Correspondence

Modeling Edges at Subpixel Accuracy Using the Local Energy Approach

M. Kisworo, S. Venkatesh, and G. West

Abstract—**In this paper we describe a new technique for 1-D and 2-D edge feature extraction to subpixel accuracy using edge models and the local energy approach. A candidate edge is modeled as one of a number of parametric edge models, and the fit is refined by a least-squared error fitting technique.**

Index Terms—**Edge detection, local energy, subpixel feature detection, computer vision.**

I. Introduction

Most images used for computer and machine vision are of low resolution because of the underlying television standards used and the need for fast acquisition and low memory requirement. Subpixel measurement is highly desirable because a low-resolution imaging system can be used for more accurate applications, such as dimensional measurement for inspection. This paper presents a new technique for subpixel measurement based on the concept of edge models and the local energy approach.

Various methods have been proposed for the measurement of edges at subpixel accuracy. MacVicar-Whelan *et al.* [1] used the gradient operator to determine the pixel location of zero crossings and then linearly interpolated the location. Hueckel [2] developed an algorithm to fit the data in Hilbert space and interpolated to compute the subpixel location of the edge. Nevatia *et al.* [3] used matched filters convolved with the data to get the maxima of the filter response and compute the subpixel step location. Tabatabai *et al.* [4] fitted the first three statistical moments to a step edge model by determining the optimal values of the moments. Huertas *et al.* [5] implemented LOG masks combined with a facet model followed by interpolation to detect edges at subpixel accuracy. Lyvers *et al.* [6] developed a subpixel edge operator that locates edges by fitting the spatial moments of a step edge model to the data.

In all of these techniques, the subpixel analysis is based on using a perfect step edge as the underlying edge model. Although useful, the choice of a step edge restricts the analysis because there are other types of edges present, such as roofs, ramps, etc. Perona *et al.* [18] propose a subpixel technique based on energy models to determine the localization of steps, peaks, and roofs. However, their subpixel technique involves an interpolation method based on fitting a 2nd-order model (a paraboloid) to the computed energy function. In this paper, an edge feature extraction technique with subpixel accuracy is presented using a general edge model. With this technique step, ramp, and roof edges can be detected and classified at subpixel accuracy.

To perform this clasification and detection, the local energy model is used. There are three reasons for this choice. First, not being a gradient-based approach, the local energy model does not suffer from the problem of amplification of high-frequency noise. Second, this model does not give rise to the detection of false positives arising from points of inflection, or points of maximum gradient in the image. Third, feature extraction processes that require the use of several optimal operators and techniques [7], [8] to detect different edge types must resolve the problem of integrating the outputs of different operators. The local energy operator identifies all feature types without the need to invoke multiple operators.

To detect and classify edges at subpixel accuracy, a parameterized function is used to model an edge. The initial model is determined by a decision process that is based on the response of the signal to the local energy filters [9]. The model is then fitted in a least-squared error sense to the signal in the energy domain. The parameters describing the best fit of the model to the data define the position of the edge to subpixel accuracy. The advantages of this method are twofold. First, it is not limited to the subpixel edge detection of monotonically increasing and decreasing sequences. Second, and more important, it can detect all feature types at subpixel accuracy robustly.

The layout of this correspondence is as follows: The underlying principles of the local energy approach are presented in Section II. The development of the edge feature extraction technique for one-dimensional signals is presented in Section III, and results are presented in Section IV. The extension to two-dimensional signals is described in Section V, followed by the results in Section VI.

II. Feature Detection Using Local Energy

Analyzing visual features, Morrone *et al.* [10] proposed an alternate method for feature extraction based on discerning how features are built up in an image, rather than by considering differential properties. This meant examining the Fourier expansion of a luminance profile function and studying the properties of the components of this expansion. It was first noted that in the Fourier expansion of a negative-going step edge, all the components are in phase and have a 90° phase in the cosine expansion at the point where the step occurs. The point of the step edge is, moreover, the only point in this profile at which this phase congruency property occurs. Needless to say, other luminance profiles exhibit different types of phase congruency. At the peak point in a positive roof profile, all the components of the Fourier expansion are in 0° phase. In some luminance profiles—for example, a trapezoidal profile—no such points of total phase congruency exist, but the points for which the variance of the phase values is minimum are still of interest. In a trapezoidal profile, these points correspond to the places where Mach bands are perceived.

To locate these points of local *phase congruency*, it is necessary to consider a quadrature pair of functions, namely, the original profile and its Hilbert transform to define *local energy*. The norm of the *local energy* function is the square root of the sums of squares of the function and its Hilbert transform. The maxima of the norm of the local energy function are coincident with both the points of maximum phase congruency and visual features. As a computationally simpler alternative to computing local energy in terms of the image and its Hilbert transform, it was suggested [10] that a quadrature pair of functions obtained by convolving the image with a set of quadrature filters be used. The norm of local energy is then computed as the square root of the sums of squares of the functions obtained by convolving the image profile with a set of quadrature masks.

Manuscript received September 12, 1991; revised July 8, 1993. Recommended for acceptance by Associate Editor E. Delp.

The authors are with the Cognitive Systems Group, School of Computing, Curtin University of Technology, Perth, Western Australia.

IEEE Log Number 9214457.

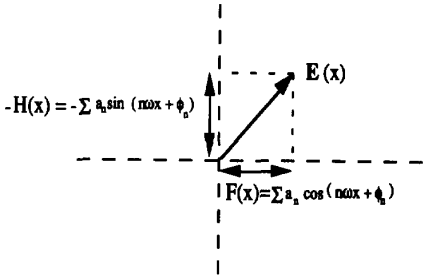

Fig. 1. The local energy function, whose components at any point x are the function $F(x)$ and its Hilbert transform $H(x)$.

Experimental results indicated that the computed maxima of the norm of the local energy function coincided with visual features.

More formally, an absolutely integratable function F defined in an interval can be expanded in terms of the *Fourier series* and is given as:

$$F(x) = \sum a_n \cos(n\omega_n x + \Phi_n), \tag{1}$$

where $a_n > 0$, Φ_n is the phase shift of the nth term and the summation is taken over positive integers. The Hilbert transform of F is given as

$$H(x) = -\sum a_n \sin(n\omega_n x + \Phi_n). \tag{2}$$

Consider an analytic signal E given as

$$E(x) = F(x) - iH(x) = (F(x), H(x))$$

that can be written as

$$\sum a_n(\cos(n\omega_n x + \Phi_n) + i \sin(n\omega_n x + \Phi_n)).$$

The function E, therefore, is in general an infinite vector sum, where the nth component has a length of a_n and a phase of $n\omega_n x + \Phi_n$.

Definition 1: The local energy function E of a function F is a complex-valued function, with a real component that is the function F and with an imaginary component that is (minus) the corresponding Hilbert transform H at the point x (see Fig. 1). Thus,

$$E(x) = F(x) - iH(x). \tag{3}$$

Theorem 1: Given an antisymmetric and symmetric quadrature filter pair, the local energy function can be computed from the complex-valued function whose real component is f_e, obtained by convolving F with the symmetric mask, and whose imaginary component is (minus) f_o, obtained by convolving F with the antisymmetric mask. The maxima of the norm of the local energy function computed in this manner coincides with points of maximum phase congruency of E. (See 11 for the full proof.)

III. Subpixel Feature Detection in One-Dimensional Signals

To detect edges at subpixel level, it is necessary to first classify the feature at pixel resolution as a step, a roof, or a ramp edge. Then, for each feature type, a model of the ideal feature is matched to the signal (in the least-squared sense) so that the parameters of the *best fit* model provide the subpixel parameters of the signal being analyzed.

A. Predicate-Based Feature Identification

To achieve the first step of feature classfication at pixel level, a predicate-based algorithm [9] has been developed, based on the observation that the output of the convolution of a signal with a set of quadrature filters characterizes the feature type of the signal

TABLE I
Output of the Cauchy Quadrature Filters when Applied to Different Feature Types

Discontinuity	Anti-symmetric	Symmetric	Model
	Maximum	Positive zero-crossing	Positive step
	Minimum	Negative zero-crossing	Negative step
	Negative zero-crossing	Maximum	Positive roof
	Positive zero-crossing	Minimum	Negative roof
	Maximum	Minimum	Positive ramp
	Maximum	Maximum	Positive ramp
	Minimum	Maximum	Negative ramp
	Minimum	Minimum	Negative ramp

[12]. Steps, ramps, and roofs have unique response patterns when convolved with the quadrature filters, and the typical responses of these feature types to Cauchy quadrature filters are shown in Table I. Other filters, including quadrature filters, have been investigated and analysed recently [16]–[18]. A mathematical derivation of the response of Cauchy filters to step and ramp edges is detailed in [19].

The responses of the quadrature filters are used to match a signal to one of the eight predefined feature types shown in Table I. Predicates are applied to the response of the signal to the antisymmetric and symmetric filters, and the predicate-based algorithm identifies feature types by this pattern matching process. The algorithm outputs the best matched model and a *measure of match* that gives the degree of match between the signal and the model. A threshold value can be applied to filter out discontinuity points that have a *measure of match* value less than the specified threshold value.

The *measure of match* value is computed using the maximum likelihood probability method. Within a window size of w, the positions of maxima, minima, and zero crossings are determined. The significance function s (see Fig. 2) of feature k termed s_k is defined as:

$$s_k = \frac{w}{2} - d_k, \tag{4}$$

where w is the window size, and d_k is the distance from the edge pixel P to the minimum, maximum, or zero crossing. Where one of the minimum, maximum, or zero crossing does not appear in the window w, the probability is set to zero and hence ignored in the analysis. The greater the values of s_k, the greater the probability that these are the best points to use. There may be many candidate points within the window, and only the most significant are taken as correct points.

B. Computation of the Edge Parameters

The next step is to fit the appropriate ideal model to the signal. For all feature types, the ideal edge model is characterized by four parameters: *edge_start, edge_steady, edge_end,* and *edge_height* (see Fig. 3). These parameters are sufficient to represent the six basic edge models. For example, a step edge has the same values of *edge_start* and *edge_steady* with *edge_end* not defined (as it will be outside the window w), while a roof edge has identical values for *edge_steady* and *edge_end*. An iterative least-squared error fitting technique with

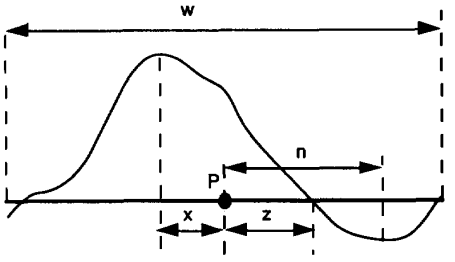

Fig. 2. Estimating the "measure of match" value. P is the edge pixel, w = window size, x = distance from P to the maxima, n = distance from P to the minima, and z = distance from P to the zero-crossing.

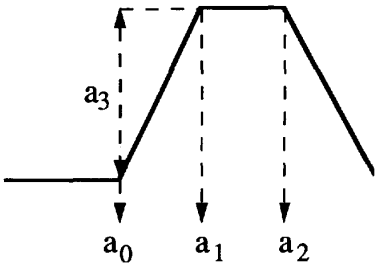

Fig. 3. Edge model identified by four edge parameters: a_0 = *edge_start*, a_1 = *edge_steady*, a_2 = *edge_end*, a_3 = *edge_height*.

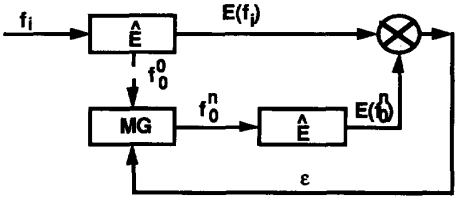

Fig. 4. The iterative least-squared error fitting. MG is the model generator that generates edge paramaters based on the value of the error ε.

a priori information is used to estimate the edge parameters indicated in Fig. 3 at subpixel accuracy. The initial application of the predicate-based algorithm at pixel level to the input signal f_i determines the feature type. This feature type is used to generate the initial edge model f_o^0, otherwise known as the *a priori* model. The local energy function of this initial model $E(f_o^0)$ is computed, and the position of the maximum is compared to that generated from the local energy function of the signal $E(f_i)$ (see Fig. 4).

The error sign ε_0 is defined as the difference between the local energies $E(f_i)$ and $E(f_o^0)$. This error signal is then used to alter the parameters of the model and generate a new model, f_o^n. The process continues iteratively (except for the step edge case, Case 1 below) until the least-squared error is obtained and the best estimate signal f_o^{op} of the original signal f_i is computed. Once the optimal model is computed, the edge parameters, *edge_start, edge_steady, edge_end,* and *edge_height* characterize the input signal at subpixel resolution.

A brief outline of the least-squared error fitting process follows.

Case 1: If the input signal f_i is a step edge, the local energy of f_i has a single maximum. To compute the edge height of f_i, we compare the local energy of f_i with that of a unit step signal f_1. The

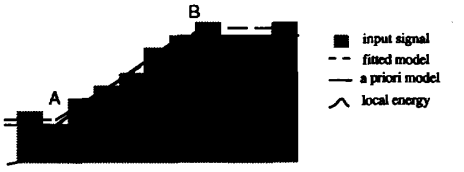

Fig. 5. Least-squared error fitting to a ramp signal.

local energy of a unit step signal f_1 is

$$E_{\text{norm}} = (f_1(x) * a(x))^2 + (f_1(x) * s(x))^2 = \left(\int_0^{+\infty} a(x)dx\right)^2 + \left(\int_0^{+\infty} s(x)dx\right)^2,$$

where s and a are the symmetric and antisymmetric filters, respectively. It has been established that the energy function of $c_1 f_1(x)$ is c_1^2 times the local energy of $f_1(x)$ [19]. Using this relation, given the local energy of $f_1(x)$, we can compute the factor c_1, which is the value of *edge_height* of the fitted model as

$$\frac{E(x)}{E_{\text{norm}}(x)} = c_1^2.$$

Note that no iteration of the algorithm is required in this case.

Case 2: If the input signal f_i is a ramp edge as shown in Fig. 5, the local energy will have two maxima, as shown at A and B. First, the *a priori* model is generated by calculating the edge height in much the same way as in Case 1. In computing the correct slope of the ramp, the position of A is fixed. If the point A is fixed, and the edge height is calculated, the question that remains is how to fix the position of the point where the ramp meets the plateau, i.e., position B. To do so, the error signal is used as a heuristic. If the error ε is positive, the slope of the model is increased, and if ε is negative, the slope of the model is decreased. This then fixes the point where the ramp meets the plateau. This initial model is then used to compute the next value of ε, and the process of fitting continues by gradient descent until the error is minimized in the least-square sense.

Case 3: If the input signal f_i is a roof edge, the fitting process is similar to that for a ramp edge. A roof edge is distinguished from a ramp by the number of energy maxima in the window under consideration. Although the local energy has two maxima for a ramp, it has three maxima for a roof. Once the positions of A and B have been established, if a third maxima exists within a distance equal to half the window size (w) pixels of B, the feature is a roof edge.

IV. Results for One-Dimensional Signals

In experiments, the Cauchy function [14] has been used for the quadrature filter functions. It is defined as a Poisson probability distribution in the frequency domain:

$$C_n(x) = x^n \exp(-x). \tag{5}$$

Transforming (5) into the Fourier domain and separating its real and imaginary parts yields

$$a_{n-1}(\omega) = -\text{Im}(1 + i\omega)^{-n} \tag{6}$$

$$s_{n-1}(\omega) = \text{Re}(1 + i\omega)^{-n}. \tag{7}$$

where i is $\sqrt{-1}$.

A value of $n = 6$ is used since this is the minimum value of n that guarantees independence to the contrast sensitivity function [14]. The symmetric and the antisymmetric functions (s and a, respectively, are

$$a_5(x) = x(6 - 20x^2 + 6x^4)/(1 + x^2)^6 \tag{8}$$

$$s_5(x) = (1 - 15x^2 + 15x^4 - x^6)/(1 + x^2)^6, \tag{9}$$

where $x = \tan(o) = t/\sigma$ and σ is a scaling parameter.

TABLE II
EDGE PARAMETERS COMPUTED BY OUR ALGORITHM, AS COMPARED WITH TABATABAI'S RESULTS

No	Input	Feature Type	Tabatabai Result	Computed edge parameters			
			step	a_0	a_1	a_2	$(a_1+a_0)/2$
1	0 0 0 1 1 1	positive step	2.5	2.5	2.5	∞	2.5
2	0 0 0.5 1 1	positive step	2.0	2.0	2.0	∞	2.0
3	1 1 1 1 0 0 0 0	negative step	3.5	3.5	3.5	∞	3.5
4	1 1 1 1 0.5 0 0	negative step	4.0	4.0	4.0	∞	4.0
5	1 1 1.5 2 2.5 3 3.5 4	positive ramp	4.0	1.5	6.5	∞	4.0
6	1 1 2 3 3.5 7 8 9 10 10	positive ramp	4.5	1.4	7.7	∞	4.5
7	4 4 3.5 3.2 2.5 2 1.5 1	negative ramp	4.2	1.8	6.5	∞	4.1
8	10 10 9 8 7.5 4 3 2 1 1	negative ramp	4.5	1.4	7.7	∞	4.5
9	0 0 0.5 0.6 0.7 0.8 0.8 0.6 0 0	positive roof	N/A	1.5	5.8	8.1	N/A
10	4 4 4 2 1.5 4 4	negative roof	N/A	2.0	3.7	4.3	N/A

Using these filters, the techniques outlined in the earlier section have been applied to one-dimensional signals and the results compared to those of Tabatabai [4]. Table II shows the results of applying the two techniques to various step edges. The input sequence of pixel values is analyzed and some intermediate parameters tabulated. Note that for most of the examples, if the edge position is defined as the mid-distance between a_0 and a_1, the results are in close agreement with those of Tabatabai. Two results (rows 9 and 10) are shown for roof edges. Both edges are detected by our method. As expected, Tabatabai's operator does not detect roofs, as it is not designed for this type of edge.

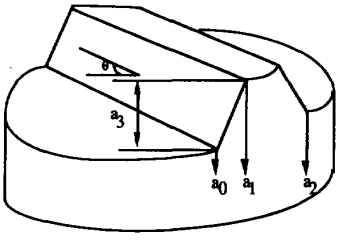

Fig. 6. Two-dimensional edge parameters at noncorner points; a_0 = *edge_start*, a_1 = *edge_steady*, a_3 = *edge_end*, a_4 = *edge_height*, and θ = *edge_orientation*.

V. EXTENSION TO TWO-DIMENSIONAL SIGNALS

The techniques for 1-D signals can be extended to 2-D signals. In particular, the technique can be extended to certain types of 2-D features. The question arises as to what is a 2-D feature. In the context of this analysis, a 2-D feature is defined as one for which the curvature along the edge describing the feature is continuous. Therefore, 2-D features include straight edges, arcs, ellipses, polynomials, etc. This does not include vertices. Vertices are important features from a computer vision point of view because of the robustness of the position and orientation of the vertex in the presence of disconnected edges and noise. Here it is argued that a vertex can be regarded as a junction between two features that satisfy the constraint of continuous curvature, e.g., between two straight lines, or between one line and an arc. The junction between a pair of features where the tangents are equal is not a vertex and will be accommodated by our technique. The current analysis is essentially concerned with modeling edges as straight-edge features. This technique works by estimating the edge orientation and then determining the direction perpendicular to that edge so that the 1-D operator can be applied in the correct direction to obtain subpixel accuracy.

Five parameters are used to model 2-D edges (Fig. 6). The first four parameters are the same as those for 1-D signals, and the fifth is the edge orientation parameter that specifies the angle that the edge makes with respect to the x-axis (θ).

A. Estimation of the Edge Orientation Parameter

The approximate edge position is computed by taking the union of the maxima of local energy in the horizontal and vertical directions [13]. The edge orientation parameter is then computed by rotating the local energy vector such that the magnitude of the perpendicular energy ($\hat{E}_p$) is maximized and as a consequence the tangential energy ($\hat{E}_t$) is minimized [15].

The edge orientation is then used to fit a straight line to the edge points within a neighborhood in a window of size w centered at the pixel of interest (in our method, we take w equal to the size of the filters). Fitting is performed using the least-squares line fitting method in windows of sizes $w-k$, $k = 0 \cdots w-3$. The line with the minimal variance is chosen to accurately represent the line parallel to the edge at the pixel. Let the line that gives minimal variance be

$$y_0(x) = mx + n. \tag{10}$$

Then, the line perpendicular to that in (10) is

$$y_1(x) = -(1/m)x + p. \tag{11}$$

These become the axes of a new coordinate frame, centered at the pixel (Fig. 7). The pixel values in the new grid are computed from the underlying pixel values by projecting the old pixel areas onto the new grid so that a new pixel is made up of proportions of areas of the old pixels.

The predicate-based algorithm is applied to the edge pixels in the new coordinate frame (in the direction θ) and each edge pixel labeled with its edge type. This process determines the initial parameters of the model that is used as the *a priori* information in the least-squared error fitting procedure. The least-squared error fitting procedure (as outlined in Section III) is then applied to an area centered on the edge pixel in the direction θ to get the best estimate of the edge parameters.

B. On the Analysis of the Algorithm

The local energy-based subpixel algorithm (detailed in [19]) has a complexity of order $O(n)$, where n is the number of edge pixels in

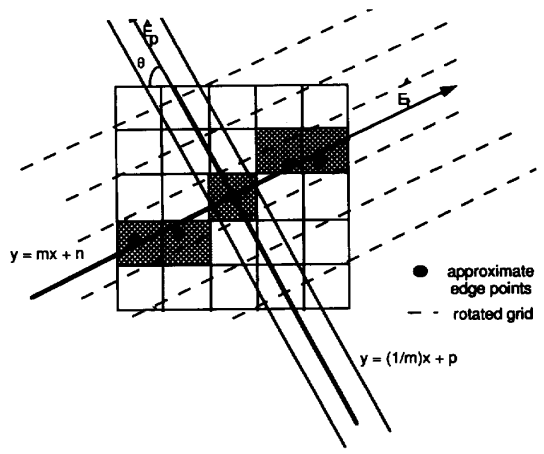

Fig. 7. Estimating the edge orientation.

the image. For a window of $m \times m$, the line fitting to compute the initial estimate of the edge orientation uses linear regression, and thus the complexity is $O(m)$. For the subsequent orientation refinement, the search space is in a circular window whose size depends upon the covariance of the line fitting performed in the initial orientation estimation; the bigger the covariance, the greater the number of possible angles to try. If the covariance is σ, the procedure is $O(\sigma)$. Similarly, the estimation of the localization parameters does not depend upon the number of input data. Thus the overall complexity of the algorithm is aproximately $O(n)$, because n is significantly larger that both m and σ. The covariance of the line fitting algorithm depends upon the SNR of the input image and the curvature of the edge boundary. In boundaries where curvature is high, such as the boundary of a circle, the covariance is higher than on boundaries that form straight lines. It has been noted from experiments that the time needed for the detection depends upon the quality of the input image. In cases where the SNR is low, the algorithm spends more time searching for the optimal orientation and computing the best fit model.

VI. Experimental Results

The techniques have been applied to real images with the results that edge parameters are extracted to subpixel accuracy. The results presented in Table III contain the results for the *edge_steady* and *edge_orientation* parameters of the step edges in the image data contained in the marked section in Fig. 8.

The *edge_orientation* θ is compared with that computed from a least-mean-squared fit of a straight line to the subpixel data, as this will give the most accurate estimate of the orientation given the assumption that the edge can be approximated by a straight line. The root mean squared error of the fit is 0.03 pixels, and the range is -0.079 to $+0.096$. The average value of θ is 57.76°, with a standard deviation of 0.078°.

The second test was performed to study the accuracy of the subpixel edge detector in detecting subpixel displacements. A straight-edge part of a precisely machined object was used. The object was located on a traveling stage capable of precisely displacing an object at 10-μm increments in the x direction. Image resolution was 100 μm. To obtain a high-SNR image, backlighting was used. A number of measurements were made for the object in its initial position and after moving 10 μm in the x direction. Averaging the two sets of results for edge location enabled statistics to be acquired for the variation in measurement due to image noise, etc. The statistical analysis of the measured values is presented in Table IV. The average positions are where the edge of the object was predicted to be, which is difficult to determine any other way. Note that it is difficult to calibrate a camera system to high accuracy because of lens errors, digitisation errors, etc. We argue that the most important statistic is the variance, which gives a measure of the variation in the determination of the edge position. In the table, the variances are small—of the order of 5 μm. The movement of the object by 10 μm results in a detected displacement of 0.08 pixels. Because the detected movement is larger than the variance, we can state that we can reliably detect subpixel movements (without averaging) to 10% accuracy.

TABLE III
Edge Parameters on a Real Image Data. For Simplicity, Only the *edge_steady* and *edge_orientation* Parameters are Presented.

x_steady	y_steady	θ	error
132.275	27.011	57.760	-0.007
131.577	27.450	57.769	-0.004
130.895	27.884	57.772	0.015
130.779	27.955	57.768	-0.051
130.683	28.018	57.767	0.052
130.660	28.033	57.765	0.069
130.530	28.117	57.780	0.039
130.498	28.136	57.771	-0.079
130.356	28.222	57.760	-0.069
130.305	28.257	57.761	0.096

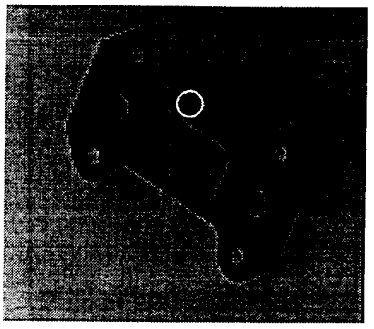

Fig. 8. The image of a machined object at 256 × 256 resolution.

TABLE IV
Detecting Subpixel Displacement. Second Location is Measured After the Object is Displaced Horizontally at 10 μm from the First Location.

First location		Second location	
Average	variance	Average	variance
120.011	0.04	120.092	0.05

VII. Discussion and Conclusion

The new edge feature extraction technique proposed allows better edge modeling than previous methods. The method is versatile in that edge parameters at subpixel accuracy can be extracted for all edge types. It does not suffer from the limitation of only being applicable to monotonic sequences or particular edge types. Further, the use of edge models enables the edge parameters to be localized robustly. The technique gives good results for real and synthetic images for both 1-D and 2-D edge types.

The proposed technique is similar to that of Perona *et al.* [18] in that their method also computes the energy of both noncomposite and composite edges by computing the energy function along the edge direction. Both methods cannot delineate the components of composite edge types, but they can localize them. However, their method involves the modeling of the local energy peak as an paraboloid to obtain subpixel level, whereas we determine the best edge model that describes the peaks correctly. While both methods can localize edges, our method has the advantage of describing all the underlying edge parameters besides localization at subpixel accuracy.

The task now is to extend the technique to accommodate different types of 2-D edges. Considering only straight edges is restrictive, and arcs, ellipses, and other curve types will be investigated. An important requirement is that the edge type must be known so the correct model can be used. Arbitrary curves will be investigated to determine if locally defining the curve as one of a number of known types allows subpixel accuracy to be determined.

References

[1] P. J. MacVicar-Whelan and T. O. Binford, "Line finding with subpixel precision," in *Proc. SPIE*, vol. 281, 1981.

[2] M. H. Hueckel, "A local visual operator which recognizes edges and lines," *J. Ass. Comput. Mach.*, vol. 20, Oct. 1973.

[3] R. Nevatia and K. Babu, "Linear feature extraction and description," *Comput. Graphics Image Processing*, vol. 13, 1980.

[4] A. J. Tabatabai and O. R. Mitchell, "Edge location to subpixel values in digital imagery," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 2, Mar 1984.

[5] A. Huertas and G. Medioni, "Detecion of intensity changes with subpixel accuracy using Laplacian-Gaussian masks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol PAMI-8, no. 5, Sept. 1986.

[6] E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves, "Subpixel measurements using a moment-based edge operator," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 12, Dec. 1989.

[7] A. Kundu, "Robust edge detection," *Pattern Recogn.*, vol. 23, 1990.

[8] D. Lee, "Coping with discontinuities in computer vision: Their detection, classification, and measurement," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 4, Apr. 1990.

[9] M. Kisworo, S. Venkatesh, and G. West, "Feature type identification using predicate-based algorithm," School of Computing, Curtin Univ., Perth, Western Australia, Tech. Rep. 5/91.

[10] M. C. Morrone and R. Owens, "Feature detection from local energy," *Pattern Recogn. Lett.*, vol. 6, 1987.

[11] S. Venkatesh, "A study of energy based models for the detection and classification of image features," Ph.D. dissertation, Dept. of Computer Science, Univ. of Western Australia, 1990.

[12] S. Venkatesh and R. Owens, "On the classification of image features," *Pattern Recogn. Lett.*, vol. 11, 1990.

[13] C. Ronse, "On the idempotence of the local energy detector," Phillips Labs, Belgium, Manuscript C235.

[14] S. A. Klein and D. M. Levi, "Hyperacuity threshold of 1 sec: Theoretical predictions and empirical validation," *J. Opt. Soc. Amer., Ser. A*, vol. 2, no. 7, 1985.

[15] M. Kisworo, S. Venkatesh, and G. West, "Two dimensional edge feature extraction using the generalized energy approach," in *Proc. IEEE TENCON* New Delhi, Aug. 1991.

[16] P. Perona, "Deformable kernels for early vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR-91*, Maui, HI, pp. 222–227, 1991.

[17] A. C. Bovik, M. Clark, and W. S. Geisler "Multi-channel texture analysis using localised spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 1, pp. 55–73, 1990.

[18] P. Perona and J. Malik, "Detecting and localising edges composed of steps, peaks and roofs," Univ. of California, Berkeley, Rep. UCB/CSD 90/590, Nov. 1990.

[19] M. Kisworo, S. Venkatesh, and G. West, "Modelling edges at subpixel accuracy using the local energy approach," School of Computing, Curtin Univ., Perth, Western Australia, Tech. Rep. 7/93.

Interpolation Using Wavelet Bases

Alex P. Pentland

***Abstract*—Efficient solutions to regularization problems can be obtained using orthogonal wavelet bases for preconditioning. Good approximate solutions can be obtained in only two or three iterations, with each iteration requiring only $O(n)$ operations and $O(n)$ storage locations. Two- and three-dimensional examples are shown using both synthetic and real range data.**

***Index Terms*— Regularization, wavelet transforms, surface interpolation, volume interpolation.**

I. Introduction

Interpolation is a common problem in computer vision. Perhaps the most well-known interpolation theory is regularization [6], [13], [3], [2], [18]. Using this approach, optimal RMS estimates of the surface can be obtained under the assumption that the surface can be characterized as a stationary Markov process. However, the most commonly used algorithms have the drawback that they require hundreds or even thousands of iterations to produce a smoothly interpolated surface. Thus there is a need for more efficient interpolation algorithms.

This paper will present efficient solutions to these interpolation problems based on the use of orthogonal wavelets as a preconditioning transform. Numerical examples using natural imagery will be shown. An implementation in C code, due to Sclaroff [14], is available by anonymous FTP from whitechapel.media.mit.edu in the file /u/ftp/misc/wavelet.reg.tar.Z.

A. Background: Regularization

Interpolation is typically formulated as the problem of constructing a piecewise-smooth surface given a sparse set of noisy measurements. Because the measurements are sparse, the problem is ill posed and requires adding a smoothing or *regularizing* term to obtain a solution in areas away from measured points.

Mathematically, the interpolation problem is most commonly formulated as finding a suitable function $\mathcal{U}$ that minimizes the sum of a smooth energy functional $\mathcal{K}(\mathcal{U})$ and a functional that depends upon boundary conditions $\mathcal{D}(\mathcal{U})$. By taking the variational derivative $\delta_{\mathcal{U}}$ of the energy functional and discretizing over a lattice of n nodes, the following matrix equation is obtained [13], [18]:

$$\lambda \mathbf{KU} + \mathbf{SU} - \mathbf{D} = 0. \tag{1}$$

In this equation, $\mathbf{U}$ is an $n \times 1$ vector of unknown displacements for each of n nodes, $\mathbf{K}$ is an $n \times n$ matrix called the *regularizing* or *smoothness* matrix, $\mathbf{D}$ is an $n \times 1$ vector whose nonzero entries are the measured sensor data, $\mathbf{S}$ is a diagonal selection matrix with ones at nodes with sensor measurements, and λ is a scalar constant that balances the relative influence of the data and regularization terms.

An interpolated surface $\mathbf{U}$ that solves (1) can be obtained by iterating a two-layer network with center-surround receptive fields [18]. Unfortunately, for typical problems several thousand iterations are required to obtain an interpolated surface; even if multiresolution

Manuscript received August 15, 1991; revised August 19, 1993. Recommended for acceptance by Associate Editor T. Boult.

The author is with the Perceptual Computing Section, The Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9214449.