On the Parameter Estimation Accuracy of Model-Matching Feature Detectors

(Technical Report CUCS-011-97)

Simon Baker

Department of Computer Science
Columbia University
New York, NY 10027

Email: simonb@cs.columbia.edu

Abstract

The performance of model-fitting feature detectors is critically dependent upon the function used to measure the degree of fit between the feature model and the image data. In this paper, we consider the class of weighted L^2 norms as potential fitting functions and study the effect which the choice of fitting function has on one particular aspect of performance, namely parameter estimation accuracy. We first derive an optimality criterion based upon how far an ideal feature instance is perturbed around the feature manifold when noise is added to it. We then show that a first-order (linear) approximation to the feature manifold results in the Euclidean L^2 norm being optimal. We next show empirically that for non-linear manifolds the Euclidean L^2 norm is no longer, in general, optimal. Finally, we present the results of several experiments comparing the performance of various weighting functions on a number of ubiquitous features.

Contents

1 Introduction

Many computer vision algorithms rely upon some form of local feature detection. Moreover, high performance feature detection is nearly always vital to subsequent processing. Although there are several different aspects to the performance of a feature detector [Abdou and Pratt, 79], here we will just consider one of them, that being parameter estimation accuracy. We also only consider a certain type of feature detector. All but a small number of feature detectors can be placed into one of two categories [Nalwa, 93]: those based upon differential invariants and those which fit parametric models to the image data. In this paper, we restrict attention to the model-fitting approach best exemplified by [Nalwa and Binford, 86], [Rohr, 92], and [Nayar *et al.*, 96].

Model-fitting feature detectors work by finding the model parameters which yield the closest match between the image data and the feature model. If the degree of fit is sufficiently good, the feature is detected and the parameters of the closest model instance are used as estimates of the feature parameters. The performance of such detectors is highly dependent upon the function used to measure the degree of fit between the model and data. Changing the fitting function will, in general, not only affect the set of pixels at which the feature is detected, but also the estimates of the feature parameters. Hence, parameter estimation accuracy is also highly dependent upon the fitting function.

The importance of the fitting function is demonstrated empirically by the results of an experiment presented in Figure 1. In this figure we plot the accuracy of estimates of the localization parameter of a step edge (on the ordinate) against the level of noise (on the abscissa) for two different distance functions, the Euclidean (unweighted) L^2 norm and a weighted L^2 norm. The step edge model which we used is outlined in Section 2 and the details of how the experiment was conducted are described in Appendix A.1. For now, the key point to note is the large effect which the choice of the fitting function has on the accuracy of parameter estimation. In this case, the weighted L^2 norm performs far better (by almost a factor of 2) than the Euclidean L^2.

As far as we can tell, the selection of the fitting function has never before been studied in a systematic manner. In fact, most detectors simply use the Euclidean L^2 norm. Examples include [Hummel, 79], [Rohr, 92], and [Baker *et al.*, 98]. The remaining detectors mostly use weighted L^2 norms, but in all cases the weighting function was chosen in an ad-hoc manner. See for example [Hueckel, 71], [Hueckel, 73], and [Hartley, 85]. Here, we consider the entire class of weighted L^2 norms as possible fitting functions and investigate the selection of the best one. As can be seen from Figure 1, this class contains functions with vastly different performance levels.

We begin by defining an optimality criterion based upon the assumption that inaccurate parameter estimation is caused by ideal features in the real world being corrupted by noise during the imaging process. Since we will represent features by their manifolds, in the same manner as [Nayar *et al.*, 96], we model this corruption by perturbing ideal

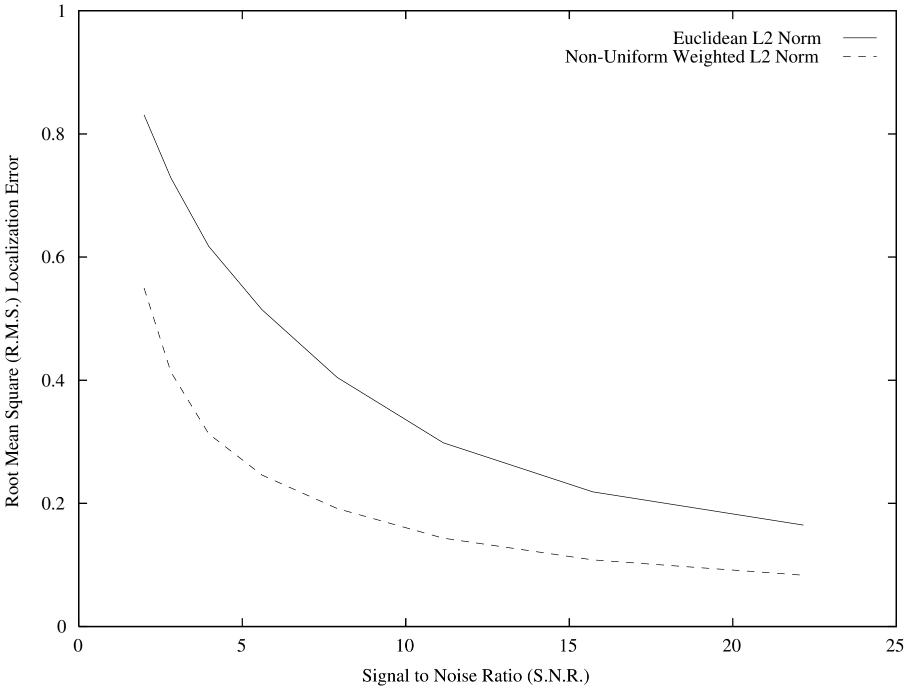


Figure 1: A comparison of the localization estimation accuracy using 2 different fitting functions for the step edge model introduced in Section 2. One of the fitting functions is the Euclidean L^2 norm and the other is a non-uniformly weighted L^2 norm. The weighting function for the non-uniform case is illustrated in Figure 2. The results demonstrate that a considerable performance improvement is possible by carefully selecting a non-uniform weighting function rather than simply using the Euclidean L^2 norm. The details of how this experiment was performed are described in Appendix A.1.

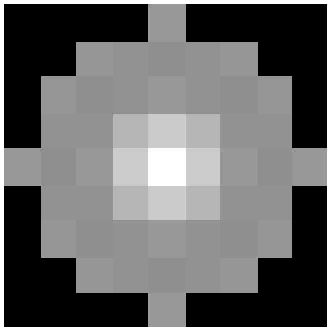


Figure 2: The weighting function used in the experiment of Figure 1. Greatest weight is given to the center-most pixels. The outer pixels are given much less weight but are all roughly equally weighted. No significance should be attached to the exact form of this weighting function. It is simply a weighting function which we discovered empirically to perform relatively well for the localization parameter of the step edge.

manifold points with noise and then measuring how far around the manifold the closest manifold point is perturbed. We average the distance perturbed around the manifold, first over the distribution of the noise and then over the parameter space, to yield our optimality criterion.

We analyze our optimality criteria in the simplified case in which the manifold is linear, or can be approximated as such. We show that, for a fairly general noise model, the optimal weighting function assigns a weight to each pixel which is inversely proportional to the variance of the noise at that pixel. Restricting to the scenario in which the variance of the noise in each pixel is the same, this means that the Euclidean L^2 norm is optimal for all linear manifolds under this noise model. This fact is somewhat surprising considering the results in Figure 1, and it implies that a linear model is not accurate enough for the step edge manifold.

Analyzing a second-order approximation to the manifold is naturally the next step, but it is significantly more involved and so is left to a separate future study. Our final contribution is to present the results of an empirical study investigating the performance of a large number of weighting functions on three ubiquitous features: the step edge, the corner and the (symmetric) line. During this investigation we considered most of the weighting functions which have been proposed in the literature (see Section 1.1) as well as a number of other plausible alternatives.

1.1 Related Work

Although the selection of an appropriate distance function is crucial to the performance of a feature detector, somewhat surprisingly, the issue has never before been studied in a systematic manner. Most existing detectors simply use the Euclidean L^2 norm, often without any discussion of the decision, including [O'Gorman, 78], [Hummel, 79], [Morgenthaler, 81], [Zucker and Hummel, 81], [Nalwa and Binford, 86], [Rohr, 92], and [Nayar *et al.*, 96].

Weighted L^2 norms have been used in a small number of previous feature detection algorithms, and date back to the work of Hueckel. In the continuous domain of [Hueckel, 71], Hueckel used the weighting function $w(x, y) = [1 - (x^2 + y^2)]^{1/2}$ where (x, y) are coordinates relative to the center of a circular feature window with unit radius. The informal justifications provided for this choice are: (a) the weighting function should be continuous, including at the periphery of the window, and (b) its value should decrease monotonically moving away from a maximum at the center of the window. In [Hueckel, 73] the weighting function remains the same to allow a closed form solution for the parameters in the norm minimization problem. However, in the appendix Hueckel suggests that, neglecting computational issues, a Gaussian weighting function may be a more appropriate choice.

Besides Hueckel's work, very few other detectors actually use non-uniform weighting functions. One example which follows Hueckel's suggestion of using a Gaussian weighting function is [Hartley, 85]. In [Lenz, 87], Lenz concentrates on the Euclidean L^2 norm, but extends some of his results to the L^2 norm in polar coordinates (with no weighting function). In Cartesian coordinates this corresponds to the weighting function $w(x, y) = 1/(x^2+y^2)^{1/2}$. A final example is [Abdou and Pratt, 79] in which it is mentioned that weighting pixels so as to reduce the influence of pixels which are distant from the center of the window improves Pratt's Figure of Merit [Pratt, 90], but few details of this are actually given.

Some papers do discuss the possibility of using a weighting function, but end up using the Euclidean L^2 norm. One example is Paton [Paton, 75], who proposes a number of alternatives including a sequence of functions similar to Hueckel's and an annular stop function which interestingly assigns zero weight to the center of the feature window:

$$w(x, y) = \begin{cases} k & \text{if } 0 < r^2 \leq x^2 + y^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $r \in (0, 1)$ is the radius of the inner boundary of the annular stop. There is no discussion of when the use of such a weighting function might be appropriate.

Weighting functions have also been used in differential-invariant based approaches to feature detection. One way to approximate the partial derivatives needed to compute the differential invariants is to first fit a surface, and to then use the partial derivatives of the surface as estimates of the partial derivatives in the underlying image. One example of such an approach is [Haralick 84]. Another example is [Meer and Weiss, 92] where both unweighted and Gaussian weighted L^2 norms are used to define and then find the closest-fitting low-order polynomial surface.

1.2 Summary

The remainder of this paper is organized as follows. In Section 2 we introduce the model-fitting approach to feature detection and describe how weighted L^2 norms may be used to measure the degree of fit between the model and image data. We also discuss how the parameter normalization and dimension reduction techniques of [Baker *et al.*, 98] can be extended to cope with the weighted L^2 norm case. In Section 3 we begin by deriving our optimality criterion and discussing a couple of simplifying assumptions. We then analyze the optimality criterion for linear manifolds and prove that the Euclidean L^2 norm is optimal assuming the noise to be uniformly distributed across the pixels. In Section 4 we present the results of the empirical comparison of various weighting functions for the step edge, corner, and line. Finally we conclude in Section 5 with a discussion of the relationship between our work and the optimal-filtering approach to edge detection best exemplified by [Canny, 86]. We also describe several open problems and opportunities for future work.

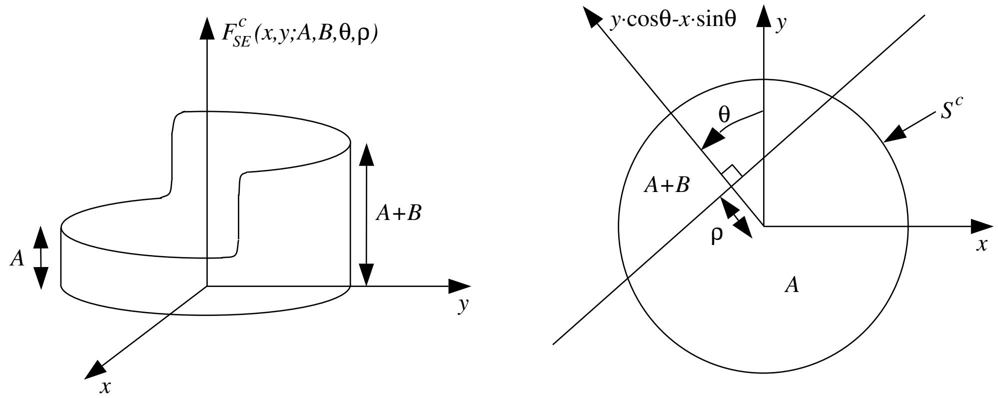


Figure 3: An example feature model. The model of the step edge includes two constant intensity regions of brightness A and $A + B$. Its other parameters are the orientation of the edge θ, the intrapixel displacement (localization) of the edge ρ, and the blur σ. This final parameter σ is introduced to model the effect of the imaging process on the step edge.

2 Background

We begin this section with a brief overview of the model-fitting approach to feature detection. Afterwards we proceed to describe the use of weighted L^2 norms to measure the degree of fit between feature models and image data. Finally, we analyze the effect that the use of a weighted L^2 norm has on the efficiency enhancing techniques of parameter normalization and dimension reduction [Baker *et al.*, 98].

2.1 Parametric Model Fitting

The approach we take to feature detection is to fit a parametric model to the image data. We follow [Nayar *et al.*, 96] and start with a continuous feature model $F^c(x, y; \mathbf{q}^c)$, where $(x, y) \in S^c$ are points within a compact feature window $S^c \subseteq \mathbf{R}^2$ and $\mathbf{q}^c$ is a vector containing the feature parameters. An example feature model for the step edge is illustrated in Figure 3 and is given by:

$$F^c_{SE}(x, y; A, B, \theta, \rho) \;=\; \begin{cases} A + B & \text{if } y \cdot \cos\theta - x \cdot \sin\theta \geq \rho \\ A & \text{if } y \cdot \cos\theta - x \cdot \sin\theta < \rho \end{cases} \tag{2}$$

Because we are trying to detect the features in a discrete image $I(n, m)$, where $(n, m) \in \mathbf{Z}^2$, we discretize[1] the feature model by incorporating the effects of the image

[1] Alternatively, it is possible to regard $I(n, m)$ as samples of an underlying intensity surface $I^c(x, y)$

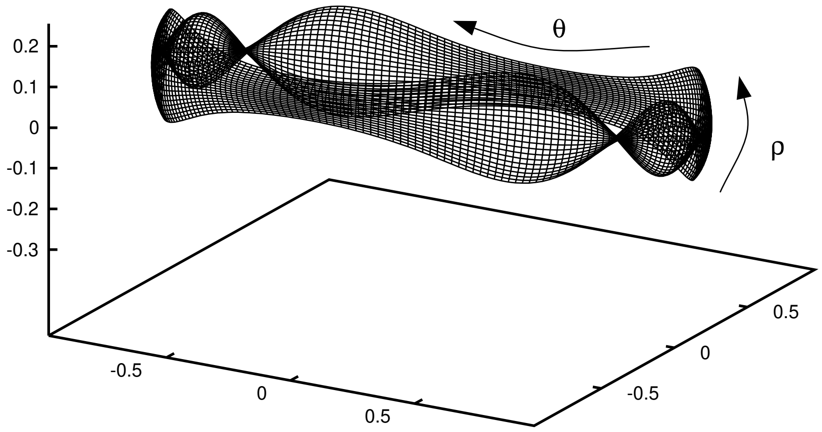


Figure 4: A view of the step edge manifold projected into $\mathbf{R}^3$. The manifold is actually embedded in $\mathbf{R}^{49}$ since we used $N = 49$ pixels, but we are unable to display such a high dimensional space. For reasons of clarity we also only display a slice through the manifold keeping three parameters (A, B, and σ) constant while allowing the other two (θ and ρ) to vary.

formation process:

$$F(n, m; \mathbf{q}) \;=\; F^c(x, y; \mathbf{q}^c) \,*\, g(x, y; \sigma) \,*\, a(x, y) \,|_{x=n, y=m} \tag{3}$$

where $\mathbf{q} = \mathbf{q}^c \cup \{\sigma\}$ are the discrete 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 CCD sensor. See [Nayar *et al.*, 96] for more details.

If N is the total number of pixels in the discrete feature window S, then each feature instance $F(n, m; \mathbf{q})$ may be regarded as a vector in $\mathbf{R}^N$. As the parameters $\mathbf{q}$ vary over their ranges, $F(n, m; \mathbf{q})$ traces out a k-parameter manifold in $\mathbf{R}^N$ where $k = \dim(\mathbf{q})$ is the number of feature parameters. The manifold for the step edge model just presented in Equation (2) is illustrated in Figure 4. Feature detection is then posed as follows. If $(a, b) \in \mathbf{Z}^2$ is a pixel in the input image, find the (parameters of the) closest point on the manifold to the vector $I(a + n, b + m) \in \mathbf{R}^N$ where $(n, m) \in S$. If the closest manifold point is sufficiently near to $I(a + n, b + m)$, the feature is detected and the parameters of the closest manifold point are used as estimates of the feature parameters. On the other hand, if the nearest manifold point is too far away we declare the absence of the feature.

where $(x, y) \in \mathbf{R}^2$. After reconstructing the continuous surface $I^c(x, y)$ from the discrete image $I(n, m)$, feature detection could instead be accomplished by measuring how well $I^c(x, y)$ fits the continuous feature model $F^c(x, y; \mathbf{q}^c)$. Although all of what follows could be placed in such a continuous framework, we believe that carefully modeling the image formation process is vital for high performance, as is argued in [Nayar *et al.*, 96], and so work with discrete models and data.

In the model-fitting approach just described, the performance of feature detection is critically dependent upon the fitting function used to find the closest point on the feature manifold. In what follows we restrict attention to weighted L^2 norms as possible fitting functions. Three reasons serve to justify this restriction: (1) all previous detectors have used weighted L^2 norms, (2) the collection of weighted L^2 norms is a large class of possible fitting functions with a large variation in performance across the class (see Figure 1), and (3) every weighted L^2 norm can be specified by its corresponding weighting function thereby facilitating any optimization process.

2.2 Weighted L^2 Norms

Every measure μ on the set of pixels S defines a different L^2 norm and which is denoted $L^2(\mu)$ [Conway, 85] [Halmos, 74]. A measure is defined by the weight $\mu(n,m) = w_{nm} \geq 0$ that it assigns to each of the pixels $(n,m) \in S$. If $\mathbf{a} = (a_{nm}) \in \mathbf{R}^N$ is a vector of pixel intensity values, its $L^2(\mu)$ norm[2] is given by:

$$\|\mathbf{a}\|_\mu \;=\; \left[\sum_{(n,m)\in S} w_{nm} \cdot a_{nm}^2 \right]^{1/2} \tag{4}$$

The Euclidean L^2 norm is distinguished as the one for which the measure or weight of each pixel is 1. Closely related to the Euclidean L^2 norm is the sum of squared differences (SSD) which is identical to the square of the Euclidean L^2 norm.

Then, the distance between the point $F(n,m;\mathbf{q})$ on the feature manifold and the vector of image data $I(a+n, b+m)$ is given by:

$$\|F(n,m;\mathbf{q}) - I(a+n,b+m)\|_\mu \;=\; \left[\sum_{(n,m)\in S} w_{nm} \cdot [F(n,m;\mathbf{q}) - I(a+n,b+m)]^2 \right]^{1/2} \tag{5}$$

As described above, feature detection is based upon the distance to the closest manifold point, which is given by:

$$\min_{\mathbf{q}} \left[\sum_{(n,m)\in S} w_{nm} \cdot [F(n,m;\mathbf{q}) - I(a+n,b+m)]^2 \right]^{1/2} \tag{6}$$

It turns out that computing the square of an L^2 norm is easier than computing the L^2 norm itself, and since the square root function is monotonic, we base feature detection upon the

[2]For this to really be a norm, as opposed to a semi-norm, we strictly require $w_{nm} > 0$ for all $(n,m) \in S$ [Conway, 85]. Similarly, we also require $w_{nm} > 0$ for Equation (8) to define an inner product rather than a semi-inner product. These are both minor technical points which can be ignored since the feature detection problem does not require the definiteness property of a distance function to be posed in a meaningful way.

square of the distance to the manifold:

$$\min_{\mathbf{q}} \|F(n,m;\mathbf{q})-I(a+n,b+m)\|_{\mu}^{2} \;=\; \min_{\mathbf{q}} \sum_{(n,m)\in S} w_{nm}\cdot[F(n,m;\mathbf{q}) - I(a+n,b+m)]^{2} \quad (7)$$

When a feature is detected, its parameters are estimated using the parameter values that actually minimize the expression in Equation (7).

Another reason for restricting attention to weighted L^2 norms is that for each one there is an underlying Hilbert Space with inner product:

$$\langle \mathbf{a}, \mathbf{b}\rangle_{\mu} \;=\; \sum_{(n,m)\in S} w_{nm}\cdot a_{nm}\cdot b_{nm} \quad (8)$$

where $\mathbf{a} = (a_{nm})$ and $\mathbf{b} = (b_{nm})$ are vectors in $\mathbf{R}^N$. As we now describe, the Hilbert Space structure allows us to generalize the parameter normalization and dimension reduction procedures proposed in [Nayar *et al.*, 96]. As described in [Nayar *et al.*, 96], these two procedures are often needed to enhance the efficiency of detection algorithms.

2.3 Parameter Normalization

In [Nayar *et al.*, 96] an important normalization is applied. For each feature instance $F(n,m;\mathbf{q}) \in \mathbf{R}^N$ the mean coordinate $\mu(\mathbf{q}) = \frac{1}{N}\sum_{(n,m)\in S} F(n,m;\mathbf{q})$ and the coordinate variance $\nu(\mathbf{q}) = [\ \sum_{(n,m)\in S}[F(n,m;\mathbf{q}) - \mu(\mathbf{q})]^2\ ]^{1/2}$ are computed. Then, the feature instance is normalized:

$$\overline{F}(n,m;\mathbf{q}) \;=\; \frac{1}{\nu(\mathbf{q})}\,[F(n,m;\mathbf{q}) \;-\; \mu(\mathbf{q})] \quad (9)$$

For many features this simple normalization reduces the dimensionality of the feature manifold by two because $\overline{F}(n,m;\mathbf{q})$ turns out to be (approximately) independent of two of the (brightness) parameters in $\mathbf{q}$. Note that: (a) the normalization must be applied to both the training feature instances and to the image data $I(a+n,b+m)$, and (b) once a normalized feature has been detected, its mean μ and coordinate variance ν can be used to recover the two parameters eliminated during normalization. See [Baker *et al.*, 98] for a description of how this is performed.

If $\mathbf{c} = (1,1,\ldots,1) \in \mathbf{R}^N$ and $\hat{\mathbf{c}} = \mathbf{c}/\|\mathbf{c}\|$ then Equation (9) may be rewritten as:

$$\overline{F}(n,m;\mathbf{q}) \;=\; \frac{F(n,m;\mathbf{q}) \;-\; \langle F(n,m;\mathbf{q}), \hat{\mathbf{c}}\rangle\,\hat{\mathbf{c}}}{\|F(n,m;\mathbf{q}) \;-\; \langle F(n,m;\mathbf{q}), \hat{\mathbf{c}}\rangle\,\hat{\mathbf{c}}\|} \quad (10)$$

where $\|\cdot\|$ is the Euclidean L^2 norm and $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. In the weighted case, normalization is performed exactly as in Equation (10) except that the weighted inner product and the weighted L^2 norm are used in place of their Euclidean equivalents. Actually, it is feasible to still use the Euclidean norm and inner product during normalization even when using a weighted L^2 norm during feature detection. The pros and cons of doing so are outside the scope of this paper. We will use weighted norms and inner products since it simplifies the analysis (in Appendix A.2) of the effect which parameter normalization has on noise.

2.4 Dimension Reduction

It is important that weighted L^2 norms can be evaluated efficiently, particularly for detectors which attempt to evaluate the expression in Equation (7) numerically. By examining Equation (5), it is immediate that $4 \cdot N - 1$ arithmetic operations are needed to evaluate $\|\overline{F}(n,m;\mathbf{q}) - \overline{I}(a+n,b+m)\|_\mu^2$. In the Euclidean case, the computation of the square of the L^2 norm is an SSD which can be performed in $3 \cdot N - 1$ arithmetic operations.

Since every L^2 norm is derived from an underlying Hilbert space, we can apply dimension reduction techniques, such as the Karhunen-Loéve (K-L) expansion [Oja, 83], to further improve the efficiency. The K-L expansion is used in several model-fitting feature detectors, including [Hummel, 79], [Zucker and Hummel, 81], and [Nayar *et al.*, 96]. Ad-hoc dimension reduction is used in other detectors, such as [Hueckel, 71], [Hueckel, 73], and [Morgenthaler, 81].

Applying dimension reduction when using a weighted L^2 norm is straightforward, as we now show. If $\{\mathbf{e}^j \mid j = 1, 2, \ldots, N\}$ is an orthonormal basis (with respect to the underlying inner product) then:

$$\|\overline{F}(n,m;\mathbf{q}) - \overline{I}(a+n,b+m)\|_\mu^2 \;=\; \sum_{j=1}^{N} \left[\langle \overline{F}(n,m;\mathbf{q}), \mathbf{e}^j\rangle_\mu - \langle \overline{I}(a+n,b+m), \mathbf{e}^j\rangle_\mu\right]^2 \quad (11)$$

After a suitable change of basis vectors, 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 $\{\mathbf{e}^j \mid j = 1, 2, \ldots, d\}$, where d is the dimension of the low dimensional subspace. Hence, after applying dimension reduction we approximate:

$$\|\overline{F}(n,m;\mathbf{q}) - \overline{I}(a+n,b+m)\|_\mu^2 \;\approx\; \sum_{j=1}^{d} \left[\langle \overline{F}(n,m;\mathbf{q}), \mathbf{e}^j\rangle_\mu - \langle \overline{I}(a+n,b+m), \mathbf{e}^j\rangle_\mu\right]^2 \quad (12)$$

Since $\langle \mathbf{a}, \mathbf{e}^j\rangle_\mu$ is the j^{th} component of $\mathbf{a}$ in the low dimensional subspace, we see that the square of the weighted L^2 norm can be estimated with an SSD in the low dimensional subspace. The only requirement is that the basis vectors are orthonormal with respect to the underlying inner product. This is a requirement that can always easily be achieved by by orthonormalizing using Gram-Schmidt. Then, the computational cost of the SSD in the low dimensional subspace is $3 \cdot d - 1$. For most features d can be chosen small enough so that this is much less than $4 \cdot N - 1$ (and also $3 \cdot N - 1$.)

In the weighted case, the equivalent[3] of the Karhunen-Loéve expansion is computed as follows. The $N \times N$ weighted covariance matrix $\mathbf{C} = (C_{nm,pq})$ is given by:

$$C_{nm,pq} = \mathrm{E}_{\mathbf{q}} \left[\overline{G}(n, m; \mathbf{q}) \cdot w_{pq} \overline{G}(p, q; \mathbf{q}) \right] \tag{13}$$

where $\mathrm{E}[\cdot]$ is the expectation operator and $\overline{G}(n, m; \mathbf{q}) = \overline{F}(n, m; \mathbf{q}) - \mathrm{E}_{\mathbf{q}} \left[\overline{F}(n, m; \mathbf{q})\right]$. The d eigenvectors of $\mathbf{C}$ with the largest eigenvalues are used as a basis for the reduced dimensional space. Since $\mathbf{C}$ is self-adjoint with respect to the inner product (even though it is not symmetric), the eigenvectors will automatically be orthogonal (at least for distinct eigenvalues) [Conway, 85], but orthonormalization is advised since for many symmetric features duplicate eigenvalues are commonplace.

3 Optimal Weighting Functions

We begin this section by deriving our optimality criterion. Then, after describing a couple of simplifying assumptions, we finally derive optimal weighting functions assuming that the manifold is linear. We first consider a 1-parameter linear manifold and afterwards generalize to the k-parameter case. We end this section by provide experimental validation of the optimality of the Euclidean L^2 norm in the case that the noise is uniformly distributed across the pixels.

3.1 Optimality Criterion

Parameter estimation is inaccurate when the closest manifold point is not at the correct place on the manifold. Assume that there is an ideal feature instance with parameters $\mathbf{q}$ in the region of the world being imaged. Then, if this feature is projected onto a window surrounding the point (a, b) in the image I, the vector $I(a + n, b + m)$ should differ from $F(n, m; \mathbf{q})$ only because of noise introduced in the imaging process. If we model the noise with the additive random vector $\boldsymbol{\eta}$, we have:

$$I(a + n, b + m) \;=\; F(n, m; \mathbf{q}) + \eta(n, m; \mathbf{q}) \tag{14}$$

[3] The covariance matrix enters the unweighted minimum mean-square error formulation of the Karhunen-Loéve expansion in [Fukunaga, 90] as an instantiation of the linear operator $\mathbf{y} \to \mathrm{E}_{\mathbf{X}}[(\mathbf{X} - \mathrm{E}_{\mathbf{X}}[\mathbf{X}])\langle \mathbf{X} - \mathrm{E}_{\mathbf{X}}[\mathbf{X}], \mathbf{y}\rangle]$. In this, the inner product is the Euclidean inner product. When generalized to a minimum weighted L^2 norm error formulation the inner product must be changed to a weighted one, but otherwise the derivation remains the same and the optimal basis vectors are still the eigenvectors of this linear operator. The realization of $\mathbf{y} \to \mathrm{E}_{\mathbf{X}}[(\mathbf{X} - \mathrm{E}_{\mathbf{X}}[\mathbf{X}])\langle \mathbf{X} - \mathrm{E}_{\mathbf{X}}[\mathbf{X}], \mathbf{y}\rangle_\mu]$ as a matrix is the weighted covariance matrix given in Equation (13).

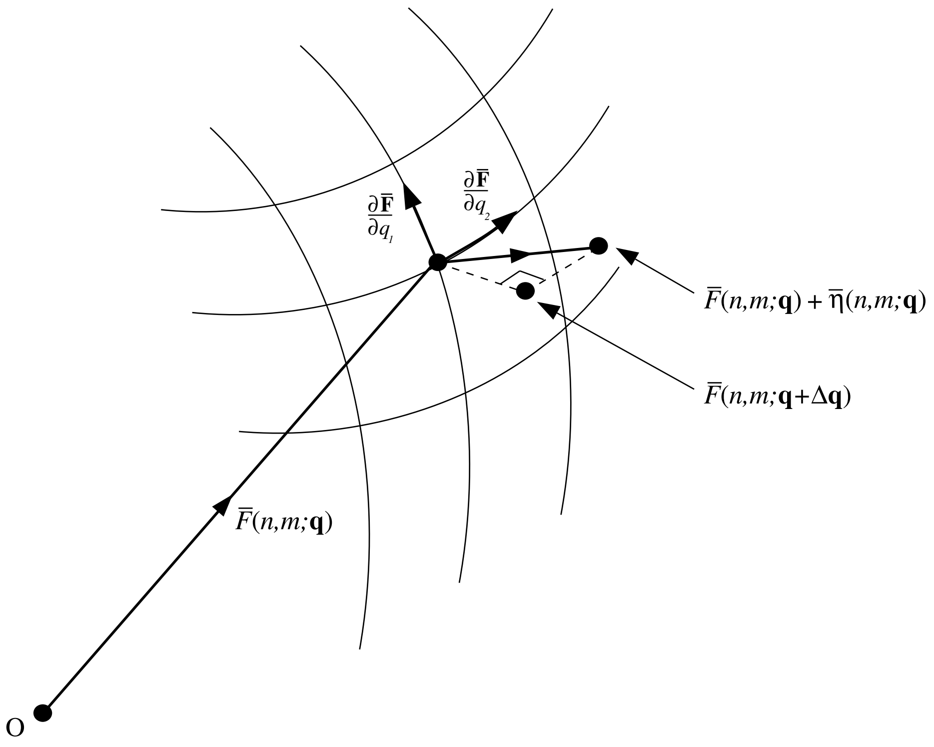


Figure 5: An illustration of a 2-parameter manifold used to motivate the optimality criterion. A normalized ideal feature instance $\overline{F}(n,m;\mathbf{q})$ is displaced by the noise $\overline{\eta}(n,m;\mathbf{q})$ to the point $\overline{F}(n,m;\mathbf{q})+\overline{\eta}(n,m;\mathbf{q})$. If the closest manifold point to this point is $\overline{F}(n,m;\mathbf{q}+\Delta\mathbf{q})$, the optimality criterion is a weighted average of the R.M.S. errors in the feature parameters $\Delta\mathbf{q}$.

During normalization the noise is modified somewhat (see Section 3.2.1 and Appendix A.2 for an description of exactly how). For now we denote the modified noise by $\overline{\eta}(n,m;\mathbf{q})$ and so the normalized image vector is given as follows:

$$\overline{I}(a+n,b+m) \;=\; \overline{F}(n,m;\mathbf{q})+\overline{\eta}(n,m;\mathbf{q}) \tag{15}$$

If the closest point on the manifold to $\overline{I}(a+n,b+m)$ is $\overline{F}(n,m;\mathbf{q}+\Delta\mathbf{q})$ then the error in estimating parameter q_i will be $\Delta q_i = \Delta q_i(\boldsymbol{\eta})$. See Figure 5 for an illustration of this situation.

To obtain a measure which is independent of the particular noise added (but of course not the distribution of the noise), we average over the noise distribution by taking the root mean squared error:

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

However, $\mathrm{RMS}_{\boldsymbol{\eta}}[\Delta q_i(\boldsymbol{\eta})]$ is still a function of the parameters $\mathbf{q}$ and so we allow the user to supply a function $\rho = \rho(\mathbf{q})$ to specify the importance of parameter estimation accuracy across the manifold. Finally, we propose the following as our measure of parameter estimation accuracy for parameter q_i:

$$\mathbf{PM}_i \;=\; \int \rho(\mathbf{q})\, \mathrm{RMS}_{\boldsymbol{\eta}}[\Delta q_i(\boldsymbol{\eta})]\, \mathrm{d}\mathbf{q} \tag{17}$$

The designer of a particular feature detector may wish to go even further and combine $\mathbf{PM}_i$ for $i = 1, 2, \ldots, k$ together with other measures of feature detection performance (for example estimates of the rates of occurrence of false positives and false negatives [Baker *et al.*, 98]) to yield a single overall performance metric. Analyzing such a measure analytically would probably be impossible and so optimization would have to be performed numerically. For this reason we study $\mathbf{PM}_i$ for fixed i in isolation.

3.2 Simplifying Assumptions

3.2.1 Noise and Normalization

In order to analyze the optimality criterion $\mathbf{PM}_i$ we need to make assumptions about the distribution of the noise $\boldsymbol{\eta}$ and how it is affected by normalization to produce the modified noise $\overline{\boldsymbol{\eta}}$. In Equation (3) of Section 2.1 we model the blur introduced by the optics and the averaging performed by the CCD. Unfortunately the operation of a CCD is a noisy process [Healey and Kondepudy, 91] and it is this which we model with the noise $\boldsymbol{\eta}$. We assume that, (a) $\eta(n, m; \mathbf{q})$ has zero mean, (b) it is pairwise independent across the pixels $(n, m) \in S$, and (c) it has variance $\sigma^2(n, m; \mathbf{q})$.

In Appendix A.2 we analyze the effect which the parameter normalization of Section 2.3 has on the noise. There we show that under weak assumptions about the feature model, and to a first order approximation, the modified noise $\overline{\eta}(n, m; \mathbf{q})$ also satisfies the first two properties and has variance:

$$\overline{\sigma}^2(n, m; \mathbf{q}) \;=\; \frac{\sigma^2(n, m; \mathbf{q})}{\sum_{(p,q)\in S} w_{pq}[F(p, q; \mathbf{q}) - (\sum_{(r,s)\in S} w_{rs})^{-1} \sum_{(r,s)\in S} w_{rs} \cdot F(r, s; \mathbf{q})]^2} \tag{18}$$

In Figure 6 we present the results of repeating the experiment of Figure 1 but vary the time at which noise is added. We compare adding noise of variance $\sigma^2(n, m; \mathbf{q})$ before normalization, with adding noise with the variance given in Equation (18) after normalization. As can be seen in the figure, parameter estimation performance is almost identical in these two scenarios and so is in agreement with the analysis of Appendix A.2.

3.2.2 Dimension Reduction

The other assumption we make is that while analyzing the optimality criteria we can ignore the effects of dimension reduction. In Figure 7 we present the results of an experiment designed to justify this assumption. We again repeat the experiment of Figure 1 but this

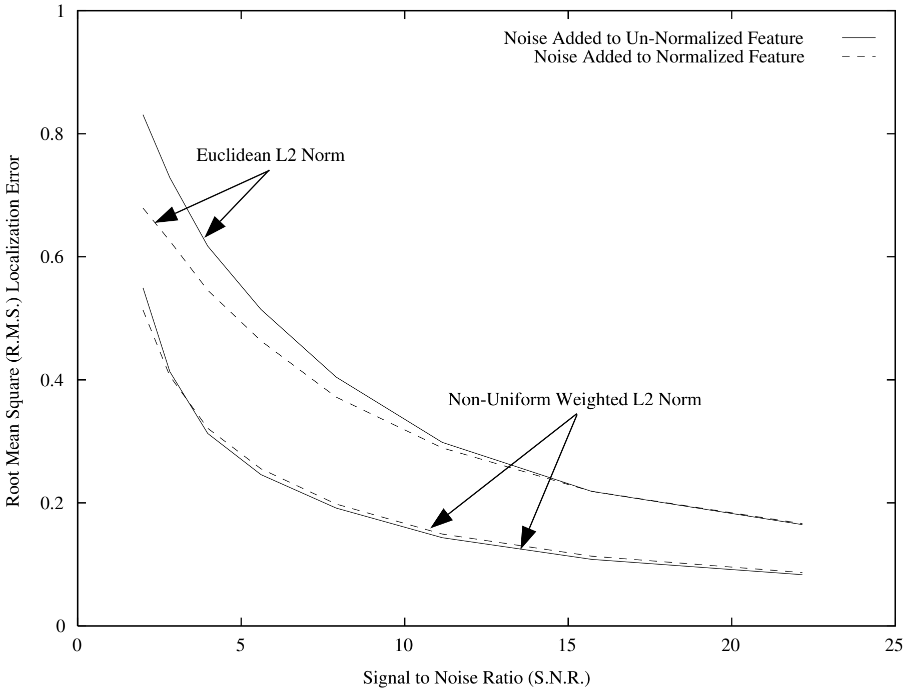


Figure 6: A comparison of noise models. We repeat the experiment of Figure 1 but vary the time at which noise is added. We compare adding noise before normalization with adding it afterwards for the two different L^2 norms: the Euclidean L^2 norm and the weighted L^2 norm using the weighting function of Figure 2. The variance of the noise added after normalization is calculated using Equation (18) where $\sigma^2(n, m; \mathbf{q})$ is the corresponding variance of the noise applied before normalization. The closeness of the corresponding graphs, especially as the S.N.R. increases, is in agreement with Equation (18) and the analysis of Appendix A.2.

time vary the dimension of the subspace in which detection is performed. We see from the figure that for very low dimensions (4 and below for the step edge), parameter estimation performance is definitely affected by the dimension of the space, but above a certain point (for the step edge at $d = 5$) the performance no longer improves with dimension. So long as we are above this point the performance using dimension reduction is approximately the same as what it would be for the entire space.

3.3 Optimal Weighting Functions for Linear Manifolds

To introduce the techniques involved in analyzing the optimality criterion $\mathbf{PM}_i$, we begin by considering the simple case of a 1-parameter linear manifold. Afterwards, we investigate the general case of a k-parameter linear manifold.

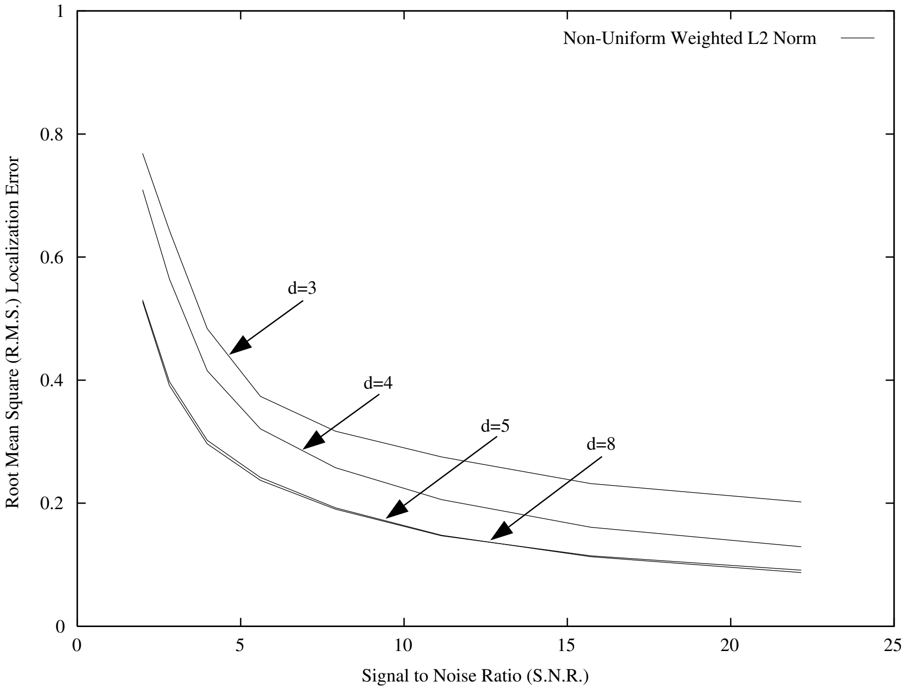


Figure 7: A comparison of parameter estimation performance as the dimension of the K-L subspace varies. We repeat the experiment of Figure 1 for the weighted L^2 norm using the weighting function of Figure 2. The results show that performance does increase with dimension, but only up to a point. Beyond a certain dimension ($d = 5$ for the step edge) the performance hardly improves. Hence, for all dimensions above $d = 5$, the performance is almost identical to what it would be without any dimension reduction.

3.3.1 1-Parameter Manifolds

If the manifold is linear or can be approximated as such, we perform a first-order Taylor expansion. When there is just one parameter q_1, this is given by:

$$\overline{\mathbf{F}}(q_1 + \Delta q_1) \;=\; \overline{\mathbf{F}}(q_1) \;+\; \Delta q_1 \frac{\mathrm{d}\overline{\mathbf{F}}}{\mathrm{d}q_1} \;+\; \mathrm{O}\left(\Delta q_1\right)^2 \tag{19}$$

The closest point on this 1-parameter linear manifold to the noisy point $\overline{\mathbf{F}}(q_1) + \overline{\boldsymbol{\eta}}(q_1)$ is the projection of the noisy point onto the manifold. Since the manifold is a line through $\overline{\mathbf{F}}(q_1)$ with direction $\frac{\mathrm{d}\overline{\mathbf{F}}}{\mathrm{d}q_1}$, the closest point is given by:

$$\overline{\mathbf{F}}(q_1) + \frac{1}{\left\| \frac{\mathrm{d}\overline{\mathbf{F}}}{\mathrm{d}q_1} \right\|_\mu^2} \left\langle \overline{\boldsymbol{\eta}}(q_1), \frac{\mathrm{d}\overline{\mathbf{F}}}{\mathrm{d}q_1} \right\rangle_\mu \frac{\mathrm{d}\overline{\mathbf{F}}}{\mathrm{d}q_1} \tag{20}$$

Equating the expressions in Equations (19) and (20) yields:

$$\Delta q_1 \;=\; \frac{\left\langle \overline{\boldsymbol{\eta}}(q_1), \frac{\mathrm{d}\overline{\mathbf{F}}}{\mathrm{d}q_1} \right\rangle_\mu}{\left\| \frac{\mathrm{d}\overline{\mathbf{F}}}{\mathrm{d}q_1} \right\|_\mu^2} \tag{21}$$

Then, using the properties of the noise $\overline{\boldsymbol{\eta}}(q_1)$ described in Section 3.2.1 it follows that:

$$\mathrm{RMS}_{\boldsymbol{\eta}}[\Delta q_1(\boldsymbol{\eta})] \;=\; \frac{\left[\sum_{(n,m)\in S} w_{nm}^2 \, \overline{\sigma}^2(n,m;q_1) \left(\frac{\mathrm{d}\overline{F}(n,m;q_1)}{\mathrm{d}q_1}\right)^2\right]^{1/2}}{\sum_{(n,m)\in S} w_{nm} \left(\frac{\mathrm{d}\overline{F}(n,m;q_1)}{\mathrm{d}q_1}\right)^2} \tag{22}$$

And so:

$$\mathbf{PM}_1 \;=\; \int \overline{\rho}(q_1) \frac{\left[\sum_{(n,m)\in S} w_{nm}^2 \, \sigma^2(n,m;q_1) \left(\frac{\mathrm{d}\overline{F}(n,m;q_1)}{\mathrm{d}q_1}\right)^2\right]^{1/2}}{\sum_{(n,m)\in S} w_{nm} \left(\frac{\mathrm{d}\overline{F}(n,m;q_1)}{\mathrm{d}q_1}\right)^2} \, \mathrm{d}q_1 \tag{23}$$

where:

$$\overline{\rho}(q_1) = \frac{\rho(q_1)}{\left[\sum_{(n,m)\in S} w_{nm} \left[F(n,m;q_1) - \frac{1}{\sum_{(r,s)\in S} w_{r,s}} \sum_{(r,s)\in S} w_{rs} \cdot F(r,s;q_1)\right]^2\right]^{1/2}} \tag{24}$$

We then have the following theorem:

Theorem 3.3.1 *If the variance of the noise $\sigma^2(n,m;q_1)$ is independent of the parameter q_1, the minimum value of $\mathbf{PM}_1$ is attained when:*

$$w_{nm} \;=\; \frac{1}{\sigma^2(n,m)} \tag{25}$$

And is given by:

$$\mathbf{PM}_1 \;=\; \int \overline{\rho}(q_1) \left[\sum_{(n,m)\in S} \left(\frac{1}{\sigma(n,m)} \frac{\mathrm{d}\overline{F}(n,m;q_1)}{\mathrm{d}q_1} \right)^2 \right]^{-1/2} \mathrm{d}q_1 \tag{26}$$

Proof Follows immediately from Lemma A.3.2 of Appendix A.3.1. □

Corollary 3.3.2 *If the variance of the noise is constant across the pixels, the Euclidean* L^2 *norm is optimal for 1-parameter linear manifolds.*

In the general case that $\sigma^2(n,m;q_1)$ is a function of q_1, there is no simple closed-form solution to the optimization problem and so $\mathbf{PM}_1$ must be optimized numerically.

3.3.2 k-Parameter Manifolds

Generalizing to a k-parameter manifold is not quite as straightforward as it appears at first glance, since the projection of the noise onto the hyperplane spanned by the first-order partial derivatives is complicated by the possibility that the partial derivatives might not be orthogonal. As in the 1-parameter case, we begin by performing a first-order Taylor expansion:

$$\overline{\mathbf{F}}(\mathbf{q}+\Delta\mathbf{q}) \;=\; \overline{\mathbf{F}}(\mathbf{q}) \;+\; \sum_{i=1}^{k} \Delta q_i \frac{\partial \overline{\mathbf{F}}}{\partial q_i} \;+\; \mathrm{O}\left(\|\Delta \mathbf{q}\|_\mu\right)^2 \tag{27}$$

To solve the problems posed by the potential lack of orthogonality, we consider the subspace spanned by all the partial derivatives except the i^{th} one:

$$\mathrm{D}_i \;=\; \bigoplus_{j\neq i} \frac{\partial \overline{\mathbf{F}}}{\partial q_j} \tag{28}$$

Then, if the projection of the i^{th} partial derivative into the subspace orthogonal to D_i is given by:

$$\mathbf{u}_i = \mathrm{P}_{\mathrm{D}_i^\perp}\left(\frac{\partial \overline{\mathbf{F}}}{\partial q_i}\right) \tag{29}$$

and $\hat{\mathbf{u}}_i = \mathbf{u}_i/\|\mathbf{u}_i\|_\mu$, it follows immediately that the projection of the noisy point $\overline{\mathbf{F}}(\mathbf{q})+\overline{\boldsymbol{\eta}}(\mathbf{q})$ onto the hyperplane spanned by the first-order partial derivatives is:

$$\overline{\mathbf{F}}(\mathbf{q}) \;+\; \langle \overline{\boldsymbol{\eta}}(\mathbf{q}), \hat{\mathbf{u}}_i\rangle_\mu \, \hat{\mathbf{u}}_i \;+\; \mathrm{P}_{\mathrm{D}_i}\left(\overline{\boldsymbol{\eta}}(\mathbf{q})\right) \tag{30}$$

where $\mathrm{P}_{\mathrm{D}_i}$ is the projection onto the subspace D_i. Equating the expressions in Equations (27) and (30), taking the inner product with $\mathbf{u}_i$, and simplifying gives:

$$\Delta q_i \;=\; \frac{\langle \overline{\boldsymbol{\eta}}(\mathbf{q}), \mathbf{u}_i\rangle_\mu}{\left\langle \frac{\partial \overline{\mathbf{F}}}{\partial q_i}, \mathbf{u}_i \right\rangle_\mu} \tag{31}$$

Then, following the same steps as in the 1-parameter case, we have:

$$\mathbf{PM}_i \;=\; \int \overline{\rho}(\mathbf{q}) \frac{\left[\sum_{(n,m)\in S} w_{nm}^2\, \sigma^2(n,m;\mathbf{q})\, u_i^2(n,m;\mathbf{q})\right]^{1/2}}{\sum_{(n,m)\in S} w_{nm}\, u_i(n,m;\mathbf{q})\, \frac{\partial \overline{F}(n,m;\mathbf{q})}{\partial q_i}}\, \mathrm{d}\mathbf{q} \tag{32}$$

where $\overline{\rho}(\mathbf{q})$ is exactly as given in Equation (24) except that q_1 is replaced with $\mathbf{q}$.

Although the expression for $\mathbf{PM}_i$ in Equation (32) appears very similar to that for $\mathbf{PM}_1$ in Equation (23), Lemma A.3.2 of Appendix A.3.1 is not applicable since $\mathbf{u}_i$ is a (very complex) function of w_{nm}. For this reason the optimization of $\mathbf{PM}_i$ is analytically much more complex than a simple generalization of the Cauchy-Schwarz inequality. The major source of difficulty is that the expansion of $\mathbf{u}_i$ in terms of the constant vectors $\frac{\partial \overline{\mathbf{F}}}{\partial q_i}$ using Gram-Schmidt has approximately $\frac{1}{2}k^2$ terms. Fortunately, in Lemma A.3.3 we are able to prove what we need just using properties of $\mathbf{u}_i$ and without performing a full Gram-Schmidt expansion. This allows us to prove Lemma A.3.4 of Appendix A.3.2 and hence:

Theorem 3.3.3 *If the variance of the noise $\sigma^2(n,m;\mathbf{q})$ is independent of the parameters $\mathbf{q}$ then:*

$$w_{nm} \;=\; \frac{1}{\sigma^2(n,m)} \tag{33}$$

is a stationary point of $\mathbf{PM}_i$.

Although this does not strictly prove that $w_{nm} = \frac{1}{\sigma^2(n,m)}$ minimizes $\mathbf{PM}_i$, it is a necessary condition, and as stated in [Hancock, 60] verification of the sufficient conditions for optimality is intractable for almost any non-trivial k-dimensional optimization problem. Because we have verified the necessary conditions and since Theorem 3.3.3 is a generalization of Theorem 3.3.1, it is very reasonable to assume the following conjecture holds:

Conjecture 3.3.4 *If the variance of the noise $\sigma^2(n,m;\mathbf{q})$ is independent of the parameters $\mathbf{q}$ then $\mathbf{PM}_i$ is minimized when:*

$$w_{nm} \;=\; \frac{1}{\sigma^2(n,m)} \tag{34}$$

and is given by:

$$\mathbf{PM}_i \;=\; \int \overline{p}(\mathbf{q}) \frac{\left[\sum_{(n,m)\in S} \frac{1}{\sigma^2(n,m)}\, u_i^2(n,m;\mathbf{q})\right]^{1/2}}{\sum_{(n,m)\in S} \frac{1}{\sigma^2(n,m)}\, u_i(n,m;\mathbf{q})\, \frac{\partial \overline{F}(n,m;\mathbf{q})}{\partial q_i}}\, \mathrm{d}\mathbf{q} \tag{35}$$

Assuming the conjecture holds, we then have the following:

Corollary 3.3.5 *If the variance of the noise is constant across the pixels, the Euclidean* L^2 *norm is optimal for all linear parametric manifolds.*

In the general case that $\sigma^2(n,m;\mathbf{q})$ is a function of $\mathbf{q}$, there is no simple closed-form solution and so $\mathbf{PM}_i$ must be optimized numerically.

3.3.3 Experiments

We conducted an experiment designed to validate the theoretical results of Sections 3.3.1 and 3.3.2 which imply that the Euclidean L^2 norm is optimal for noise with the same variance at each pixel. We constructed a synthetic 2-parameter linear manifold in $\mathbf{R}^5$

Table 1: The results of an experiment designed to validate the theoretical results of Sections 3.3.1 and 3.3.2. We display the R.M.S. parameter estimation errors for a linear 2-parameter manifold in $\mathbf{R}^5$ and for a number of different weighting functions. We used the uniform weighting function, a number of perturbations to it $w_n^1, \ldots, w_n^5$, and a large number of randomly generated weighting functions. We see that the uniform weighting function does perform the best thereby validating the theoretical results.

	Uniform	w_n^1	w_n^2	w_n^3	w_n^4	w_n^5	Min-Random	Max-Random
Parameter 1	12.53	12.64	12.72	12.65	12.66	12.55	12.60	18.06
Parameter 2	12.52	12.57	12.53	12.63	12.54	12.65	12.56	18.03

with $\frac{\partial \mathbf{F}}{\partial q_1} = (1.0, 2.0, -1.0, 3.0, 0.5)$ and $\frac{\partial \mathbf{F}}{\partial q_2} = (-0.5, 1.0, 1.0, 2.0, -3.5)$. We then applied the experimental procedure of Appendix A.1 for a number of different weighting functions. First we tested the uniform weighting function. Next, we considered small perturbations to the uniform weighting function. In particular we tried the 5 weighting functions $w_n^1, \ldots, w_n^5$ where $n = 1, \ldots, 5$ and:

$$w_j^i = \begin{cases} 0.4 & \text{if } i \neq j \\ 0.6 & \text{if } i = j \end{cases} \tag{36}$$

Finally, we used randomly generated weighting functions from the surface of the unit ball $\sum_{i=1}^5 w_i^2 = 1$. The way to generate points uniformly from a unit ball is to pick the coordinates independently from the Gaussian distribution and then normalize by dividing each coordinate by the Euclidean L^2 norm of the weight vector [Knuth, 81].

In Table 1 we present the results of the experiment. For each of the two manifold parameters, we give the R.M.S. error in its estimate for each of the weighting functions. For the random weighting functions we just present the best-ever and the worst-ever performances. As can be seen the Euclidean L^2 norm does perform the best, validating the theoretical results of Sections 3.3.1 and 3.3.2.

4 Empirical Comparison of Weighting Functions

In this section we present an empirical comparison of a number of plausible weighting functions for 3 ubiquitous features: the step edge, the symmetric line, and the corner. The parametric model of the step edge is presented in Section 2.1. The corresponding models for the line and the corner are the same as those used in [Baker *et al.*, 98]. The reader is referred to that paper for a full description of the models. We begin this section by briefly enumerating the weighting functions which we studied. Afterwards we describe the experimental results for each of the 3 features in turn.

4.1 Weighting Functions

In Figure 8 we illustrate the weighting functions which we used in the empirical comparison. In this figure, the weighting functions are displayed in an 81 pixel disc, which is the window which was used for the experiments on the corner and the line. For the step edge the experiments were conducted on a 49 pixel disc, and so for those experiments the weighting functions were scaled appropriately to the smaller window size. Except where stated otherwise, we describe the weighting functions in units that correspond to the window having unit radius, and the functions are expressed in terms of (x, y) coordinates relative to the center of the window. Finally, the weighting functions are scaled so that the most heavily weighted pixel is displayed at the maximum intensity (*ie.* pure white.)

In Figures 8(a) and (b) we show the Gaussian weighting function:

$$w(x, y) \ = \ e^{\frac{x^2+y^2}{2\sigma^2}} \tag{37}$$

suggested by Hueckel [Hueckel, 73] and used by Hartley in [Hartley, 85]. In Figure 8(a) the value of σ for the Gaussian is 1.0 pixels and in Figure 8(b) it is 2.0 pixels. Next, in Figure 8(c) we show the weighting function:

$$w(x, y) \ = \ [1 - (x^2 + y^2)]^{1/2} \tag{38}$$

which was actually used by Hueckel in [Hueckel, 71] and [Hueckel, 73].

The weighting function:

$$w(x, y) \ = \ \frac{1}{(x^2 + y^2)^{1/2}} \tag{39}$$

which was briefly studied by Lenz in [Lenz, 87] has a discontinuity at the origin, and so we modify it slightly. Instead, we used the weighting function:

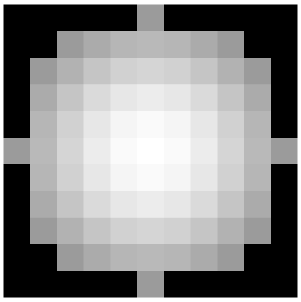
$$w(x, y) \ = \ \frac{1}{(k + x^2 + y^2)^{1/2}} \tag{40}$$

for various values of k. This function for $k = 1.0$ is presented in Figure 8(d) and for $k = 0.16$ in Figure 8(e). In Figures 8(f)–(h) we present three weighting functions that take their maximum value midway between the center of the window and its periphery. In Figure 8(f) the weighting function is the "Square Root Ramp" weighting function:

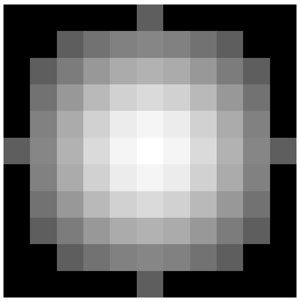
$$w(x, y) \ = \ \left(1 - \left|0.5 - \sqrt{x^2 + y^2}\right|\right)^{1/2} \tag{41}$$

The other two such weighting functions are "Modified Lenz" weighting functions. These functions decay just like the Lenz weighting function, but the maximum value is attained midway between the center of the window and its periphery:

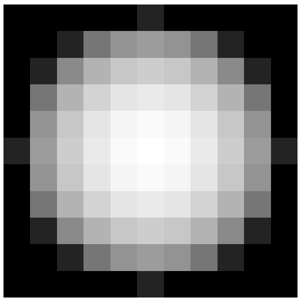
$$w(x, y) \ = \ \frac{1}{(k + (0.5 - \sqrt{x^2 + y^2})^2)^{1/2}} \tag{42}$$



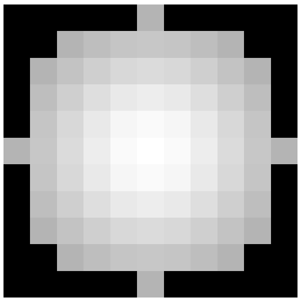
(a) Gaussian, $\sigma = 1$



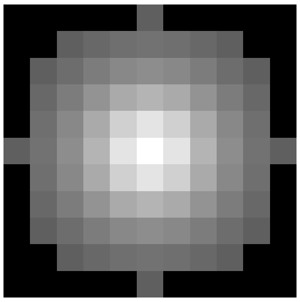
(b) Gaussian, $\sigma = 2$



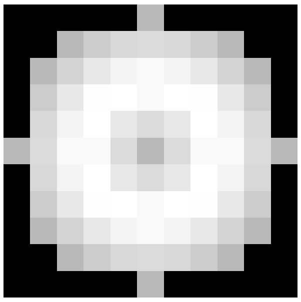
(c) Hueckel



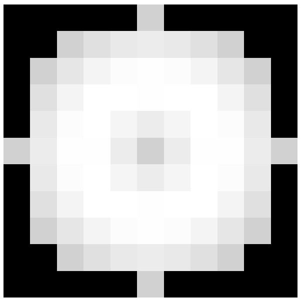
(d) Lenz, $k = 1.0$



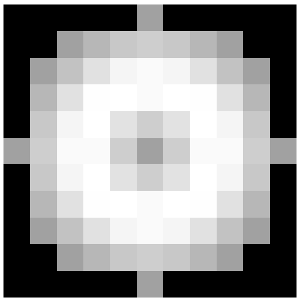
(e) Lenz, $k = 0.16$



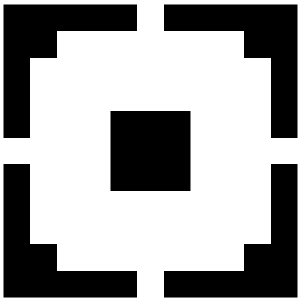
(f) Square Root Ramp



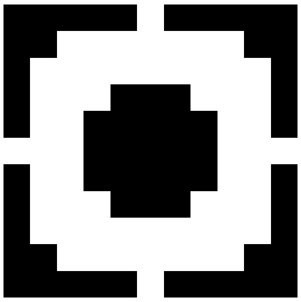
(g) Mod. Lenz, $k = 0.5$



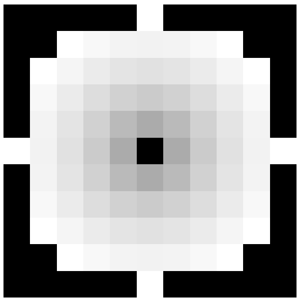
(h) Mod. Lenz, $k = 0.25$



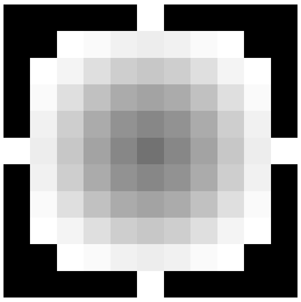
(i) Paton, $r = 0.4$



(j) Paton, $r = 0.57$



(k) Fourth Root



(l) Refl. Lenz, $k = 0.25$

Figure 8: The weighting functions used in the empirical study displayed in an 81 pixel disc. (See the main text for a complete description.)

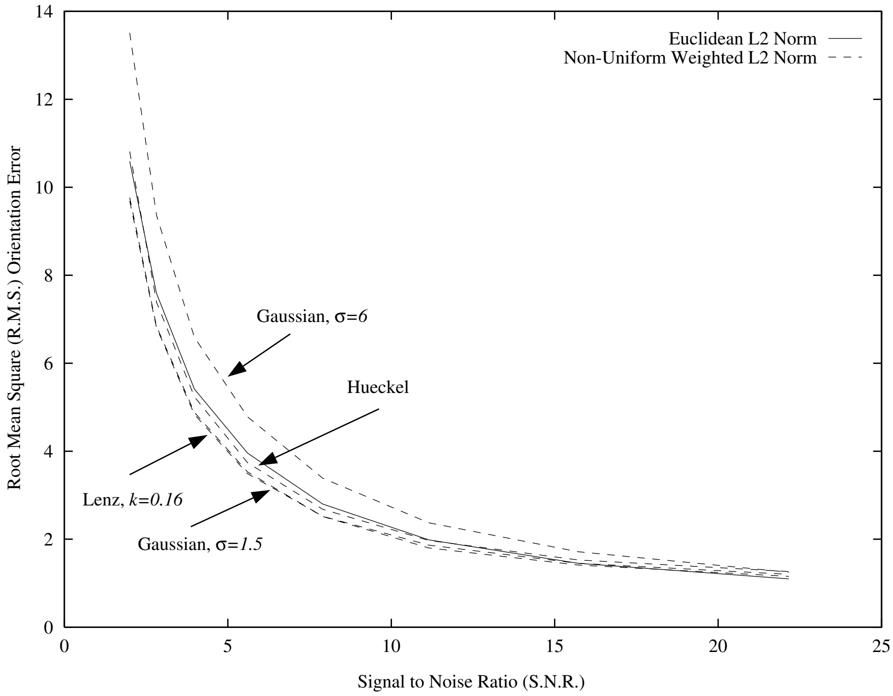


Figure 9: The estimation accuracy of the orientation parameter θ for the step edge. For high values of σ, the Gaussian weighting function performs worse than the Euclidean L^2 norm, but for low values of σ, the Gaussian performs better. The Hueckel and Lenz (for $k = 0.16$) weighting functions both perform marginally better than the Euclidean L^2 norm. The Lenz weighting function and the Gaussian (for $\sigma = 1.5$) perform the best in terms of orientation estimation accuracy.

In Figure 8(g) we plot this function for $k = 0.5$, and in Figure 8(h) for $k = 0.25$.

Next, in Figures 8(i) and (j) we display the annular stop function suggested by Paton in [Paton, 75] for radius values of 0.4 and 0.57. Since this function has a step discontinuity, we also considered smooth weighting functions which increase with the distance from the center of the window. In particular, we considered the "Fourth Root" weighting function:

$$w(x, y) = \sqrt[4]{x^2 + y^2} \tag{43}$$

which is displayed in Figure 8(k), and the "Reflected Lenz" weighting function which is decays just like the Lenz weighting function, but which has the maximum value at the edge of the window:

$$w(x, y) = \frac{1}{(k + (1.0 - \sqrt{x^2 + y^2})^2)^{1/2}} \tag{44}$$

The Reflected Lenz weighting function is displayed in Figure 8(l) for $k = 0.25$.

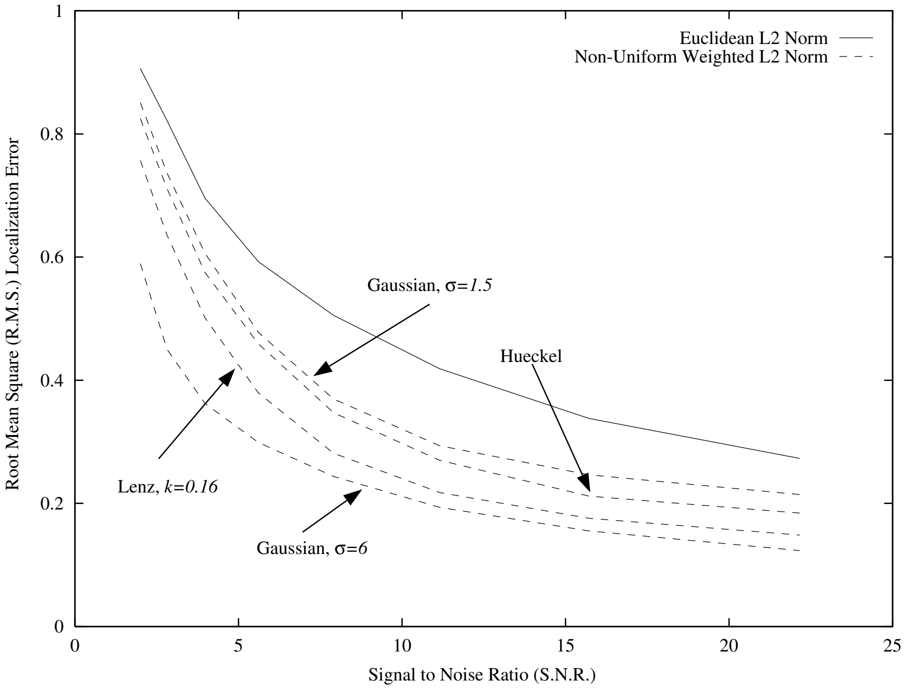


Figure 10: The estimation accuracy of the localization parameter ρ for the step edge. There is a large performance difference for the localization parameter with the worst performer having approximately twice the error which the best performer has. The best performer is the Gaussian with a high value of σ, next is the Lenz (with $k = 0.16$), then the Hueckel and low σ Gaussian, and finally the worst performer is the unweighted Euclidean L^2 norm.

4.2 Step Edge

The step edge model of Section 2.1 has 5 parameters: the orientation θ, the localization ρ, the blur σ, the lower intensity level A, and the intensity step B. We studied the estimation accuracy of each of these parameters for the Euclidean L^2 norm, the Gaussian weighting function (for various settings of σ), Hueckel's weighting function, and the Lenz weighting function (for various settings of k.) The results for the parameters θ, ρ, σ, and A are presented in Figures 9–12. The results for parameter B are similar to those for parameter A and are omitted.

In Figure 9, we present the results for the orientation parameter θ. All the weighting functions perform almost equally well, but the Gaussian weighting function (for $\sigma = 1.5$) and the Lenz weighting function (for $k = 0.16$) perform the best overall. The Hueckel weighting function also marginally outperforms the Euclidean L^2 norm. For large values of σ (over around $\sigma = 3.0$), the Gaussian weighting function performs worse than the Euclidean L^2 norm.

Next, in Figure 10 we present the results for the localization parameter. Here, we see

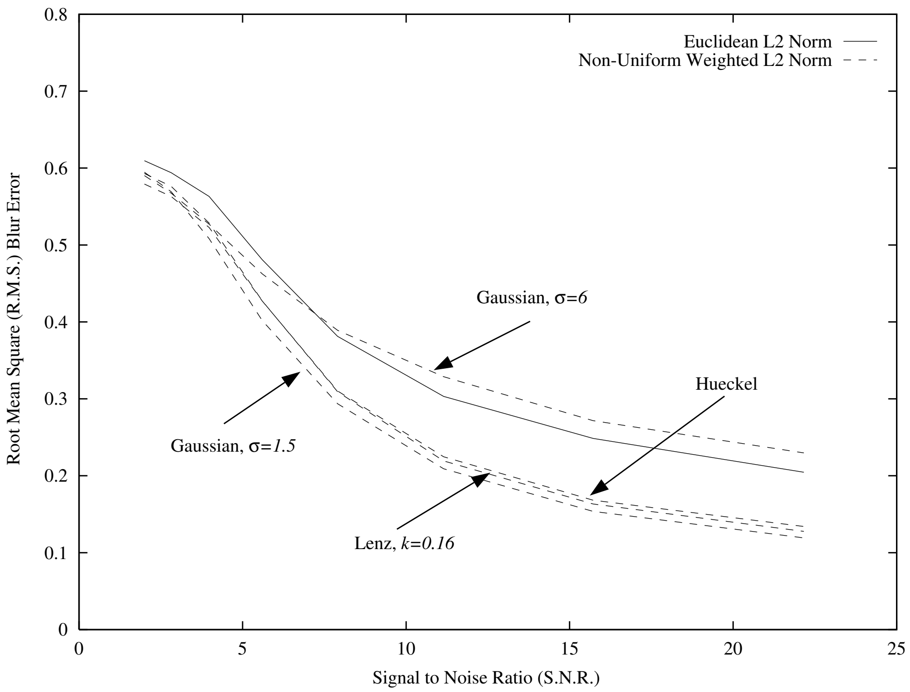


Figure 11: The estimation accuracy of the blur parameter σ for the step edge. Like for the localization parameter, there is a big difference in performance across the weighting functions. However, here the performance variation does depend upon the level of noise. For high noise levels, all weighting functions perform equally badly, and almost as badly as they would do should they just guess the value of the blur randomly. This indicates that the blur parameter is very hard to estimate for high noise levels. For lower noise levels, the Lenz (for $k = 0.16$), Hueckel, and low σ Gaussian weighting functions all perform almost equally well, and much better than the Euclidean L^2 norm and the high σ Gaussian.

that there is a wide range of performance levels across the weighting functions. The best performer has approximately half the error that the worse one does. All the non-uniformly weighted L^2 norms do far better than the Euclidean L^2 norm. The best performer is the Gaussian weighting function with $\sigma = 6.0$. As σ is decreased, the performance slowly reduces, but for all values of σ considered, the Gaussian outperformed the Euclidean L^2 norm. After the high σ Gaussian, the next best performer is the modified Lenz weighting function (with $k = 0.16$), and after that the Hueckel weighting function.

In Figure 11, we present the results for the blur parameter. For high noise levels almost all the weighting functions perform equally well. On further inspection, we see that they perform about as well as an approach which randomly guesses the blur parameter would do. This indicates that for high noise levels, the effect of the noise and the effect of the blur parameter on the appearance of the feature are comparable, and so estimating the blur parameter accurately is essentially impossible. For lower noise levels, the range of

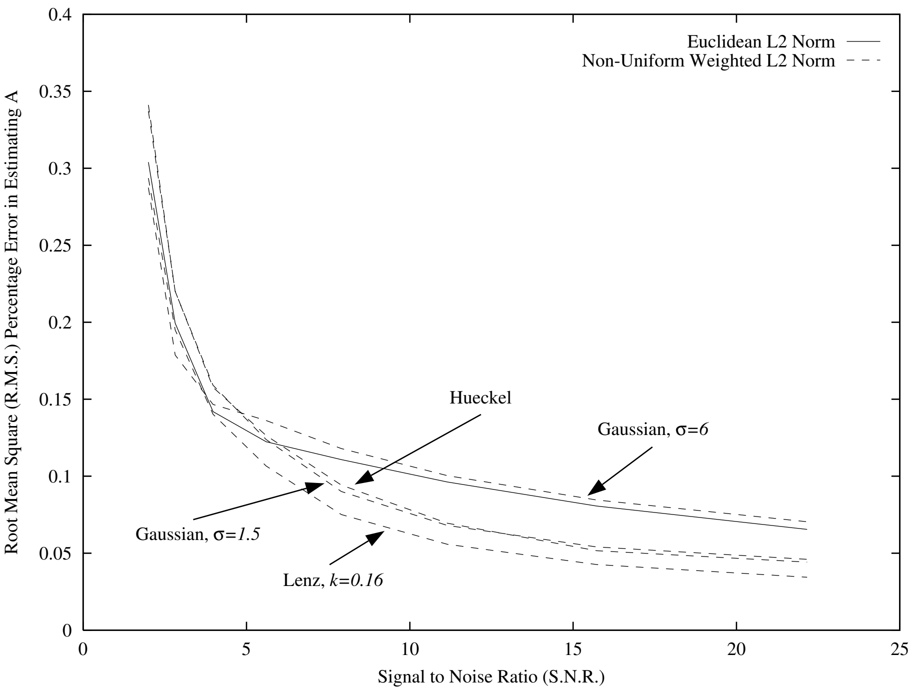


Figure 12: The estimation accuracy of the lower intensity level parameter A for the step edge. The relative performance of the weighting function for the lower intensity level parameter A is very similar to that for the blur parameter. For high noise levels, all weighting functions perform similarly. For lower noise levels the Lenz, Hueckel, and low σ Gaussian all perform comparably, and considerably better than the Euclidean L^2 norm and the high σ Gaussian.

performance across weighting functions is similar to what it is for the localization parameter (approximately a factor of 2.) The Lenz, low σ Gaussian, and Hueckel detectors perform almost equally well, and approximately twice as well as the Euclidean L^2 norm and the high σ Gaussian.

Finally, we present in Figure 12 the results for the lower intensity level A. The results for the intensity step B are very similar and so omitted. Since the parameter A is normalized, it is not explicitly represented on the manifold, and so is somewhat different. The main cause of errors in estimating the normalized parameters A and B is inaccurate estimates of the other parameters, particularly σ and ρ but to a lesser extent θ also. This is because the other 3 parameters are inputs to the procedure to invert the normalization. (See [Baker *et al.*, 98] for more details.) In the figure, we see that the results are similar to those for the localization parameter and the blur parameter. For high noise levels all weighting functions perform similarly. Presumably, the performance is limited by the estimate of the blur parameter. For lower noise levels, the modified Lenz weighting function performs the best, with the low σ Gaussian and Hueckel weighting functions slightly worse, and the Euclidean L^2 norm and high σ Gaussian substantially worse still.

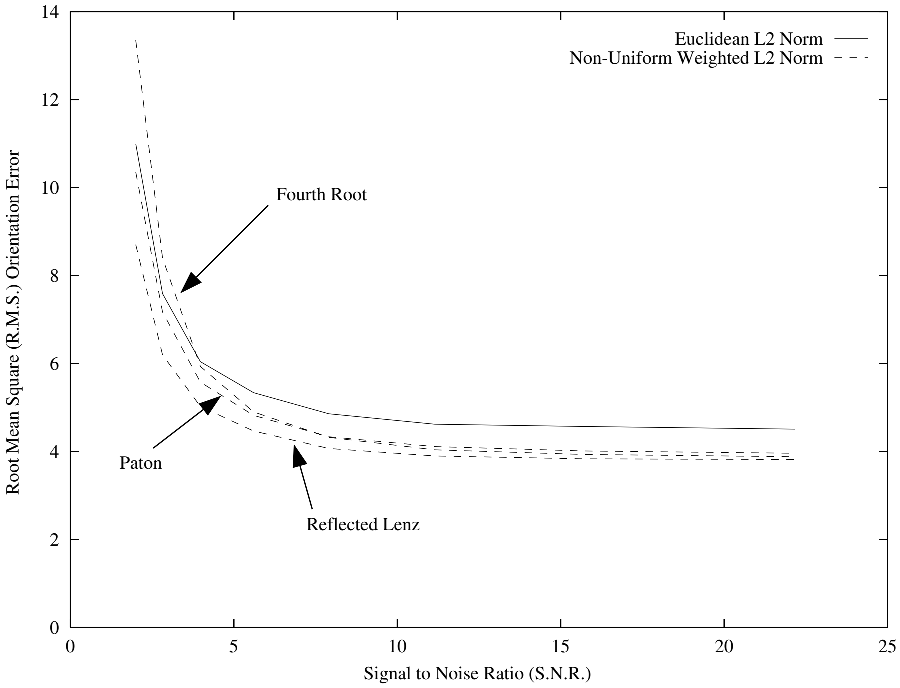


Figure 13: The estimation accuracy of the orientation parameter θ_1 for the corner. For low noise levels all three of the non-uniform weighted L^2 norms marginally outperform the Euclidean L^2 norm. The Reflected Lenz (for $k = 0.5$) does the best, but the Paton (for $r = 0.57$) and Fourth Root weighting functions are close behind. For high noise levels, the Paton and Lenz weighting functions still do better than the Euclidean L^2 norm, but the Fourth Root does slightly worse.

Overall, from Figures 9–12 it can be seen that the Lenz weighting function (for $k = 0.16$) performs the best. It performs marginally worse than the high σ Gaussian for the localization parameter ρ, but otherwise it is always about the best, and the high σ Gaussian performs poorly for all of the other parameters. The Euclidean L^2 norm performs relatively poorly for all the parameters with the exception of the orientation parameter θ.

4.3 Corner

The Corner model (see [Baker *et al.*, 98]) has 5 parameters: the orientation of the corner θ_1, the angle subtended by the corner θ_2, the blur σ, the lower intensity level A, and the intensity step B. We investigated the estimation accuracy of these parameters for the Euclidean L^2 norm, Paton's annular stop function (for various values of r), the Reflected Lenz weighting function (for various values of k), and the Fourth Root weighting function. The results for the parameters θ_1, θ_2, σ, and B are presented in Figures 13–16. The results for parameter A are similar to those for parameter B

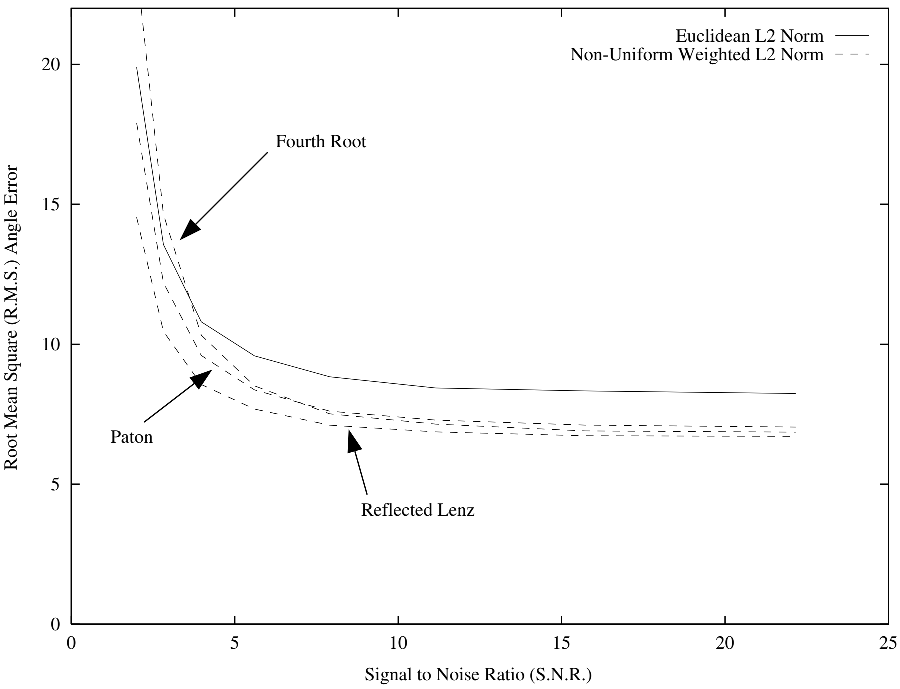


Figure 14: The estimation accuracy of the angle subtended by the corner θ_2. The results are qualitatively the same as those for the orientation parameter θ_1, but this is to be expected since there is a close relationship between the two angle parameters. All three of the non-uniform weighted L^2 norms outperform the Euclidean L^2 norm. For low noise levels the Reflected Lenz (for $k = 0.5$) does the best, but the Paton (for $r = 0.57$) and Fourth Root weighting functions are close behind. For high noise levels, the Paton and Lenz weighting functions still do better than the Euclidean L^2 norm, but the Fourth Root does slightly worse.

In Figure 13 we present the results for the orientation parameter θ_1. As can be seen, the non-uniform weighted L^2 norms perform slightly better than the Euclidean L^2 norm. The best performer is the Reflected Lenz weighting function, followed by Paton's annular stop function. The Fourth Root weighting function does slightly worse for high noise levels. Next, in Figure 14 we present the results for the angle subtended by the corner. The results are qualitatively very similar, as is the relative performance of the weighting functions The major difference is that the estimation accuracy of this angle is almost a factor of 2 worse for all of the weighting functions. This is to be expected since the orientation parameter can be regarded as an average of the estimates of the two edges forming the corner, and the angle subtended by the corner is the difference of these two estimates.

The results for the blur parameter of the corner in Figure 15 also show that the non-uniform weighting functions do significantly better than the Euclidean L^2 norm. The Reflected Lenz weighting function does the best, followed by the Fourth Root, and then the Paton annular stop. The final two parameters for the corner are the two intensity

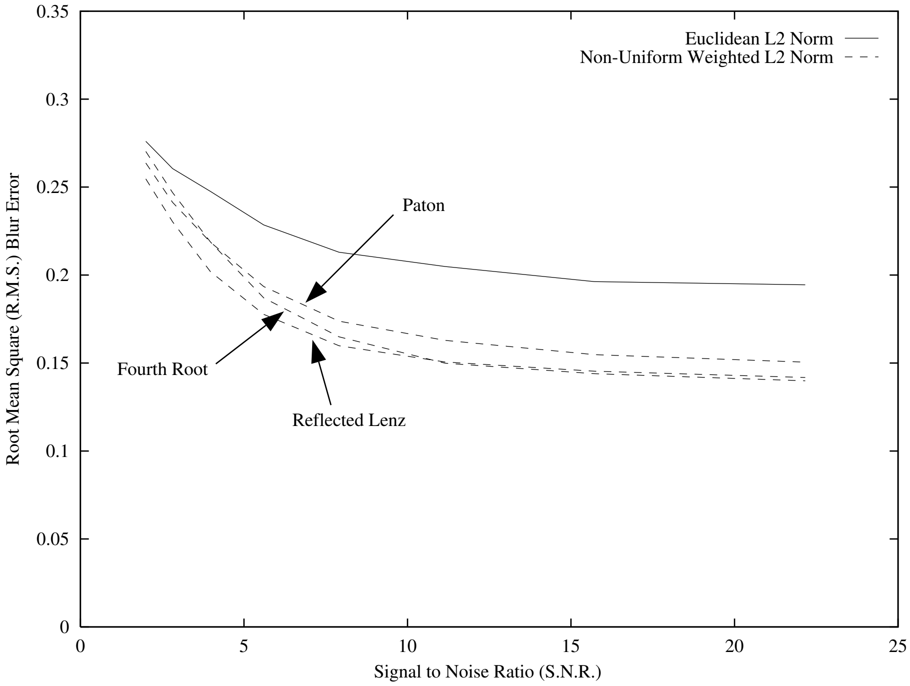


Figure 15: The estimation accuracy of the blur parameter σ for the corner. Again, all three non-uniform weighted L^2 norms do significantly better than the Euclidean L^2 norm. The Reflected Lenz weighting function does the best, followed by the Fourth Root, and then the Paton annular stop, and finally the Euclidean L^2 norm.

parameters A and B. For the base level A all weighting functions perform pretty well, and there is not much difference between them, so we omit the results. The results for B (see Figure 16) are noticeably worse than for A (presumably because the upper intensity level of the corner occupies a much smaller areas of the window on average.) For B, the non-uniform weighting functions again do substantially better than the Euclidean L^2 norm, with the Reflected Lenz and Fourth Root weighting functions doing marginally better than Paton's annular stop.

Overall, from Figures 13–16 it can be seen that weighting the outside pixels more than the center ones improves the accuracy of parameter estimation performance for the corner. This is consistent with intuition. The best weighting function overall which we found was the Reflected Lenz weighting function with $k = 0.5$. The Paton annular stop does slightly better than the Fourth root for the two angle parameters, and slightly worse for the other parameters.

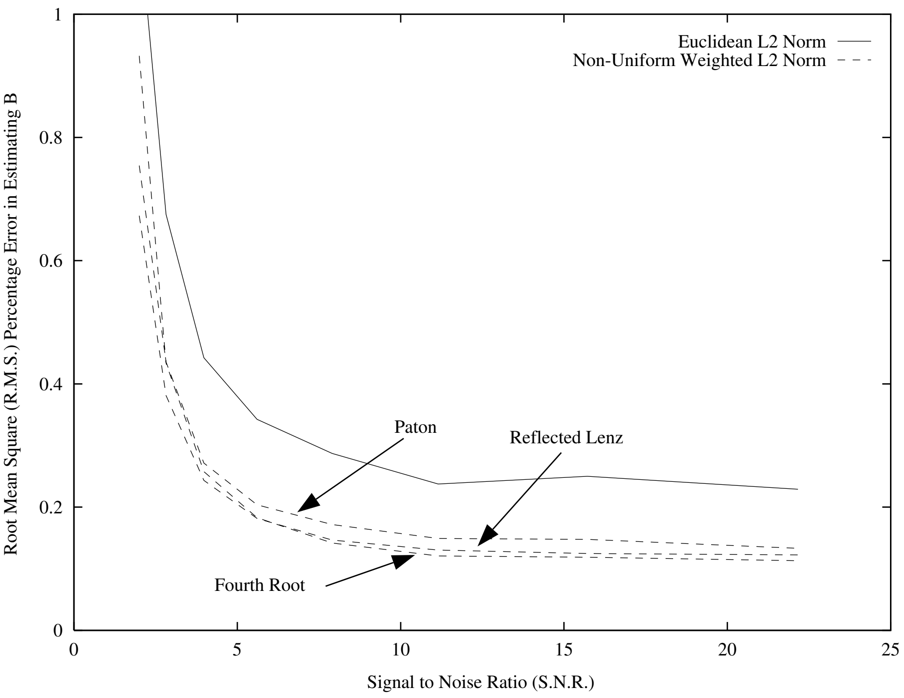


Figure 16: The estimation accuracy of the intensity step parameter B for the corner. Again, all three non-uniform weighted L^2 norms perform significantly better than the Euclidean L^2 norm. The Reflected Lenz and Fourth Root weighting functions do a little bit better than Paton's annular stop.

4.4 Line

The line model (see [Baker *et al.*, 98]) has 6 parameters: the orientation of the line θ, the localization of the line ρ, the blur parameter σ, the width of the line w, the base intensity A, and the intensity step B. We studied the estimation accuracy of each of these parameters for the Euclidean L^2 norm, the Gaussian weighting function (for various values of σ), the Modified Lenz weighting function (for various settings of k), and the Square Root Ramp function. The results for parameters θ, ρ, σ, and w are presented in Figures 17–20. The results for the two intensity parameters A and B are consistent with the performance for the other parameters and are omitted.

The results for the orientation parameter θ of the line are presented in Figure 17. It can be seen that the Euclidean L^2 norm and the Modified Lenz weighting functions are the best performers and do almost exactly the same. The Square Root Ramp function does ever so slightly worse. The Gaussian weighted L^2 norm does noticeably worse, particularly for high noise levels. The results for the localization parameter in Figure 18 are very similar. Again, the Euclidean L^2 norm, the Modified Lenz weighting function, and the Square Root Ramp function do the best, with the Gaussian weighted L^2 norm doing significantly worse.

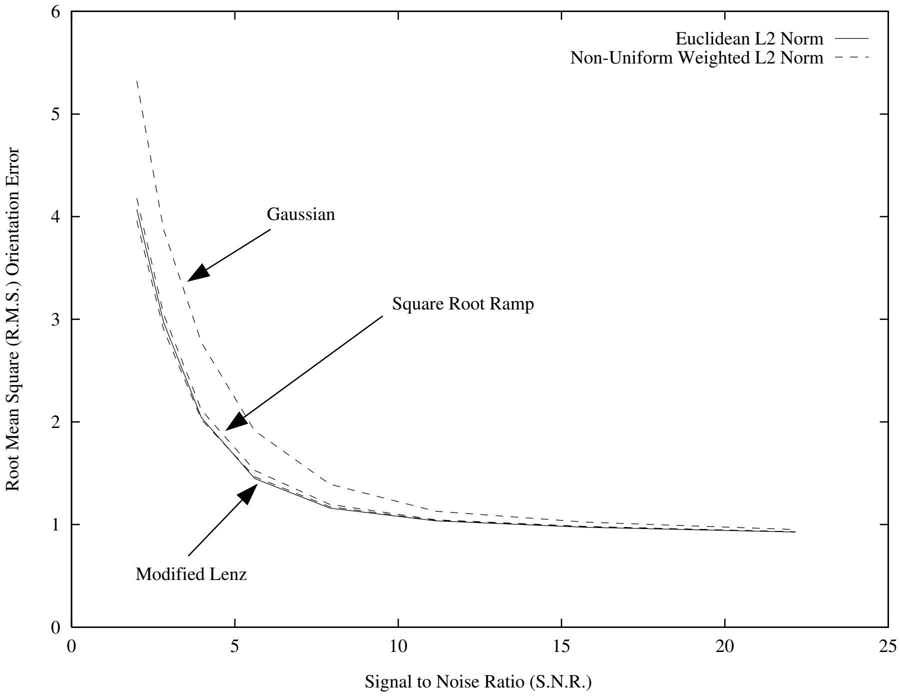


Figure 17: The estimation accuracy of the orientation parameter θ for the line. The Euclidean L^2 norm and the Modified Lenz weighting functions are the best performers and do almost exactly the same. The Square Root ramp function does ever so slightly worse. The Gaussian weighted L^2 norm does substantially worse for high noise levels. Comparing the performance to that of the step edge, we see that estimating the orientation of the line is a lot easier than it is for the step edge, presumably because it can be regarded as an average of the estimates of the two step edges which make up the line.

For both the orientation and the localization, the results for the line are, in general, much better than for the step edge. This is presumably because the estimates of these two parameters can be regarded as the average of the estimates for the same parameters of the two step edges which go to form the line. On the other hand, the result for the width are not as good, as can be seen in Figure 19. (All other things being equal it should be expected that the width of the line is harder to estimate than the localization of the step edge because the range of values it can take is larger.) As seen in Figure 19, the best performer for the width is the Gaussian weighting function. The Square Root Ramp function and the Modified Lenz function do slightly better than the Euclidean L^2 norm for low noise levels, and slightly worse for high noise levels. Finally, in Figure 20 we present the results for the blur parameter σ of the line. The best performer is again the Gaussian weighting function, followed by the Square Root Ramp function, the Modified Lenz function, and the Euclidean L^2 norm in that order.

Overall, the best performers for the line are the two weighting functions which give

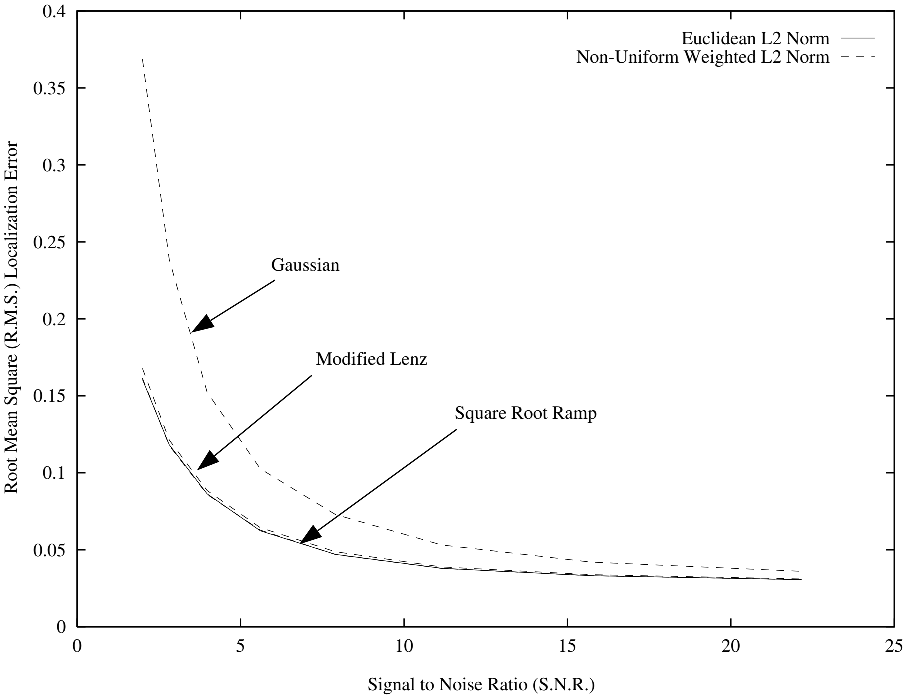


Figure 18: The estimation accuracy of the localization parameter ρ of the line. For high values of σ, the Gaussian weighting function performs worse than the Euclidean L^2 norm, but for low values of σ, the Gaussian performs better. The Hueckel and Lenz (for $k = 0.16$) weighting functions both perform marginally better than the Euclidean L^2 norm. The Lenz weighting function and the Gaussian (for $\sigma = 1.5$) perform the best in terms of orientation estimation accuracy.

the most weight to the pixels midway between the center and the edge of the window; *ie.* the Modified Lenz function and the Square Root Ramp function. This is intuitively reasonable because this is where the edges which go to form the line are approximately located. However, the difference between these weighting functions and the Euclidean L^2 norm is much smaller than the difference for the step edge and the corner. In fact, for the orientation and localization parameters, there is essentially no difference in performance between the Euclidean L^2 norm and the weighting functions which give the most weight to the pixels midway between the center and the edge of the window.

5 Discussion

5.1 Relationship with Optimal Filtering

We now present a brief discussion of the relationship between our work and the optimal filtering approach to edge detection best exemplified by [Canny, 86], but also studied

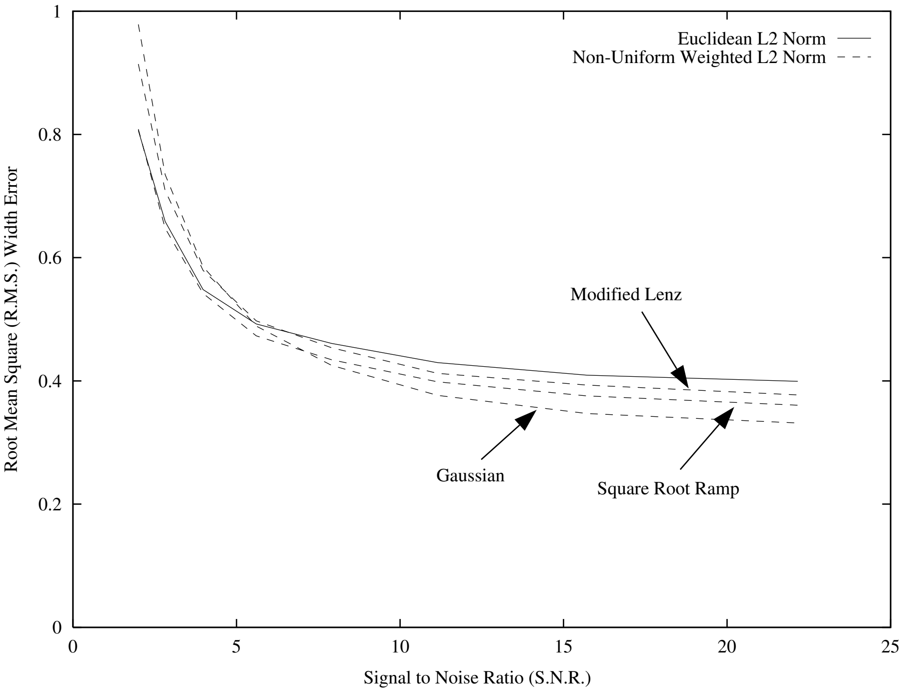


Figure 19: The estimation accuracy of the width parameter w for the line. The best performer is the Gaussian weighting function. The Square Root Ramp function and the Modified Lenz function do slightly better than the Euclidean L^2 norm for low noise levels, and slightly worse for high noise levels.

in [Modestino and Fries, 77], [Shanmugam *et al.*, 79], [Boie *et al.*, 86], [Deriche, 87], and [Sarkar and Boyer, 91]. In doing so, we will attempt to characterize the most important similarities and differences between the two major approaches to feature detection as categorized in [Nalwa, 93]: parametric model matching and using differential invariants. We note that there has already been some previous work attempting to unify different approaches to edge detection including [Abramatic, 81] and [Rosenfeld, 81].

Taking Equation (7) and replacing unnormalized vectors with their normalized equivalents, we see that the model fitting approach to feature detection is based upon the value of:

$$\min_{\mathbf{q}} \sum_{(n,m)\in S} w_{nm} \cdot \left[\overline{F}(n,m;\mathbf{q}) - \overline{I}(a+n,b+m)\right]^2 \tag{45}$$

After multiplying out the expression in the square brackets Equation (45) simplifies to:

$$2 \;-\; 2 \cdot \max_{\mathbf{q}} \sum_{(n,m)\in S} w_{nm} \cdot \overline{F}(n,m;\mathbf{q}) \cdot \overline{I}(a+n,b+m) \tag{46}$$

In this paper we studied the selection of the weighting function w_{nm} to optimize one aspect of feature detection performance, namely, parameter estimation accuracy.

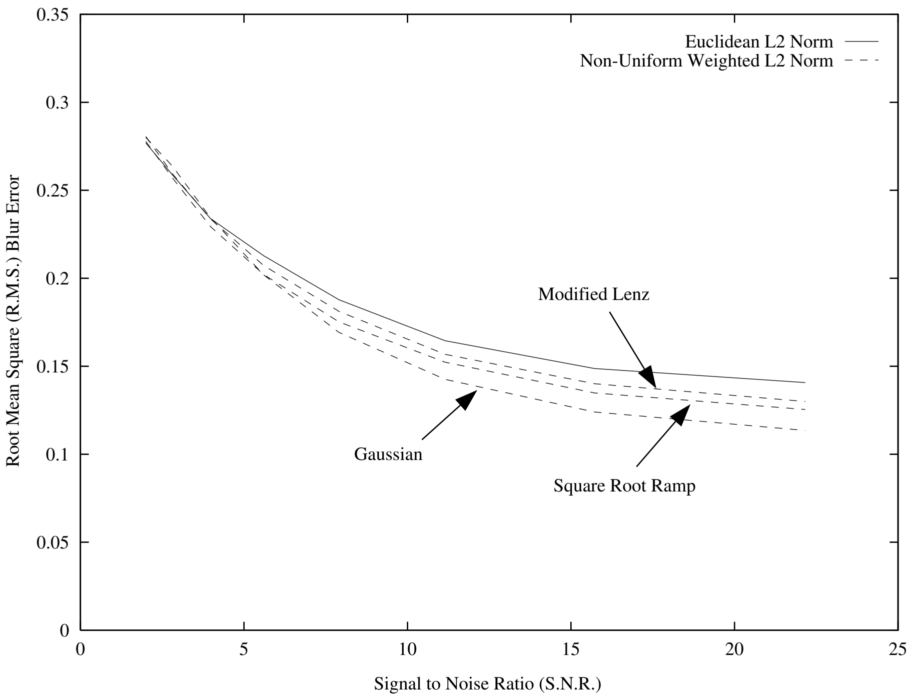


Figure 20: The estimation accuracy of the blur parameter σ for the line. The best performer is again the Gaussian weighting function, followed by the Square Root Ramp function, the Modified Lenz function, and the Euclidean L^2 norm in that order.

In [Canny, 86], Canny investigates the selection of the filter $f^c(x)$ to optimize the performance of a 1-D edge detector which declares edges at local maxima of:

$$\int_{-W}^{+W} I^c(a+x) \cdot f^c(-x)\,\mathrm{d}x \tag{47}$$

where $I^c(x)$ is the continuous 1-D input image (signal) and $[-W, W]$ is the 1-D feature window. If we translate the 1-D continuous setting of [Canny, 86] into the 2-D discretized setting which we used in this paper, the integral over the continuous 1-D window $[-W, W]$ becomes a sum over the discrete 2-D window S, and so the expression in Equation (47) becomes:

$$\max_{\{\rho,\theta\}} \sum_{(n,m)\in S} f(n+\rho\cos\theta, m+\rho\sin\theta;\theta) \cdot I(a+n, b+m) \tag{48}$$

where $f(x;y;\theta)$ is a 2-D version of the 1-D filter $f^c(x)$ rotated by an angle $180^o + \theta$ so as to be able to detect edges in that direction. Taking the maximum over the localization parameter ρ has been introduced to represent the operation of taking a local maxima and taking the maximum over the orientation parameter θ to denote the fact we have to consider edges in all possible directions. In [Canny, 86], the angle θ which maximizes the expression is taken to be the direction of the gradient, and then the local maxima is computed by finding the zero crossings of a second-order directional derivative in the direction of the gradient. Then, from Equation (48) we see that in a discrete 2-D setting the equivalent of [Canny, 86] would be to study the selection of $f(x, y; 0^o)$ to optimize a measure of the performance of edge detection. Comparing Equations (46) and (48) we note the following similarities and differences between the two approaches.

5.1.1 Similarities

The first similarity is the form of the quantity that is maximized and which then defines the edge (feature). In both cases the expression is a discrete correlation. In the model-matching approach we correlate with the kernel $w_{nm} \cdot \overline{F}(n, m; \mathbf{q})$ and in the optimal filtering approach the kernel is $f(n + \rho\cos\theta, m + \rho\sin\theta; \theta)$. The second similarity is that both this paper and the optimal filtering work investigate the selection of the correlation kernel to optimize some measure of performance. We finally note that selecting a uniform weighting function corresponds to choosing a matched-filter. It is then not surprising that the Euclidean L^2 norm is optimal for linear manifolds since matched filters are optimal for certain performance measures [Canny, 86] [Boie *et al.*, 86].

5.1.2 Differences

The first major difference between the two approaches is the domain over which the maximization is performed. So long as the feature $F(n, m; \mathbf{q})$ has a localization parameter $\rho \in \mathbf{q}$ and an orientation parameter $\theta \in \mathbf{q}$, the formulation in this paper can be regarded as a generalization of that in [Canny, 86] because there is an implicit spatial local maximization and a implicit maximization over orientation performed when we find the closest manifold point. However, most of the features which we consider have additional parameters.

Another difference is that in [Canny, 86] the formulation is 1-D and the extension to 2-D by rotating the filter in the direction of the gradient is somewhat ad-hoc. A better way to extend from 1-D to 2-D is possible when the filter $f^c(x, y; \theta)$ is steerable [Freeman and Adelson, 91]. In this case, the maximum over θ can be computed by first correlating with, say, $f(n, m; 0^o)$ and $f(n, m; 90^o)$, and then deducing a formula for the correlation with $f(n, m; \theta)$ as a function of θ. This function may then be maximized by taking the derivative with respect to θ. It so happens that the gradient of the (suboptimal) Gaussian filter eventually used in [Canny, 86] is steerable [Freeman and Adelson, 91] and the maximum response can be shown to be in the direction of the gradient. The other (optimal) filters which Canny derives are not, in general, steerable.

An important parameter omitted from the maximization in Equation (48) is the blurring or scale parameter σ which is assumed to be fixed in [Canny, 86]. It is suggested there that the Canny edge detector be applied for a number of different values of the blurring (scale) parameter. It is outside the scope of this paper to discuss the merits of such an approach. We just note that scale is an important unresolved issue and that our analysis demonstrates the treatment of it to be one of the major differences between the two approaches to feature detection.

The final significant difference is normalization. In the model matching approach both the feature model and the image data are normalized, whereas in the filtering approach the correlation is applied to the raw image data. Again, it is outside the scope of this paper to discuss the merits of normalization, but we note that it is another important difference between the two approaches to feature detection and that it therefore warrants further investigation.

5.2 Open Problems and Future Work

5.2.1 Analysis of Non-Linear Manifolds

A natural extension to this paper is in to investigate the optimality criterion for non-linear manifolds. As good first step would be to model the manifold using the first two terms in the Talyor expansion. This is a technically more demanding task than in the linear case. In the 1-parameter case, the manifold is a parabola. Finding the closest point on a parabola to a fixed point can be solved in at least two ways. One way involves solving a cubic equation. Perhaps a more promising way to proceed is to use constrained optimization [Hancock, 60], since it leads to a linear system of equations. The k-parameter case is even more difficult.

5.2.2 Feature Detection Robustness

In this paper we have investigated the selection of the fitting-function which optimizes the parameter estimation accuracy of a model-matching feature detector. A similar investigation for other measures of feature detection performance would be worthwhile. In particular, measures of feature detection robustness (such as rates of occurrence of false-positives and false-negatives) are of fundamental importance. Several technical difficulties arise when analyzing such measures. The most important such difficulty is the selection of a characteristic model for a "not-feature" [Nalwa and Binford, 86]. It appears that the only way to proceed is to assume the existence of two parametric manifolds, one characterizing the feature and one characterizing a "not feature." The manifold characterizing the "not feature" would need to depend upon the feature in question. Finding a perfect model for a "non-feature" may be difficult, but the use of a reasonable approximation should still yield acceptably good weighting functions. A reasonable example for the step edge might be a slope with uniform gradient, and a reasonable example for a corner might be a step edge. With two manifolds, it should then be possible to develop a general theory of how to select a weighting function that maximally discriminates between them.

5.2.3 Other Measures of Performance

Another idea is to try to analyze measures which are closely related to the performance of a feature detector. A simple example is the signal-to-noise ratio, which is a measure often used in the design of optimal-filtering feature detectors [Canny, 86]. It is straightforward to derive the weighting function which maximizes the signal-to-noise ratio, but unfortunately it turns out to be degenerate. The optimal weighting function always assigns a weight of 1 to exactly one of the pixels and 0 to the rest. At first glance, this is somewhat surprising given the fact that a matched-filter maximizes the signal-to-noise ratio. The difference is that in Canny's approach, the filtering operation corresponds to the computation of a distance in a 1-dimensional space. The matched-filter defines the direction in which it is optimal to compute this 1-dimensional distance. In our setting, we are considering distance functions in an N-dimensional space, and unfortunately the signal-to-noise ratio is maximized by a degenerate 1-dimensional weighting function. This is because the signal-to-noise ratio is too simple a measure of performance to take advantage of the full N-dimensional space.

Acknowledgements

I wish to thank my parents for a number of useful discussions. Thanks also go to my advisor Shree Nayar for reading several drafts of this paper, making a number of useful suggestions, and generally guiding this research.

A Appendix

A.1 Experimental Procedure

At a number of points we have presented experimental results comparing the parameter estimation performance of several weighted L^2 norms. We now briefly describe the experimental procedure used. This procedure is almost identical to that of [Baker *et al.*, 98] and also that of [Nalwa and Binford, 86]. The reader is referred to these papers for a more complete discussion of it.

The procedure assumes that it has been passed as inputs, (a) the signal to noise ratio snr to use, (b) the number of times to iterate M, (c) a subroutine that can evaluate the feature model $F(n, m; \mathbf{q})$ for any set of parameter values q_i in their ranges $[a_i, b_i]$, (d) a subroutine to randomly sample the Gaussian distribution, and (e) an implementation of a feature detector for the feature $F(n, m; \mathbf{q})$ using the weighting function of interest. We used the detector proposed in [Baker *et al.*, 98] extended to deal with weighted L^2 norms as described in Section 2. The procedure is then as follows:

Computing R.M.S. Parameter Estimation Error

1) For $i = 1 \dots k$, set $err[i] = 0.0$

2) Repeat the following M times:

   a) For $i = 1 \dots k$, randomly generate q_i from $[a_i, b_i]$

   b) Evaluate the feature model at $\mathbf{q} = (q_1, \dots, q_k)$ to give $F(n, m; \mathbf{q})$

   c) Set $\nu^2 = \sum_{(n,m)\in S}[F(n, m; \mathbf{q}) - \frac{1}{N}\sum_{(p,q)\in S} F(p, q; \mathbf{q})]^2$

   d) For all $(n, m) \in S$ add Gaussian noise with variance[4] $\left(\frac{2\times\nu}{snr}\right)^2$ to $F(n, m; \mathbf{q})$

   e) Apply the feature detector to estimate the parameters $\mathbf{q}' = (q'_1, \dots, q'_k)$

   f) For $i = 1 \dots k$, set $err[i] = err[i] + (q_i - q'_i) \times (q_i - q'_i)$

3) For $i = 1 \dots k$, output $\sqrt{\frac{err[i]}{M}}$ as the R.M.S. error in estimating parameter q_i

A.2 Noise and Normalization

In Section 3.2.1 we described the assumptions which we made about the noise $\eta(n, m; \mathbf{q})$. In particular we assumed that, (a) the noise has zero mean, (b) it is pairwise independent across the pixels $(n, m) \in S$, and (c) it has variance $\sigma^2(n, m; \mathbf{q})$. This noise is added to the raw feature $F(n, m; \mathbf{q})$ rather than the normalized one $\overline{F}(n, m; \mathbf{q})$. Since the feature detection algorithm of [Baker *et al.*, 98] works with normalized features we will now analyze the properties of the noise as it affects the normalized feature $\overline{F}(n, m; \mathbf{q})$. This normalized noise is denoted $\overline{\eta}(n, m; \mathbf{q})$. Under weak assumptions about the feature, the noise, and the weighting function, we will now show that to a first order approximation $\overline{\eta}(n, m; \mathbf{q})$ also satisfies the first two properties and has variance:

$$\overline{\sigma}^2(n, m; \mathbf{q}) = \frac{\sigma^2(n, m; \mathbf{q})}{\sum_{(p,q)\in S} w_{pq}\left[F(p, q; \mathbf{q}) - (\sum_{(r,s)\in S} w_{rs})^{-1} \sum_{(r,s)\in S} w_{rs} \cdot F(r, s; \mathbf{q})\right]^2} \tag{49}$$

We begin with a lemma which shows that to a first order approximation, the normalization procedure first scales the noise by the reciprocal of:

$$\Gamma = \left[\sum_{(n,m)\in S} w_{nm}\left[F(n, m; \mathbf{q}) - \frac{1}{\sum_{(p,q)\in S} w_{pq}} \sum_{(p,q)\in S} w_{pq} \cdot F(p, q; \mathbf{q})\right]^2\right]^{1/2} \tag{50}$$

and then projects it into an $N - 2$ dimensional subspace of $\mathbf{R}^N$.

[4] This is derived from the definition of signal to noise ratio for arbitrary features proposed in [Baker *et al.*, 98]: $snr = \frac{2\times\nu}{\sigma_{\text{noise}}}$ where σ^2_{noise} is the variance of the noise.

Lemma A.2.1 *Suppose that noise $\eta(n, m; \mathbf{q})$ is added to the feature instance $F(n, m; \mathbf{q})$ to give $F'(n, m; \mathbf{q}) = F(n, m; \mathbf{q}) + \eta(n, m; \mathbf{q})$ and then $F'(n, m; \mathbf{q})$ is normalized. If we write $\overline{F'}(n, m; \mathbf{q}) = \overline{F}(n, m; \mathbf{q}) + \overline{\eta}(n, m; \mathbf{q})$ then:*

$$\overline{\boldsymbol{\eta}} \;=\; \mathrm{P}_{(\mathbf{c}\oplus\mathbf{F})^\perp}\left(\frac{\boldsymbol{\eta}}{\Gamma}\right) + \mathrm{O}\left(\frac{\|\boldsymbol{\eta}\|_\mu}{\Gamma}\right)^2 \tag{51}$$

where $\mathrm{P}_{(\mathbf{c}\oplus\mathbf{F})^\perp}$ is the linear projection onto the subspace orthogonal to the feature $\mathbf{F}$ and the vector $\mathbf{c} = (1, 1, \ldots, 1)$.

Proof From $F'(n, m; \mathbf{q}) = F(n, m; \mathbf{q}) + \eta(n, m; \mathbf{q})$ and $\hat{\mathbf{c}} = \mathbf{c}/\|\mathbf{c}\|_\mu$ we have:

$$\mathbf{F}' - \langle \mathbf{F}', \hat{\mathbf{c}} \rangle_\mu \, \hat{\mathbf{c}} \;=\; \mathbf{F} - \langle \mathbf{F}, \hat{\mathbf{c}} \rangle_\mu \, \hat{\mathbf{c}} + \boldsymbol{\eta} - \langle \boldsymbol{\eta}, \hat{\mathbf{c}} \rangle_\mu \, \hat{\mathbf{c}} \tag{52}$$

Set $\mathbf{G} = \mathbf{F} - \langle \mathbf{F}, \hat{\mathbf{c}} \rangle_\mu \, \hat{\mathbf{c}}$ and $\boldsymbol{\zeta} = \boldsymbol{\eta} - \langle \boldsymbol{\eta}, \hat{\mathbf{c}} \rangle_\mu \, \hat{\mathbf{c}}$. Then, dividing Equation (52) by its norm gives:

$$\overline{F'}(n, m; \mathbf{q}) \;=\; \frac{G(n, m; \mathbf{q}) + \zeta(n, m; \mathbf{q})}{\left[\sum_{(n,m)\in S} w_{nm} [G(n, m; \mathbf{q}) + \zeta(n, m; \mathbf{q})]^2\right]^{1/2}} \tag{53}$$

Multiplying out the denominator of the RHS and noting that $\Gamma = \|\mathbf{G}\|_\mu$ yields:

$$\overline{F'}(n, m; \mathbf{q}) \;=\; [G(n, m; \mathbf{q}) + \zeta(n, m; \mathbf{q})] \left[\Gamma^2 + 2 \langle \boldsymbol{\zeta}, \mathbf{G} \rangle_\mu + \|\boldsymbol{\zeta}\|_\mu^2\right]^{-1/2} \tag{54}$$

Since $\overline{\mathbf{F}} = \mathbf{G}/\|\mathbf{G}\|_\mu$ we have:

$$\overline{\mathbf{F}'} \;=\; \left[\overline{\mathbf{F}} + \frac{\boldsymbol{\zeta}}{\Gamma}\right] \left[1 + 2\left\langle \frac{\boldsymbol{\zeta}}{\Gamma}, \overline{\mathbf{F}} \right\rangle_\mu + \mathrm{O}\left(\frac{\|\boldsymbol{\zeta}\|_\mu}{\Gamma}\right)^2\right]^{-1/2} \tag{55}$$

Expanding as a power series and rearranging gives:

$$\overline{\boldsymbol{\eta}} \;=\; \overline{\mathbf{F}'} - \overline{\mathbf{F}} \;=\; \frac{\boldsymbol{\zeta}}{\Gamma} - \left\langle \frac{\boldsymbol{\zeta}}{\Gamma}, \overline{\mathbf{F}} \right\rangle_\mu \overline{\mathbf{F}} + \mathrm{O}\left(\frac{\|\boldsymbol{\zeta}\|_\mu}{\Gamma}\right)^2 \tag{56}$$

Since $\boldsymbol{\zeta} = \boldsymbol{\eta} - \langle \boldsymbol{\eta}, \hat{\mathbf{c}} \rangle_\mu \, \hat{\mathbf{c}}$ and $\hat{\mathbf{c}} \perp \overline{\mathbf{F}}$ we can simplify further:

$$\overline{\boldsymbol{\eta}} \;=\; \frac{\boldsymbol{\eta}}{\Gamma} - \left\langle \frac{\boldsymbol{\eta}}{\Gamma}, \hat{\mathbf{c}} \right\rangle_\mu \hat{\mathbf{c}} - \left\langle \frac{\boldsymbol{\eta}}{\Gamma}, \overline{\mathbf{F}} \right\rangle_\mu \overline{\mathbf{F}} + \mathrm{O}\left(\frac{\|\boldsymbol{\zeta}\|_\mu}{\Gamma}\right)^2 \tag{57}$$

The proof is then completed by noting that $\{\hat{\mathbf{c}}, \overline{\mathbf{F}}\}$ is an orthonormal basis for $\mathbf{c} \oplus \mathbf{F}$ and that $\|\boldsymbol{\zeta}\|_\mu \leq \|\boldsymbol{\eta}\|_\mu$. □

Lemma A.2.1 would be sufficient for our purposes if it was not for the projection from $\mathbf{R}^N$ into the $N-2$ dimensional subspace $(\mathbf{c} \oplus \mathbf{F})^\perp$. When N is large, it is intuitively reasonable to expect that this projection will not significantly affect the character of the noise. To prove such a result we assume that the weighting function, the noise, and the feature are all fairly evenly distributed across the pixels. Since $\sum_{(n,m)\in S} w_{nm}^2 = 1$, the first part of this assumption corresponds to $w_{nm}^2 \approx 1/N$. If we assume that $\{\hat{\mathbf{a}}, \hat{\mathbf{b}}\}$ is an orthonormal basis for $\mathbf{c} \oplus \mathbf{F}$, then from $\|\hat{\mathbf{a}}\|_\mu^2 = 1$ we get $w_{nm}a_{nm}^2 \approx 1/N$. Similarly for $\hat{\mathbf{b}}$ we get $w_{nm}b_{nm}^2 \approx 1/N$. What we actually assume is that there is a $k \geq 1$ such that $w_{nm}^2 \leq k/N$, $a_{nm}^2 \leq \sqrt{k/N}$, and $b_{nm}^2 \leq \sqrt{k/N}$. We also assume that for all n, m, p, q we have $\sigma_{nm}^2 \leq k \cdot \sigma_{pq}^2$. Then we are able to prove the following lemma.

Lemma A.2.2 *Suppose that $\boldsymbol{\eta} = (\eta_{nm})$ is a random vector in $\mathbf{R}^N$ with* (a) $\mathrm{E}[\eta_{nm}] = 0$ *and* (b) $\mathrm{E}[\eta_{nm}\eta_{pq}] = \delta_{np}\delta_{mq}\sigma_{nm}^2$ *where δ_{ij} is the Kronecker delta function and σ_{nm}^2 is the variance of the random vector. Further, assume that $\hat{\mathbf{a}} = (a_{nm})$ and $\hat{\mathbf{b}} = (b_{nm})$ are constant unit vectors in $\mathbf{R}^N$ with $\left\langle \hat{\mathbf{a}}, \hat{\mathbf{b}} \right\rangle_\mu = 0$ and denote the projection of the noise onto the subspace orthogonal to $\hat{\mathbf{a}} \oplus \hat{\mathbf{b}}$ by $\boldsymbol{\eta}' = \mathrm{P}_{(\hat{\mathbf{a}}+\hat{\mathbf{b}})^\perp}(\boldsymbol{\eta})$. Then, if there exists a number $k \geq 1$ such that $\forall n, m, p, q : \sigma_{nm}^2 \leq k \cdot \sigma_{pq}^2$ and $\forall n, m : w_{nm}, a_{nm}^2, b_{nm}^2 \leq \sqrt{k/N}$ we have:*

(a) $\mathrm{E}[\eta'_{nm}] = 0$

(b) $\mathrm{E}[\eta'_{nm}\eta'_{pq}] = \sigma_{nm}^2 \left[\delta_{np}\delta_{mq} + \mathrm{O}\left(\frac{1}{N}\right)\right]$

Proof The proof of property (a) follows immediately from the fact that:

$$\boldsymbol{\eta}' \;=\; \boldsymbol{\eta} - \langle \hat{\mathbf{a}}, \boldsymbol{\eta} \rangle_\mu \,\hat{\mathbf{a}} - \left\langle \hat{\mathbf{b}}, \boldsymbol{\eta} \right\rangle_\mu \hat{\mathbf{b}} \tag{58}$$

combined with the linearity of the expectation operator and the bilinearity of the inner product. To prove part (b) we first consider $E[\eta'_{nm}\eta'_{nm}]$. By expanding the inner products in Equation (58), squaring, taking the expectation, and then simplifying we have:

$$E[\eta'_{nm}\eta'_{nm}] \;=\; \sigma_{nm}^2 \left[1 - w_{nm}\left(a_{nm}^2 + b_{nm}^2\right)\right]^2 + \sum_{(p,q)\neq(n,m)} \sigma_{pq}^2 w_{pq}^2 \left(a_{pq}a_{nm} + b_{pq}b_{nm}\right)^2 \tag{59}$$

Using the assumptions we have:

$$E[\eta'_{nm}\eta'_{nm}] \;=\; \sigma_{nm}^2 \left[1 - \mathrm{O}\left(\frac{1}{N}\right)\right]^2 + \sigma_{nm}^2 \sum_{(p,q)\neq(n,m)} \mathrm{O}\left(\frac{1}{N^2}\right) \tag{60}$$

And so we get:

$$E[\eta'_{nm}\eta'_{nm}] \;=\; \sigma_{nm}^2 \left[1 + \mathrm{O}\left(\frac{1}{N}\right)\right] \tag{61}$$

Finally we must consider $E[\eta'_{nm}\eta'_{pq}]$ where $(n, m) \neq (p, q)$. Again from Equation (58) we derive:

$$\begin{aligned} E[\eta'_{nm}\eta'_{pq}] \;&=\; -\,w_{nm}a_{nm}\sigma_{nm}^2\left(a_{pq} + b_{pq}\right) - w_{pq}a_{pq}\sigma_{pq}^2\left(a_{nm} + b_{nm}\right) \\ &\quad + \sum_{(r,s)\in S} w_{rs}^2\sigma_{rs}^2\left(a_{nm}a_{rs} + b_{nm}b_{rs}\right)\left(a_{pq}a_{rs} + b_{pq}b_{rs}\right) \end{aligned} \tag{62}$$

And again using the assumptions we have:

$$E[\eta'_{nm}\eta'_{pq}] \; = \sigma^2_{nm} \cdot \mathrm{O}\left(\frac{1}{N}\right) \tag{63}$$

which completes the proof. □

Combining Lemmas A.2.1 and A.2.2 we can prove that to a first order approximation, $\overline{\boldsymbol{\eta}}$ has mean zero, is pairwise independent across the pixels, and has variance given by Equation (49). To do this, we set $\hat{\mathbf{a}}$ of Lemma A.2.2 to be $\hat{\mathbf{c}}$ of Lemma A.2.1, and $\hat{\mathbf{b}}$ to be $\overline{\mathbf{F}}$. Then, the assumptions in Lemma A.2.2 which we need to make are that the normalized feature $\overline{\mathbf{F}}$, the weighting function, and the noise are reasonably evenly distributed across the pixels.

The assumption about the weighting function is reasonable since we expect that a large fraction of the window is fairly heavily weighted even if a lot of the pixels are lightly weighted. It is also reasonable to assume that the noise does not vary too much over the window. Finally, the assumption about $\overline{\mathbf{F}}$ is also very reasonable since it is satisfied for any feature that has a significant fraction of the pixels at approximately the minimum intensity and a significant fraction at approximately the maximum. This is the case for all edges where roughly half of the pixels are at the minimum intensity and the other half are at the maximum. It also holds for lines and corners, so long the lines are not too wide or too thin, and the angles subtended by the corners are not too large (close to 360°) or too small (close to 0°).

A.3 Optimization Problems

In this section, we present a number of lemmas which are needed to derive the optimality of the weighting functions in Section 3.3. We first present the results required for 1-parameter linear manifolds and afterwards those required for k-parameter linear manifolds.

A.3.1 1-Parameter Linear Manifolds

Lemma A.3.1 *Given two constant vectors* $\mathbf{a} = (a_{nm}) \in \mathbf{R}^N$ *and* $\mathbf{b} = (b_{nm}) \in \mathbf{R}^N$ *where* $\forall n, m \,:\, a_{nm}, b_{nm} > 0$*, the minimum value of:*

$$\mathbf{P}_1(w_{nm}) \; = \; \frac{\left[\sum_{(n,m)\in S} \cdot b_{nm} w^2_{nm}\right]^{1/2}}{\sum_{(n,m)\in S} a_{nm} \cdot w_{nm}} \tag{64}$$

is attained when $w_{nm} = a_{nm}/b_{nm}$ *and is given by* $\left[\sum_{(n,m)\in S} a^2_{nm}/b_{nm}\right]^{-1/2}$.

Proof Setting $v_{nm} = w_{nm} \cdot \sqrt{b_{nm}}$ yields:

$$\mathbf{P}_1(v_{nm}) \;=\; \frac{\left[\sum_{(n,m)\in S} v_{nm}^2\right]^{1/2}}{\sum_{(n,m)\in S} \frac{a_{nm}}{\sqrt{b_{nm}}} \cdot v_{nm}} \tag{65}$$

which, by the Cauchy-Schwarz inequality, is minimized when $v_{nm} = \frac{a_{nm}}{\sqrt{b_{nm}}}$ or equivalently when $w_{nm} = a_{nm}/b_{nm}$. Plugging this expression into $\mathbf{P}_1(w_{nm})$ completes the proof. □

Lemma A.3.2 *Suppose the two vector functions of the parameters $\mathbf{a}(\mathbf{q}) = (a_{nm}(\mathbf{q})) \in \mathbf{R}^N$ and $\mathbf{b}(\mathbf{q}) = (b_{nm}(\mathbf{q})) \in \mathbf{R}^N$ are related by $a_{nm}(\mathbf{q}) = c_{nm} \cdot b_{nm}(\mathbf{q})$ where $\mathbf{c} = (c_{nm})$ is a constant vector. Also assume that $\forall n, m, \mathbf{q} \,:\, a_{nm}(\mathbf{q}), b_{nm}(\mathbf{q}), \rho(\mathbf{q}) > 0$ where $\rho(\mathbf{q})$ is a scalar function of the parameters. Then, the minimum value of:*

$$\mathbf{P}_2(w_{nm}) \;=\; \int \rho(\mathbf{q}) \frac{\left[\sum_{(n,m)\in S} b_{nm}(\mathbf{q}) \cdot w_{nm}^2\right]^{1/2}}{\sum_{(n,m)\in S} a_{nm}(\mathbf{q}) \cdot w_{nm}} \,\mathrm{d}\mathbf{q} \tag{66}$$

is attained when $w_{nm} = c_{nm}$ and is given by $\int \rho(\mathbf{q}) \left[\sum_{(n,m)\in S} a_{nm}^2(\mathbf{q})/b_{nm}(\mathbf{q})\right]^{-1/2} \mathrm{d}\mathbf{q}$.

Proof From Lemma A.3.1 the integrand is minimized when $w_{nm} = a_{nm}(\mathbf{q})/b_{nm}(\mathbf{q}) = c_{nm}$. Since c_{nm} is independent of $\mathbf{q}$, the integral itself is also minimized when $w_{nm} = c_{nm}$. Inserting this expression into $\mathbf{P}_2(w_{nm})$ completes the proof. □

A.3.2 k-Parameter Linear Manifolds

Lemma A.3.3 *Suppose that we are given two constant vectors $\boldsymbol{\sigma} = (\sigma_{nm}) \in \mathbf{R}^N$ and $\mathbf{a} = (a_{nm}) \in \mathbf{R}^N$, and a constant Linear subspace $\mathrm{D} \subset \mathbf{R}^N$. Then if $\mathbf{b} = (b_{nm}) = \mathrm{P}_{\mathrm{D}^\perp}(\mathbf{a})$ is the projection of $\mathbf{a}$ onto the subspace orthogonal to D, then $w_{nm} = \sigma_{nm}^{-2}$ is a stationary point of:*

$$\mathbf{P}_3(w_{nm}) \;=\; \frac{\sum_{(n,m)\in S} w_{nm}^2 \sigma_{nm}^2 b_{nm}^2}{\left[\sum_{(n,m)\in S} w_{nm} a_{nm} b_{nm}\right]^2} \tag{67}$$

Proof If we denote $\sum_{(n,m)\in S} w_{nm}^2 \sigma_{nm}^2 b_{nm}^2$ by **top** and $\left[\sum_{(n,m)\in S} w_{nm} a_{nm} b_{nm}\right]^2$ by **bot**, then we have:

$$\mathbf{bot} \times \left.\frac{\partial \mathbf{top}}{\partial w_{pq}}\right|_{w_{nm}=\sigma_{nm}^{-2}} = 2\left[b_{pq}^2 + \sum_{(n,m)\in S} \frac{b_{nm}}{\sigma_{nm}^2}\frac{\partial b_{nm}}{\partial w_{pq}}\right]\left[\sum_{(n,m)\in S} \frac{b_{nm} a_{nm}}{\sigma_{nm}^2}\right]^2 \tag{68}$$

And:

$$\mathbf{top} \times \left.\frac{\partial \mathbf{bot}}{\partial w_{pq}}\right|_{w_{nm}=\sigma_{nm}^{-2}} = 2\left[a_{pq} b_{pq} + \sum_{(n,m)\in S} \frac{a_{nm}}{\sigma_{nm}^2}\frac{\partial b_{nm}}{\partial w_{pq}}\right]\left[\sum_{(n,m)\in S} \frac{b_{nm} a_{nm}}{\sigma_{nm}^2}\right]\left[\sum_{(n,m)\in S} \frac{b_{nm}^2}{\sigma_{nm}^2}\right] \tag{69}$$

Setting equal and simplifying gives:

$$\left[b_{pq}^2 + \left\langle \mathbf{b}, \frac{\partial \mathbf{b}}{\partial w_{pq}} \right\rangle_\mu \right] \langle \mathbf{a}, \mathbf{b} \rangle_\mu \;=\; \left[a_{pq} b_{pq} + \left\langle \mathbf{a}, \frac{\partial \mathbf{b}}{\partial w_{pq}} \right\rangle_\mu \right] \langle \mathbf{b}, \mathbf{b} \rangle_\mu \tag{70}$$

But since $\mathbf{b} = \mathrm{P}_{\mathrm{D}^\perp}(\mathbf{a})$ it follows that $\langle \mathbf{a}, \mathbf{b} \rangle_\mu = \langle \mathbf{b}, \mathbf{b} \rangle_\mu$ which when differentiated gives:

$$a_{pq} b_{pq} + \left\langle \mathbf{a}, \frac{\partial \mathbf{b}}{\partial w_{pq}} \right\rangle_\mu \;=\; b_{pq}^2 + 2 \left\langle \mathbf{b}, \frac{\partial \mathbf{b}}{\partial w_{pq}} \right\rangle_\mu \tag{71}$$

Hence, it remains to be shown that:

$$\left\langle \mathbf{b}, \frac{\partial \mathbf{b}}{\partial w_{pq}} \right\rangle_\mu \;=\; 0 \tag{72}$$

Since $\mathbf{b} = \mathbf{a} - \mathbf{c}$, where $\mathbf{a}$ is a constant and $\mathbf{c} = \mathbf{c}(w_{nm}) \in \mathrm{D}$, we have:

$$\frac{\partial \mathbf{b}}{\partial w_{pq}} \in \mathrm{D} \tag{73}$$

and so the proof is completed by noting that $\mathbf{b} \in \mathrm{D}^\perp$. $\square$

Lemma A.3.4 *Suppose that we are given a constant vector $\boldsymbol{\sigma} = (\sigma_{nm}) \in \mathbf{R}^N$, a Linear subspace $\mathrm{D} = \mathrm{D}(\mathbf{q}) \subset \mathbf{R}^N$ which is a function of the parameters $\mathbf{q}$, and a scalar function of the parameters $\rho = \rho(\mathbf{q})$. Further, assume that $\mathbf{a}(\mathbf{q}) = (a_{nm}(\mathbf{q})) \in \mathbf{R}^N$ is a vector function of the parameters $\mathbf{q}$, and $\mathbf{b}(\mathbf{q}) = (b_{nm}(\mathbf{q})) = \mathrm{P}_{\mathrm{D}^\perp}(\mathbf{a})$ is the projection of $\mathbf{a}$ onto the subspace orthogonal to D. Then, $w_{nm} = \sigma_{nm}^{-2}$ is a stationary point of:*

$$\mathbf{P}_4(w_{nm}) \;=\; \int \rho(\mathbf{q}) \frac{\left[\sum_{(n,m)\in S} w_{nm}^2 \sigma_{nm}^2 b_{nm}^2(\mathbf{q})\right]^{1/2}}{\sum_{(n,m)\in S} w_{nm} a_{nm}(\mathbf{q}) b_{nm}(\mathbf{q})} \, \mathrm{d}\mathbf{q} \tag{74}$$

Proof Follows immediately from Lemma A.3.3, together with the fact that the integral operator commutes with the partial derivative and the fact that the square-root is a monotonic function of its argument. $\square$

References

[Abdou and Pratt, 79] I.E. Abdou and W.K. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Proceedings of the IEEE,* 67:753–763, 1979.

[Abramatic, 81] J.F. Abramatic, "Why the Simplest 'Hueckel' Edge Detector Is a Roberts Operator," *Computer Graphics and Image Processing,* 17:79–83, 1981.

[Baker *et al.*, 98] S. Baker, S.K. Nayar, and H. Murase, "Parametric Feature Detection," *International Journal of Computer Vision,* to appear in 1998.

[Boie *et al.*, 86] R.A. Boie, I.J. Cox, and P. Rehak, "On Optimum Edge Recognition using Matched Filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 100–108, 1986.

[Canny, 86] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[Conway, 85] J.B. Conway, *A Course in Functional Analysis,* Springer-Verlag, 1985.

[Deriche, 87] R. Deriche, "Using Canny's Criteria To Derive A Recursively Implemented Optimal Edge Detector," *International Journal of Computer Vision,* 1:167–187, 1987.

[Freeman and Adelson, 91] W.T. Freeman and E.H. Adelson, "The Design and Use of Steerable Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 13:891–906, 1991.

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

[Halmos, 74] P.R. Halmos, *Measure Theory,* Springer-Verlag, 1974.

[Hancock, 60] H. Hancock, *Theory of Maxima and Minima,* Dover, 1960.

[Haralick 84] R.M. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6:58–68, 1984.

[Hartley, 85] R. Hartley, "A Gaussian-Weighted Multiresolution Edge Detector," *Computer Vision, Graphics, and Image Processing,* 30:70–83, 1985.

[Healey and Kondepudy, 91] G. Healey and R. Kondepudy "Modeling and Calibrating CCD Cameras for Illumination Insensitive Machine Vision," *SPIE Optics, Illumination, and Image Sensing for Machine Vision VI,* 1614:121–132, 1991.

[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.

[Knuth, 81] D.E. Knuth, *The Art of Computer Programming, Volume II: Seminumerical Algorithms*, Addison-Wesley, 1981.

[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.

[Meer and Weiss, 92] P. Meer and I. Weiss, "Smoothed Differentiation Filters for Images," Journal of Visual Communication and Image Representation, 3:58–72, 1992.

[Modestino and Fries, 77] J.W. Modestino and R.W. Fries, "Edge Detection in Noisey Images Using Recursive Digital Filtering," *Computer Graphics and Image Processing,* 6:409–433, 1977.

[Morgenthaler, 81] D.G. Morgenthaler, "A New Hybrid Edge Detector," *Computer Graphics and Image Processing,* 16:166–176, 1981.

[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.

[Nayar *et al.*, 96] 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, San Francisco, 1996.

[O'Gorman, 78] F. O'Gorman, "Edge Detection Using Walsh Functions," *Artificial Intelligence,* 10:215–223, 1978.

[Oja, 83] E. Oja, *Subspace Methods of Pattern Recognition*, Research Studies Press, 1983.

[Paton, 75] K. Paton, "Picture Description Using Legendre Polynomials," Computer Graphics and Image Processing, 4:40–54, 1975.

[Pratt, 90] W.K. Pratt, *Digital Image Processing,* John Wiley & Sons, 1990.

[Rohr, 92] K. Rohr, "Recognizing Corners by Fitting Parametric Models," *International Journal of Computer Vision,* 9:213–230, 1992.

[Rosenfeld, 81] A. Rosenfeld, "The Max Roberts Operator is a Hueckel-Type Edge Detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 3:101–103, 1981.

[Sarkar and Boyer, 91] S. Sarkar and K.L. Boyer, "On Optimal Infinite Impulse Response Edge Detection Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 13:1154–1171, 1991.

[Shanmugam *et al.*, 79] K.S. Shanmugam, F.M. Dickey, and J.A. Green, "An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1:37–49, 1979.

[Zucker and Hummel, 81] S.W. Zucker and R.A. Hummel, "A Three-Dimensional Edge Operator," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:324–331, 1981.