quad (3)$$

and the $\mathrm{L}^2(w)$ norm of $\mathbf{v}_1$ is: $\|\mathbf{v}_1\|_w = [\,\langle \mathbf{v}_1, \mathbf{v}_1\rangle_w\,]^{1/2}$. The Euclidean L^2 norm is distinguished as the weighted L^2 norm for which the measure of each pixel is 1.0.

Since the square root function is monotonic and computing the square of an L^2 norm is easier than computing the L^2 norm itself, we actually base feature detection upon the square of the distance to the closest manifold point: $\min_{\mathbf{q}} \|F(n, m; \mathbf{q}) - I(a+n, b+m)\|^2_w =$

$$\min_{\mathbf{q}} \sum_{(n,m)\in S} w(n, m)\,[F(n, m; \mathbf{q}) - I(a+n, b+m)]^2\,. \quad (4)$$

The algorithm in [Baker *et al.*, 1998] that is used to find the parameters $\mathbf{q}$ that actually minimize the expression in Equation (4) consists of three steps:

1. Apply a parameter normalization to eliminate two of the intensity parameters from $\mathbf{q}$.
2. Use the Karhunen-Loéve expansion to reduce the dimension of the space that the manifold lies in.
3. Find the closest point on the feature manifold using a coarse-to-fine search.

Using a weighted L^2 norm poses no difficulty for the coarse-to-fine search. I now describe how parameter normalization and dimension reduction can be performed using a weighted L^2 norm.

2.3 Parameter Normalization

For each feature $F(n, m; \mathbf{q})$ the mean coordinate $\mu(\mathbf{q}) = \frac{1}{N}\sum_{(n,m)\in S} F(n, m; \mathbf{q})$ and the coordinate standard deviation $\nu(\mathbf{q}) = [\,\sum_{(n,m)\in S}[F(n, m; \mathbf{q}) - \mu(\mathbf{q})]^2\,]^{1/2}$ are computed. Then, the feature instance is normalized:

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

For many features this simple normalization reduces the number of parameters by two because $\overline{F}(n, m; \mathbf{q})$ turns out to be independent of two of the parameters in $\mathbf{q}$.

If $\mathbf{c} = (1, 1, \ldots, 1) \in \mathbf{R}^N$ and $\hat{\mathbf{c}} = \mathbf{c}/\|\mathbf{c}\|$ then Equation (5) can be rewritten as:

$$\overline{F}(n, m; \mathbf{q}) \;=\; \frac{F(n, m; \mathbf{q}) \,-\, \langle F(n, m; \mathbf{q}), \hat{\mathbf{c}}\rangle\, \hat{\mathbf{c}}}{\|F(n, m; \mathbf{q}) \,-\, \langle F(n, m; \mathbf{q}), \hat{\mathbf{c}}\rangle\, \hat{\mathbf{c}}\|} \quad (6)$$

where $\|\cdot\|$ is the Euclidean L^2 norm and $\langle\,\cdot\,,\,\cdot\,\rangle$ is the Euclidean inner product. When using a weighted L^2 norm, normalization is performed exactly as in Equation (6) except that the weighted inner product and weighted L^2 norm are used in place of their Euclidean equivalents.

2.4 Dimension Reduction

In [Baker *et al.*, 1998] the Karhunen-Loéve (K-L) expansion [Fukunaga, 1990] is applied as a dimension reduction technique. Applying dimension reduction when using a weighted L^2 norm is straightforward. If $\{\mathbf{e}^j \mid j = 1, 2, \ldots, N\}$ is an orthonormal basis with respect to $\langle\cdot,\cdot\rangle_w$, then: $\|\overline{F}(n, m; \mathbf{q}) - \overline{I}(a+n, b+m)\|^2_w =$

$$\sum_{j=1}^{N} \left[\langle \overline{F}(n, m; \mathbf{q}), \mathbf{e}^j\rangle_w - \langle \overline{I}(a+n, b+m), \mathbf{e}^j\rangle_w\right]^2. \quad (7)$$

After a change of basis, dimension reduction corresponds to discarding a number of the basis vectors (without loss of generality the last few) and restricting attention to the low dimensional subspace spanned by $\{\mathbf{e}^j \mid j = 1, 2, \ldots, d\}$. After applying dimension reduction we approximate: $\|\overline{F}(n, m; \mathbf{q}) - \overline{I}(a+n, b+m)\|^2_w \approx$

$$\sum_{j=1}^{d} \left[\langle \overline{F}(n, m; \mathbf{q}), \mathbf{e}^j\rangle_w - \langle \overline{I}(a+n, b+m), \mathbf{e}^j\rangle_w\right]^2. \quad (8)$$

Since $\langle \mathbf{a}, \mathbf{e}^j\rangle_w$ is the j^{th} component of $\mathbf{a}$ in the low dimensional subspace, we see that the weighted L^2 norm

[1] For Equation (3) to define an inner product, rather than a semi-inner product, we strictly require $w(n, m) > 0$ for all $(n, m) \in S$.

can be computed with exactly the same cost as the Euclidean L^2 norm: ie. $3 \cdot d - 1$ arithmetic operations.

The Karhunen-Loéve expansion for a weighted L^2 norm is computed as follows. The $N \times N$ weighted covariance matrix $\mathbf{C} = (C_{nm,pq})$ is computed using:

$$C_{nm,pq} = \mathrm{E}_{\mathbf{q}} \left[\overline{G}(n,m;\mathbf{q}) \cdot w(p,q) \cdot \overline{G}(p,q;\mathbf{q}) \right] \quad (9)$$

where $\mathrm{E}[\cdot]$ is the expectation operator and $\overline{G}(n,m;\mathbf{q}) = \overline{F}(n,m;\mathbf{q}) - \mathrm{E}_{\mathbf{q}}\left[\overline{F}(n,m;\mathbf{q})\right]$. Then, the d eigenvectors of $\mathbf{C}$ with the largest eigenvalues are used for $\mathbf{e}^1 \ldots \mathbf{e}^d$

3 Optimality Criteria

3.1 Feature Detection Robustness

Feature detection is not robust when either the detector fails to detect a feature in the scene (a false negative) or mistakenly detects a feature that is not present in the scene (a false positive.) To define our optimality criteria for feature detection robustness, we assume that, in addition to a parametric model of the feature, we also have a parametric model of what is not the feature; ie we have a model for the non-feature $NF^c(x,y;\mathbf{nq}^c)$. For the step edge model in Figure 1, a suitable non-feature might be a constant gradient slope with 3 parameters:

$$NF^c_{SE}(x,y;A,g,\theta) \;=\; A + g\cdot(y \cdot \cos\theta - x \cdot \sin\theta) \quad (10)$$

where A is the intensity of the center pixel, g is the magnitude of the gradient of the slope, and θ is the angle that the direction of steepest ascent makes with the positive y-axis. The non-feature model is discretized in an analogous way to the way the feature was in Equation (2).

Assume that there is a feature with parameters $\mathbf{q}$ in the scene. If this feature is projected onto a window surrounding the pixel (a,b) in the image I, the vector $I(a+n, b+m)$ should differ from $F(n,m;\mathbf{q})$ only because of the noise introduced in the imaging process. If this image noise is modeled with the additive random vector $\boldsymbol{\eta} = (\eta(n,m))$, we have:

$$I(a+n,b+m) \;=\; F(n,m;\mathbf{q}) + \eta(n,m). \quad (11)$$

We assume that this image vector is mistakenly classified as a non-feature if it is closer to the non-feature manifold than it is to the feature manifold; ie. if:

$$\min_{\mathbf{nq}'} \|I(a+n,b+m) - NF(n,m;\mathbf{nq}')\|^2_w \;<\; \min_{\mathbf{q}'} \|I(a+n,b+m) - F(n,m;\mathbf{q}')\|^2_w\,. \quad (12)$$

We therefore estimate the probability of a false negative for a feature with parameters $\mathbf{q}$ to be: $\mathrm{P}_{\boldsymbol{\eta}}(\mathrm{FN} \mid \mathbf{q}) =$

$$\mathrm{P}_{\boldsymbol{\eta}} \left[\min_{\mathbf{nq}'} \|F(n,m;\mathbf{q}) + \eta(n,m) - NF(n,m;\mathbf{nq}')\|^2_w < \min_{\mathbf{q}'} \|F(n,m;\mathbf{q}) + \eta(n,m) - F(n,m;\mathbf{q}')\|^2_w \right]. \quad (13)$$

Now, $\mathrm{P}_{\boldsymbol{\eta}}(\mathrm{FN} \mid \mathbf{q})$ is still a function of $\mathbf{q}$. Hence, we assume that $\mathrm{P}(\mathbf{q})$ denotes the *a priori* likelihood of a feature $F(m,n;\mathbf{q})$ with parameters $\mathbf{q}$ appearing in the scene. Finally, we propose the following as our optimality criterion for false negatives:

$$\mathbf{FN} \;=\; \int \mathrm{P}(\mathbf{q})\, \mathrm{P}_{\boldsymbol{\eta}}(\mathrm{FN} \mid \mathbf{q})\, \mathrm{d}\mathbf{q}. \quad (14)$$

By reversing the roles of the feature and the non-feature, we propose the following as our optimality criterion for false positives:

$$\mathbf{FP} \;=\; \int \mathrm{P}(\mathbf{nq})\, \mathrm{P}_{\boldsymbol{\eta}}(\mathrm{FP} \mid \mathbf{nq})\, \mathrm{d}\mathbf{nq} \quad (15)$$

where $\mathrm{P}_{\boldsymbol{\eta}}(\mathrm{FP} \mid \mathbf{nq})$ is defined in an analogous way to $\mathrm{P}_{\boldsymbol{\eta}}(\mathrm{FN} \mid \mathbf{q})$ and $\mathrm{P}(\mathbf{nq})$ is the *a priori* likelihood of the occurrence of the non-feature $NF(m,n;\mathbf{nq})$.

3.2 Parameter Estimation Accuracy

Parameter estimation is inaccurate when the closest manifold point is not at the correct place on the manifold. Again, we model the imaging noise with the additive random vector $\boldsymbol{\eta} = (\eta(n,m))$. If the closest point on the manifold to $I(a+n,b+m)$ is $F(n,m;\mathbf{q}+\Delta\mathbf{q})$, the error in estimating parameter q_i will be $\Delta q_i = \Delta q_i(\boldsymbol{\eta})$. To obtain a measure which is independent of the particular noise added, we average over the noise distribution by taking the root mean squared error:

$$\mathrm{RMS}_{\boldsymbol{\eta}}[\Delta q_i(\boldsymbol{\eta})] \;=\; \sqrt{\mathrm{E}_{\boldsymbol{\eta}}[(\Delta q_i(\boldsymbol{\eta}))^2]}. \quad (16)$$

Again, $\mathrm{RMS}_{\boldsymbol{\eta}}[\Delta q_i(\boldsymbol{\eta})]$ is still a function of the parameters $\mathbf{q}$. Hence, we propose the following as our measure of the parameter estimation accuracy for parameter q_i:

$$\mathbf{PE}_{q_i} \;=\; \left(\int \mathbf{P}(\mathbf{q})\, \left(\mathrm{RMS}_{\boldsymbol{\eta}}[\Delta q_i(\boldsymbol{\eta})]\right)^2\, \mathrm{d}\mathbf{q} \right)^{1/2} \quad (17)$$

where $\mathrm{P}(\mathbf{q})$ is the *a priori* likelihood of the feature $F(m,n;\mathbf{q})$ appearing in the scene.

3.3 Combinations of Optimality Criteria

In addition to the probability of a false positive $\mathbf{FP}$, the probability of a false negative $\mathbf{FN}$, and the parameter estimation accuracy $\mathbf{PE}_{q_i}$, it is straightforward to also consider combinations of these optimality criteria. For example, we considered the overall feature detection robustness $\mathbf{ODR} = \sqrt{\mathbf{FP} \times \mathbf{FN}}$, the combined parameter estimation accuracy:

$$\mathbf{CPE} \;=\; \left(\prod_{i=1}^{|\mathbf{q}|} \mathbf{PE}_{q_i} \right)^{1/|\mathbf{q}|}, \quad (18)$$

and the overall feature detection performance $\mathbf{ODP} = \mathbf{ODR} \times \mathbf{CPE}$.

4 Optimization of the Optimality Criteria

4.1 Analysis for Linear Manifolds

We derived the optimal weighting functions for $\mathbf{PE}_{q_i}$ assuming that the feature manifolds are approximated with

Table 1: Computed values of the optimality criteria for the step edge.

	$\mathbf{PE}_A$	$\mathbf{PE}_B$	$\mathbf{PE}_\theta$	$\mathbf{PE}_\rho$	$\mathbf{PE}_\sigma$	$\mathbf{CPE}$	$\mathbf{FDR}$	$\mathbf{ODP}$
Euclidean	0.261	0.438	6.982	0.441	0.509	0.709	0.390	0.825
Optimal	0.234	0.372	6.957	0.438	0.507	0.706	0.376	0.822

Table 2: Computed values of the optimality criteria for the corner.

	$\mathbf{PE}_A$	$\mathbf{PE}_B$	$\mathbf{PE}_{\theta_1}$	$\mathbf{PE}_{\theta_2}$	$\mathbf{PE}_\sigma$	$\mathbf{CPE}$	$\mathbf{FDR}$	$\mathbf{ODP}$
Euclidean	0.120	0.519	10.448	17.813	0.288	1.274	0.234	0.897
Optimal	0.112	0.459	10.398	17.738	0.287	1.251	0.221	0.854

Table 3: Computed values of the optimality criteria for the symmetric line.

	$\mathbf{PE}_A$	$\mathbf{PE}_B$	$\mathbf{PE}_\theta$	$\mathbf{PE}_\rho$	$\mathbf{PE}_\sigma$	$\mathbf{PE}_w$	$\mathbf{CPE}$	$\mathbf{FDR}$	$\mathbf{ODP}$
Euclidean	0.276	0.690	4.510	0.216	0.355	0.992	0.635	0.00358	0.00681
Optimal	0.223	0.654	4.500	0.215	0.354	0.989	0.632	0.00238	0.00475

their first-order Taylor expansions. This assumption is reasonable when the noise level is not too high. For linear manifolds, finding the closest point on the manifold is simply a weighted least squares problem. So, finding the optimal weighting function corresponds to selecting the weighting function that gives the best linear unbiased estimate of the solution to a weighted least squares problem [Lawson and Hanson, 1974] [Plackett, 1960]. The answer to this problem was found by Aitken in [Aitken, 1934]. Assuming the noise is independently distributed across the pixels, the optimal weighting function is:

$$w(n,m) \;=\; \frac{1}{\mathrm{Var}_{\boldsymbol{\eta}}[\eta(n,m)]} \qquad (19)$$

where $\mathrm{Var}_{\boldsymbol{\eta}}[\eta(n,m)] = \mathrm{E}_{\boldsymbol{\eta}}[\eta^2(n,m)] - \mathrm{E}_{\boldsymbol{\eta}}[\eta(n,m)]^2$ is the variance of the noise $\boldsymbol{\eta}$ in pixel (n,m). Note that Equation (19) implies that if the noise $\boldsymbol{\eta}$ is independently and identically distributed across the pixels the Euclidean L^2 norm is optimal for parameter estimation accuracy.

4.2 Numerical Optimization

The use of realistic multi-parameter feature models, combined with a non-linear sensing model, leads to very complicated expressions for the optimality criteria. Canny used a numerical algorithm to optimize his relatively simple optimality criterion for arbitrary 1-D features in [Canny, 1986]. Here, I follow the same approach for arbitrary 2-D features.

Since N is typically in the range 25–100, to make the problem tractable I assume that the weighting function is rotationally symmetric. Such an assumption is reasonable if there is no *a priori* knowledge about the orientation of features in the image. I also assume that the weighting function can be approximated by a low order polynomial in the distance from the center of the window. I found that a degree 5 polynomial was sufficient, and that the best way to parameterize this polynomial is in terms of the value of the polynomial at equally spaced distances from the center of the window. The radius of the window was 4 pixels in my experiments. Hence, I parameterized the polynomial in terms of its value at distance $0, 1, 2, 3, 4$ pixels from the center.

I used Powell's Method [Press *et al.*, 1992] to compute the optimal weighting functions. Powell's Method works by repeatedly performing 1-dimensional optimizations though the current best estimate of the minimum. Each 1-dimensional optimization is performed using Brent's Method, which iteratively samples the optimality criterion, performs a parabolic fit, and then replaces one of the sample points with the minimum of the parabola [Press *et al.*, 1992]. After each application of Brent's Method, the best estimate of the minimum is updated and a new direction is chosen. The key decision in Powell's Method is how to choose and update the set of directions in which the 1-dimensional optimizations are performed. I used the technique of "discarding the direction of largest decrease." described in [Press *et al.*, 1992].

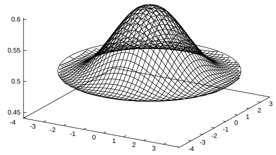
Using Monte Carlo integration to perform the averaging over the parameter space and the noise distribution, it takes approximately 1–2 minutes to evaluate any of my optimality criteria at a single sample point. Typically, Powell's Method samples the optimality criteria a few hundred times resulting in an overall running time of a few hours, which is fine for a preprocessing step.

4.3 Numerical Results

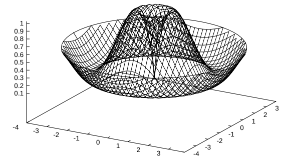
I applied Powell's algorithm to three of the features considered in [Baker *et al.*, 1998], namely, the step edge, the corner, and the symmetric line. Some of the optimality criteria require non-feature models as well as feature models. I used the constant gradient slope defined in Equation (10) as the non-feature for the step edge, and the step edge itself as the non-feature for both the corner and the symmetric line. The optimality criteria also require probability distributions to be provided for the feature parameters and the noise. I assumed that $\mathrm{P}(\mathbf{q})$ and $\mathrm{P}(\mathbf{nq})$ are both uniformly distributed. The noise $\boldsymbol{\eta}$ was chosen to be Gaussian and independently distributed across the pixels, with signal to noise ratio 2.0.

The results of applying Powell's algorithm to the three features are displayed in Tables 1–3. In each table, I present the numeric value of the optimality criteria for both the Euclidean L^2 norm and for the optimal L^2 norm. There are a couple of points that should be noted:

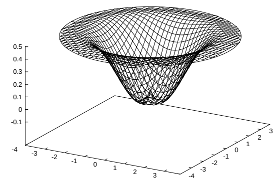
- The parameter estimation accuracy results are in close agreement with the analysis of Section 4.1. There I showed that the Euclidean L^2 norm would be optimal if the manifolds were linear. Therefore it



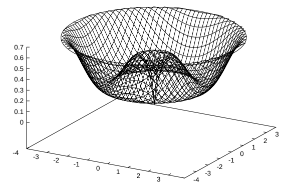
(a) $\mathbf{PE}_\rho$ for the Step Edge



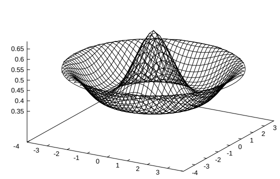
(b) $\mathbf{FDR}$ for the Step Edge



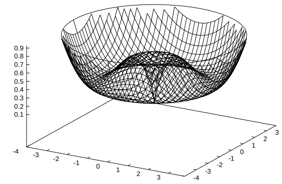
(c) $\mathbf{CPE}$ for the Corner



(d) $\mathbf{ODP}$ for the Corner



(e) $\mathbf{CPE}$ for the Line



(f) $\mathbf{FDR}$ for the Line

Figure 2: A selection of optimal weighting functions computed by applying Powell's algorithm to the optimality criteria proposed in Section 3.

is no surprise that the optimal weighted L^2 norm is only marginally better than the Euclidean L^2 norm for the non-normalized parameters.

- For feature detection robustness, significant improvement is possible over the Euclidean L^2 norm. For the step edge and corner the improvement if around 5%, and for the symmetric line it is around 35%. For the overall detection performance, almost no improvement is possible for the step edge, a 5% improvement is possible for the corner, and a 30% improvement is possible for the line.

In Figure 2, I display 3–D plots of several of the optimal weighting functions. The optimal weighting function for the parameter estimation accuracy of the sub-pixel localization ρ of the step edge is presented in Figure 2(a). The overall form of this optimal weighting function is as expected. Intuitively, the center-most pixels are the most important when estimating sub-pixel localization [Canny, 1986]. The results in Figure 2(c) for the combined parameter estimation accuracy of the corner are also in agreement with intuition. The central pixels do not change much as the parameters of the corner vary, so it is natural that they are substantially down-weighted.

5 Conclusion

The major contribution of this paper is a framework for the estimation of optimal weighting functions for feature detection. We have proposed optimality criteria for the two fundamental aspects of feature detection performance and have presented a numerical algorithm that can be used to automatically compute the optimal weighting function for almost any feature and optimality criterion. The resulting optimal weighted L^2 norms improve the general purpose feature detector of [Baker *et al.*, 1998] with negligible computational cost.

References

[Abdou and Pratt, 1979] I.E. Abdou and W.K. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5):753–763, May 1979.

[Aitken, 1934] A.C. Aitken. On least squares and linear combinations of observations. *Proceedings of the Royal Society of Edinburgh A*, 55:42–47, 1934.

[Baker *et al.*, 1998] S. Baker, S.K. Nayar, and H. Murase. Parametric feature detection. *International Journal of Computer Vision*, 27(1):27–50, 1998.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

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

[Hartley, 1985] R. Hartley. A Gaussian-weighted multiresolution edge detector. *Computer Vision, Graphics, and Image Processing*, 30:70–83, 1985.

[Hueckel, 1971] M.H. Hueckel. An operator which locates edges in digitized pictures. *Journal of the Association for Computing Machinery*, 18(1):113–125, January 1971.

[Hueckel, 1973] M.H. Hueckel. A local visual operator which recognizes edges and lines. *Journal of the Association for Computing Machinery*, 20(4):634–647, October 1973.

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

[Lawson and Hanson, 1974] C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

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

[Morgenthaler, 1981] D.G. Morgenthaler. A new hybrid edge detector. *Computer Graphics and Image Processing*, 16:166–176, 1981.

[Nalwa and Binford, 1986] V.S. Nalwa and T.O. Binford. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):699–814, November 1986.

[O'Gorman, 1978] F. O'Gorman. Edge detection using Walsh functions. *Artificial Intelligence*, 10:215–223, 1978.

[Paton, 1975] K. Paton. Picture description using Legendre polynomials. *Computer Graphics and Image Processing*, 4:40–54, 1975.

[Plackett, 1960] R.L. Plackett. *Principles of Regression Analysis*. Oxford Univeristy Press, 1960.

[Press *et al.*, 1992] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.

[Rohr, 1992] K. Rohr. Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, 9(3):213–230, 1992.

[Zucker and Hummel, 1981] S.W. Zucker and R.A. Hummel. A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3):324–331, May 1981.