COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING **29**, 143–167 (1985)

Topographic Classification of Digital Image Intensity Surfaces Using Generalized Splines and the Discrete Cosine Transformation*

LAYNE T. WATSON, THOMAS J. LAFFEY[†], AND ROBERT M. HARALICK

Departments of Computer Science and Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

Received February 12, 1983; revised March 27, 1984

A complete mathematical treatment is given for describing the topographic primal sketch of the underlying grey tone intensity surface of a digital image. Each picture element is independently classified into a unique descriptive label, invariant under monotonically increasing grey tone transformations, from the set {peak, pit, ridge, ravine, saddle, flat, and hillside}, with hillside having subcategories {inflection point, slope, convex hill, concave hill, and saddle hill}. The topographic classification is based on the first and second directional derivatives of the estimated image intensity surface. Two different sets of basis functions, generalized splines and the discrete cosine basis, are used to estimate the image intensity surface. Zero-crossings of the first directional derivative are identified as location of interest in the image. © 1985 Academic Press, Inc.

1.0. INTRODUCTION

Describing the fundamental structure of a digital image in a way which is invariant under monotonically increasing grey tone transformations is a primary problem encountered in any computer vision system. Information in an image is captured in the form of its light intensity variations. These light intensity variations must be used to explain the orientation and reflectance of the surfaces on the objects being imaged. We believe that the classification of the image intensity surface into a complete set of topographic types will be a definite aid in this regard.

Marr [16] argues that the first level of visual processing is the computation of a rich description of grey level changes present in an image and all subsequent computations are done in terms of this description which he calls the *primal sketch*. Grey level changes are usually associated with edges and Marr's primal sketch has for each area of grey level change a description which includes type, position, orientation, and fuzziness of edge. Marr [17] illustrates that it is possible from this information to reconstruct the image to a reasonable degree.

The topographic labeling presented in this paper is very much in the spirit of Marr's primal sketch. However, instead of concentrating on grey level changes as edges as Marr does, we concentrate on all types of 2-dimensional grey level variations. We consider each area on an image to be a spatial distribution of grey levels which constitute a surface or facet of grey tone intensities having a specific shape. It is intuitive that if the shape of the grey tone intensity surface could be described for each pixel, then by assembling all the shape fragments one could reconstruct in a relative way the entire surface of the image gray tone intensity values.

[*] This research has been supported by the National Science Foundation under grant MCS-8102872.

[†] Present address: Lockheed Palo Alto Research Labs, Palo Alto, CA 94304.

0734-189X/85 $3.00

A topographic pixel labeling does not describe the shape in an absolute way, but defines it only by general form. For example, if a surface has the shape of a peak, then we know that there is a point on the surface which is a relative maximum and all the points on the surface near the peak monotonically decrease away from the peak. Furthermore, we expect the hillsides around the peak to be concave down since we assume a surface having continuity in all derivatives.

Knowing that a pixel's surface has the shape of a peak does not tell us precisely where in the pixel the peak occurs and also does not tell us the height of the peak nor the magnitude of the slope around the peak. The topographic labeling, however, does satisfy Marr's [16] primal sketch requirement of containing a symbolic description of the grey tone intensity changes. Furthermore, upon computing and binding to each topographic label numerical descriptors such as gradient magnitude and direction, directions of the extrema of the second directional derivative along with their values, a reasonable absolute description of each shaped surface can be obtained.

The major topographic categories defined here, peak, pit, ridge, ravine, saddle, flat, and hillside, are invariant under monotonically increasing transformations of the grey tone values. This invariance is important, because the visual interpretation of an image is usually not changed by enhancing the image with monotonically increasing grey tone transformations. For proof of this invariance see Haralick, Watson, and Laffey [9].

1.1. Facet Model

The facet model states that all processing of digital image data has its final authoritative interpretation relative to what the processing does to the underlying grey tone intensity surface. The digital image pixel values are noisy sampled observations of the underlying surface. Thus, in order to do any processing, we at least have to estimate at each pixel position what this underlying surface is. This requires a model which describes what the general form of the surface would be in the neighborhood of any pixel if there were no noise. To estimate the surface from the neighborhood around a pixel then amounts to estimating the free parameters of the general form. It is important to note that if a different general form is assumed, then a different estimate of the surface is produced. Thus the assumption of a particular general form is necessary and has consequences.

The general forms we use in this paper are generalized quadratic B-splines and the classical discrete cosine basis. Having estimated this surface around each pixel, the first and second directional derivatives are easily computed by analytical means. The topographic classification of the underlying image surface is totally based on the first and second directional derivatives. We classify each surface point as peak, pit, ridge, ravine, saddle, flat, or hillside, with hillside being broken down further into the subcategories inflection point, convex hill, concave hill, saddle hill, and slope. Our set of topographic labels is complete in the sense that every combination of values of the first and second directional derivative is uniquely assigned to one of the classes.

1.2. Previous Work

Detection of topographic structures in a digital image is not a new idea. There have been a wide variety of techniques described to detect pits, peaks, ridges, ravines, and the like.

Peucker and Johnston [19] characterize the surface shape by the sequence of positive and negative differences as successive surrounding points are compared to the central point. Peucker and Douglas [20] describe several variation of this method for detecting one of the shapes from the set {pit, peak, pass, ridge, ravine, break, slope, flat}. They start with the most frequent feature (slope) and proceed to the less frequent, thus making it an order-dependent algorithm.

Johnston and Rosenfeld [12] attempt to find peaks by finding all points P such that no points in an n by n neighborhood surrounding P have greater elevation than that of P. Pits are found in an analogous manner. To find ridges, they identify points that are either east–west or north–south elevation maxima. This is done using a "smoothed" array in which each point is given the highest elevation in a 2×2 square containing it, and on this array also, east–west and north–south maxima are found. Ravines are found in a similar manner.

Paton [18] uses a six term quadratic expansion in Legendre polynomials fitted to a small disk around each pixel. The most significant coefficients of the second-order polynomial yield a descriptive label chosen from the set {constant, ridge, valley, peak, bowl, saddle, ambiguous}. He uses the continuous least squares fit formulation in setting up the surface fit equations as opposed to the discrete least squares fit used in the facet model. The continuous fit is a more expensive computation than the discrete fit and results in a step-like approximation.

Grender's [5] algorithm compares the grey level elevation of a central point with surrounding elevations at a given distance around the perimeter of a circular window and the radius of the window may be increased in successive passes through the image. His topographic labeling set consists of {slope, ridge, valley, knob, sink, saddle}.

Toriwaki and Fukumura [24] take a totally different approach from all the others. They use two local features of grey level pictures, connectivity number, and coefficient of curvature, for classification of the pixel into {peak, pit, ridge, ravine, hillside, pass}. They then describe how to extract structural information from the image once the labelings have been made. This structural information consists of ridge lines, ravine lines, and the like.

Hsu, Mundy, and Beaudet [10] use a quadratic surface approximation at every point on the image surface. The principal axes of the quadratic approximation are used as directions in which to segment the image. Lines emanating from the center pixel in these directions thus provide natural boundaries of patches approximating the surface. They then selectively generate the principal axes from some critical points distributed over an image and interconnect them into a network to get an approximation of the image data. In this network, which they call the *web representation*, the axes divide the image into regions, and show important features such as edges and peaks. They are then able to extract a set of primitive features from the nodes of the network by mask matching. Global features, such as ridge lines, are obtained by state transition rules.

Lee and Fu [15] define a set of 3×3 templates which they convolve over the image to give each class, except plain, a figure of merit. Their set of labels include {none, plain, slope, ridge, valley, foot, shoulder}. Thresholds are used to determine into which class the pixel will fall. In their scheme a pixel may satisfy the definition of zero, one, or more than one class. Ambiguity is resolved by choosing the class with the highest figure of merit.

1.3. A Mathematical Approach

From the previous discussion one can see that a wide variety of methods and labels were proposed to describe the topographical structure in a digital image. Some of the methods require multiple passes through the image while others may give ambiguous labels to a pixel. Many of the methods are heuristic in nature. The Hsu, Mundy, and Beudet [10] approach is the most similar to the one discussed here.

Our classification approach is based on the estimation of the first and second-order directional derivatives. Thus, we regard the digital picture function as a sampling of the underlying function f, where some kind of random noise is added to the true function values. To estimate the first and second partials we must assume some kind of parametric form for the underlying function f. The classifier must use the sampled brightness values of the digital picture function to estimate the parameters, and then make decisions regarding the locations of relative extrema of partial derivatives based on the estimated values of the parameters.

In Section 2 we discuss the mathematical properties of the topographic structures in terms of the directional derivatives in the continuous surface domain. Because a digital image is a sampled surface and each pixel has an area associated with it, characteristic topographic structures may occur anywhere within a pixel's area. Thus the implementation of the mathematical topographic definitions is not entirely trivial. In Section 3 we discuss the implementation of the classification scheme on a digital image. To identify those categories which are local 1-dimensional extrema, such as peak, pit, ridge, ravine, and saddle, we search inside the pixel's area for a zero-crossing of the first directional derivative. The directions in which we seek the zero-crossing are along the lines of extreme curvature. In Section 4 we discuss two possible surface estimation schemes: generalized splines and the discrete cosine transformation. In Section 5 we summarize the algorithm for topographic classification using the surface estimation techniques discussed in Section 4. In Section 6 we show the results of the classifier on several test images.

2.0. THE MATHEMATICAL CLASSIFICATION OF TOPOGRAPHIC STRUCTURES

In this section we formulate our notion of topographic structures on continuous surfaces. For completeness we first describe the quantities we need to perform our calculations, then the definitions of the topographic labels, and finally we summarize the mathematical properties of each of our topographic categories.

2.1. The Mathematical Properties

We will use the following notation to describe the mathematical properties of our various topographical categories for continuous surfaces. Let

∇f = gradient vector of a function f

$\|\nabla f\|$ = gradient magnitude

$\omega^{(1)}$ = direction in which second directional derivative has greatest magnitude

$\omega^{(2)}$ = direction orthogonal to $\omega^{(1)}$

λ_1 = value of second directional derivative in the direction of $\omega^{(1)}$

λ_2 = value of second directional derivative in the direction of $\omega^{(2)}$

$$\nabla f \cdot \omega^{(1)} = \text{value of first directional derivative in the direction of } \omega^{(1)}$$
$$\nabla f \cdot \omega^{(2)} = \text{value of first directional derivative in the direction of } \omega^{(2)}$$

Without loss of generality, we assume $|\lambda_1| \geq |\lambda_2|$.

Each type of topographic structure in our classification scheme is defined in terms of the above quantities. In order to calculate these values, the first and second-order partials with respect to r and c need to be approximated. These five partials are

$$\partial f/\partial r, \qquad \partial f/\partial c, \qquad \partial^2 f/\partial r^2, \qquad \partial^2 f/\partial c^2, \qquad \partial^2 f/\partial r\, \partial c.$$

The gradient vector is simply $(\partial f/\partial r, \partial f/\partial c)$. The second directional derivatives may be calculated by forming the *Hessian*, where the Hessian is a 2 × 2 matrix defined as:

$$H = \begin{vmatrix} \partial^2 f/\partial r^2 & \partial^2 f/\partial r\, \partial c \\ \partial^2 f/\partial c\, \partial r & \partial^2 f/\partial c^2 \end{vmatrix}.$$

Hessian matrices are used extensively in nonlinear programming. The eigenvalues of the Hessian are the values of the extrema of the second directional derivative, and their associated eigenvectors are the directions in which the second directional derivative is extremized. Furthermore, the two directions represented by the eigenvectors are orthogonal to one another. Since H is a 2 × 2 symmetric matrix, calculation of the eigenvalues and eigenvectors can be done efficiently and accurately using the method of Rutishauser [21]. We may obtain the values of the first directional derivative in the direction of either extrema of the second directional derivative by simply taking the dot product of the gradient with the appropriate eigenvector

$$\nabla f \cdot \omega^{(1)}$$
$$\nabla f \cdot \omega^{(2)}.$$

There is a direct relationship between the eigenvalues λ_1 and λ_2 and curvature in the directions $\omega^{(1)}$ and $\omega^{(2)}$: When the first directional derivative $\nu f \cdot \omega^{(i)} = 0$, then $\lambda_i/(1 + (\nabla f)^2)^{1/2}$ is the curvature in the direction $\omega^{(i)}$, $i = 1$ or 2.

2.2. The Topographic Categories

Having the gradient magnitude and direction, and the eigenvalues and eigenvectors of the Hessian, we now describe the topographic classification algorithm.

2.2.1. Peak and Pit

A peak (knob) occurs where there is a local maximum in all directions. In other words, we are on a peak if no matter what direction we look in, we see no point that is as high as the one we are on. At a peak the gradient is zero and the second directional derivative is negative in all directions. To test whether the second directional derivative is negative in all directions, we just have to examine the value of the second directional derivative in the directions which make it smallest and largest. A point is therefore classified as a peak if it satisfies the conditions

$$\|\nabla f\| = 0, \qquad \lambda_1 < 0, \qquad \lambda_2 < 0.$$

A pit (sink, bowl) is identical to a peak except it is a local minimum in all directions rather than a local maximum. A point is classified as a pit if it satisfies the conditions

$$\|\nabla f\| = 0, \qquad \lambda_1 > 0, \qquad \lambda_2 > 0.$$

2.2.2. *Ridge and Ravine*

A ridge occurs on a ridge line, a curve consisting of a series of ridge points. As we walk along the ridge line the points to the right and left of us are lower than the ones we are on. Furthermore, the ridge line may be flat, sloped upward, sloped downward, curving upward, or curving downward. A ridge occurs where there is a local maximum in one direction. It, therefore, must have negative second directional derivatives in the direction across the ridge and also a zero first directional derivative in that same direction. The direction in which the local maximum occurs may correspond to either of the directions in which the curvature is extremized, since the ridge itself may be curved. For nonflat ridges, this leads to the first two cases below for ridge characterization. If the ridge is flat then the ridge line is horizontal and the gradient is zero along it. This corresponds to the third case. A point is therefore classified as a ridge if it satisfies any one of three sets of conditions

$$\begin{gathered} \|\nabla f\| \neq 0, \qquad \lambda_1 < 0, \qquad \nabla f \cdot \omega^{(1)} = 0, \\ \text{or} \\ \|\nabla f\| \neq 0, \qquad \lambda_2 < 0, \qquad \nabla f \cdot \omega^{(2)} = 0, \\ \text{or} \\ \|\nabla f\| = 0, \qquad \lambda_1 < 0, \qquad \lambda_2 = 0. \end{gathered}$$

This definition for ridge can lead to possibly unexpected results. For example, all points on a right circular cone, except the vertex, will be labeled ridge. Whether one wishes to call these points ridge points or something else is a matter of taste. These points are classified as ridge points because as one walks up the cone toward the vertex the points to the right and left are lower than the one you are on. This continuum of ridges may or may not be acceptable depending on your viewpoint. Since this classification scheme is complete, unambiguous, and generally coincides with intuition, we prefer to accept the continuum of ridges.

A ravine (valley) is identical to a ridge except it is a local minimum in one direction rather than maximum. A point is classified as a ravine if it satisfies any one of three sets of conditions

$$\begin{gathered} \|\nabla f\| \neq 0, \qquad \lambda_1 > 0, \qquad \nabla f \cdot \omega^{(1)} = 0; \\ \text{or} \\ \|\nabla f\| \neq 0, \qquad \lambda_2 > 0, \qquad \nabla f \cdot \omega^{(2)} = 0, \\ \text{or} \\ \|\nabla f\| = 0, \qquad \lambda_1 > 0, \qquad \lambda_2 = 0. \end{gathered}$$

2.2.3. *Saddle*

A saddle occurs where there is a local maximum in one direction and a local minimum in a perpendicular direction. It, therefore, must have positive curvature in one direction and negative curvature in a perpendicular direction. At a saddle the gradient magnitude must be zero and the extrema of the second directional derivative must have opposite signs. A point is classified as a saddle if it satisfies the conditions

$$\|\nabla f\| = 0, \qquad \lambda_1 * \lambda_2 < 0.$$

2.2.4. *Flat*

A flat (plain) is a simple, horizontal surface. It, therefore, must have zero gradient and no curvature. A point is classified as a flat if it satisfies the conditions

$$\|\nabla f\| = 0, \qquad \lambda_1 = 0, \qquad \lambda_2 = 0.$$

2.2.5. *Hillside*

A hillside point is anything not covered by the previous categories. It has a nonzero gradient and no strict extrema in the direction of maximum and minimum second directional derivative. If the hill is simply a tilted flat (i.e., has constant gradient), we call it a *slope*. If its curvature is positive (upward), we call it a *convex hill*. If its curvature is negative (downward), we call it a *concave hill*. If the curvature is up in one direction and down in a perpendicular direction, we call it a *saddle hill*.

A point on a hillside is an *inflection point* if it has a zero-crossing of the second directional derivative taken in the direction of the gradient. The inflection point class is the same as the *step edge* as defined by Haralick [8] who classifies a pixel as a step edge if there is some point in the pixel's area having a zero-crossing of the second directional derivative taken in the direction of the gradient.

To determine the conditions for a point to be a hillside, we just take the complement of the disjunction of the conditions given for all the previous classes. Thus if there is no curvature, then the gradient must be nonzero. If there is curvature, then the point must not be a relative extremum. Therefore, a point is classified as a hillside if all three sets of the following conditions are true ("→" represents the operation of logical implication):

$$\lambda_1 = \lambda_2 = 0 \rightarrow \|\nabla f\| \neq 0$$

and

$$\lambda_1 \neq 0 \rightarrow \nabla f \cdot \omega^{(1)} \neq 0$$

and

$$\lambda_2 \neq 0 \rightarrow \nabla f \cdot \omega^{(2)} \neq 0.$$

We can differentiate between different classes of hillsides by the values of the second directional derivative. The distinction can be made as follows:

SLOPE	if $\lambda_1 = \lambda_2 = 0$,
CONVEX	if $\lambda_1 \geq \lambda_2 \geq 0$, $\lambda_1 \neq 0$,
CONCAVE	if $\lambda_1 \leq \lambda_2 \leq 0$, $\lambda_1 \neq 0$,
SADDLE HILL	if $\lambda_1 * \lambda_2 < 0$.

A slope, convex, concave, or saddle hill is classified as an INFLECTION POINT if there is a zero-crossing of the second directional derivative in the direction of maximum first directional derivative (i.e., the gradient).

2.3. *Summary of the Topographic Categories*

A summary of the mathematical properties of our topographic structures on continuous surfaces can be found in Table 1. The table exhaustively defines the topographic classes by their gradient magnitude, second directional derivative extrema values, and the first directional derivatives taken in the directions which extremize second directional derivative. Each entry in the table is either 0, +, −, or *. The "0" means not significantly different from zero, "+" means significantly different from zero on the positive side, "−" means significantly different from zero on the negative side, and "*" means that it does not matter. The label "Cannot Occur" means that it is impossible for the gradient to be nonzero and the first directional derivative to be zero in two orthogonal directions.

From Table 1 one can see that our classification scheme is complete. All possible combinations of first and second directional derivatives have a corresponding entry in the table. Each topographic category has a set of mathematical properties which uniquely determines it.

(Note: Special attention is required for the degenerate case $\lambda_1 = \lambda_2 \neq 0$, where $\omega^{(1)}$ and $\omega^{(2)}$ can be *any* two orthogonal directions. In this case, there *always* exists an extreme direction ω which is orthogonal to ∇f, and thus the first directional derivative $\nabla f \cdot \omega$ is *always* zero in an extreme direction. To avoid spurious zero directional derivatives, choose $\omega^{(1)}$ and $\omega^{(2)}$ such that $\nabla f \cdot \omega^{(1)} \neq 0$ and $\nabla f \cdot \omega^{(2)} \neq 0$, unless the gradient is zero.)

TABLE 1
Mathematical Properties

$\|\nabla f\|$	λ_1	λ_2	$\nabla f \cdot \omega^{(1)}$	$\nabla f \cdot \omega^{(w)}$	Label
0	−	−	0	0	Peak
0	−	0	0	0	Ridge
0	−	+	0	0	Saddle
0	0	0	0	0	Flat
0	+	−	0	0	Saddle
0	+	0	0	0	Ravine
0	+	+	0	0	Pit
+	−	−	−, +	−, +	Hillside
+	−	*	0	*	Ridge
+	*	−	*	0	Ridge
+	−	0	−, +	*	Hillside
+	−	+	−, +	−, +	Hillside
+	0	0	*	*	Hillside
+	+	−	−, +	−, +	Hillside
+	+	0	−, +	*	Hillside
+	+	*	0	*	Ravine
+	*	+	*	0	Ravine
+	+	+	−, +	−, +	Hillside
+	*	*	0	0	Cannot Occur

3.0. THE TOPOGRAPHIC CLASSIFICATION ALGORITHM

The definitions of Section 2 cannot be used directly since there is a problem of where in a pixel's area to apply the classification. If the classification were only applied to the point at the center of each pixel, then a pixel having a peak near one of its corners, for example, would get classified as a concave hill rather than as a peak. The problem is that the topographic classification we are interested in must be a sampling of the actual topographic surface classes. Most likely, the interesting categories of peak, pit, ridge, ravine, and saddle will never occur precisely at a pixel's center, and if they do occur in a pixel's area, then the pixel must carry that label rather than the class label of the pixel's center point. Thus one problem we must solve is to determine the dominant label for a pixel given the topographic class label of every point in the pixel. The next problem we must solve is to determine, in effect, the set of all topographic classes occurring within a pixel's area without having to do the impossible brute force computation.

For the purpose of solving these problems we divide the set of topographic labels into two subsets: (1) those that indicate a strict local 1-dimensional extremum has occurred {peak, pit, ridge, ravine, and saddle}, and (2) those that do not indicate a strict local 1-dimensional extremum has occurred {flat and hillside}. By 1-dimensional we mean along a line (in a particular direction). A strict local 1-dimensional extremum can be located by finding those points within a pixel's area where a zero-crossing of the first directional derivative occurs. The actual location of the zero-crossing is discussed in Section 4.

So that we do not search the pixel's entire area for the zero-crossing, we only search in the directions of extreme second directional derivatives $\omega^{(1)}$ and $\omega^{(2)}$. Since these directions are well aligned with curvature properties, the change of overlooking important topographic structure is minimized, and, more importantly, the computational cost is small.

When $\lambda_1 = \lambda_2 \neq 0$, the directions $\omega^{(1)}$ and $\omega^{(2)}$ are not uniquely defined. We handle this case by searching for a zero-crossing in the direction given by $H^{-1} * \nabla f$. This is the *Newton direction* and it points directly toward the extremum of a quadratic surface.

For inflection point location (first derivative extremum), we search along the gradient direction for a zero-crossing of second directional derivative. For 1-dimensional extrema, there are four cases to consider: no zero-crossing, one zero-crossing, two zero-crossings, and more than two zero-crossings. The next four subsections discuss these cases.

3.1. Case 1: No Zero-Crossing

If no zero-crossing is found along either of the two extreme directions within the pixel's area, then the pixel cannot be a local extremum, and therefore must be assigned a label from the set {flat or hillside}. If the gradient is zero, we have a flat. If it is nonzero, we have a hillside. If the pixel is a hillside we classify it further into {inflection point, slope, convex hill, concave hill, or saddle hill}. If there is a zero-crossing of the second directional derivative in the direction of the gradient within the pixel's area, the pixel is classified as an inflection point. If no such zero-crossing occurs, the label assigned to the pixel is based on the gradient magnitude and Hessian eigenvalues calculated at the center of the pixel, local coordinates (0, 0), as in Table 2.

TABLE 2

$\|\nabla f\|$	λ_1	λ_2	Label
0	0	0	Flat
+	−	−	Concave Hill
+	−	0	Concave Hill
+	−	+	Saddle Hill
+	0	0	Slope
+	+	−	Saddle Hill
+	+	0	Convex Hill
+	+	+	Convex Hill

3.2. *Case 2: One Zero-Crossing*

If a zero-crossing of the first directional derivative is found within the pixel's area, then the pixel is a strict local 1-dimensional extremum and must be assigned a label from the set {peak, pit, ridge, ravine, or saddle}. At the location of the zero-crossing, the Hessian and gradient are recomputed and if the gradient magnitude at the zero-crossing is zero, Table 3 is used.

If the gradient magnitude is nonzero then the choice is either ridge or ravine. If the second directional derivative in the direction of the zero-crossing is negative, we have a ridge. If it is positive, we have a ravine. If it is zero, we compare the function value at the center of the pixel $f(0,0)$, with the function value at the zero-crossing $f(r,c)$. If $f(r,c)$ is greater than $f(0,0)$, we call it a ridge, otherwise we call it a ravine.

3.3. *Case 3: Two Zero-Crossings*

If we have two zero crossings of the first directional derivative, one in each direction of extreme curvature, then the Hessian and gradient must be recomputed at each zero-crossing. Using the procedure described in Section 3.2 we assign a label to each zero-crossing. We call these labels LABEL1 and LABEL2. The final classification given the pixel is based on these two labels and is given in Table 4.

If both labels are identical, the pixel is given that label. In the case of both labels being ridge, the pixel may actually be a peak, but experiments have shown that this case is rare. An analogous argument can be made for both labels being ravine. If one label is ridge and the other ravine, this indicates we are at, or very near to, a saddle point, and thus the pixel is classified as a saddle. If one label is peak and the other ridge, we choose the category giving us the "most information," which in this case is

TABLE 3

$\|\nabla f\|$	λ_1	λ_2	Label
0	−	−	Peak
0	−	0	Ridge
0	−	+	Saddle
0	+	−	Saddle
0	+	0	Ravine
0	+	+	Pit

TABLE 4

LABEL1	LABEL2	Resulting Label
Peak	Peak	Peak
Peak	Ridge	Peak
Pit	Pit	Pit
Pit	Ravine	Pit
Saddle	Saddle	Saddle
Ridge	Ridge	Ridge
Ridge	Ravine	Saddle
Ridge	Saddle	Saddle
Ravine	Ravine	Ravine
Ravine	Saddle	Saddle

peak. The peak is a local maximum in all directions while the ridge is a local maximum in only one direction. Thus peak conveys more information about the image surface. An analogous argument can be made if the labels are pit and ravine. Similarly, a saddle gives us more information than a ridge or valley. Thus a pixel is assigned saddle if its zero-crossings have been labeled ridge and saddle, or ravine and saddle.

It is apparent from the table that not all possible label combinations are accounted for. Some combinations, such as PEAK and PIT, are omitted because of the assumption that the underlying surface is smooth and sampled frequently enough that a peak and pit will not both occur within the same pixel's area. If such a case occurs, our convention is to arbitrarily choose one of LABEL1 or LABEL2 as the resulting label for the pixel.

3.4. Case 4: More Than Two Zero-Crossings

If more than two zero-crossings occur within a pixel's area, then in at least one of the extrema directions there are two zero-crossings. If this happens we choose the zero-crossing closest to the pixel's center and ignore the other. If we ignore the further zero-crossings, then this case is identical to Case 3. This situation has yet to occur in the experiments we have run.

3.5. Threshold Selection

In order to determine whether the gradient magnitude is nonzero or not, we use a *mixed error* criterion. The thresholds are based on an absolute error (ABSERR) and a relative error. The calculation of the gradient threshold, $\mathrm{G_T}$, is done as follows:

$$\mathrm{G_T} = \mathrm{Max}\{\mathrm{ABSERR}, \mathrm{ABSERR} * f(r,c)\},$$

where $f(r,c)$ is the estimated image grey tone at (r,c). The calculation of the eigenvalue threshold λ_T is

$$\lambda_\mathrm{T} = \mathrm{Max}\{\mathrm{ABSERR}, \mathrm{ABSERR} * \|H\|_\infty\},$$

where

$$\|H\|_\infty = \mathrm{Max}\{|H_{11}| + |H_{12}|, |H_{21}| + |H_{22}|\}.$$

$\|H\|_\infty$ is called the *max norm* of the Hessian matrix. The choice of ABSERR is a user input parameter based on the accuracy of the calculations. We use 10^{-4} for our value of ABSERR.

4.0. SURFACE ESTIMATION

In this section we discuss the estimation of the parameters required by the topographic classification scheme of Section 2.0 using a new approach based on *generalized splines*, and a classic approach, *the discrete cosine transformation*. It is important to note that the classification scheme of Section 2 and the algorithm of Section 3 are both independent of the method used to estimate the first and second order partials of the underlying digital image intensity surface at each sampled point. We now describe the two different approaches in detail.

4.1. Generalized Spline Approach

Considerable computational experience with polynomial facet models of digital image surfaces has been obtained by Haralick [6] and others. Polynomials are convenient to manipulate, but their degree of approximation is known to be comparatively low. Cubic polynomials (represented in terms of discretely orthogonal polynomials) have been used successfully for edge detection [8] and topographic classification [9]. However, there are common image surfaces which are not polynomial-like over reasonably sized windows (every smooth surface is polynomial-like in a neighborhood of a point, but the neighborhood may be very small). The motivation for the present work was the need for a set of basis functions for surface approximation with the following properties:

(1) They can exactly replicate quadratic surfaces, which include such common image surface features as peak, pit, flat, slope, saddle, convex hill, concave hill, and saddle hill.

(2) They can approximate well other common surface features such as step edge (inflection point), ridge, ravine, and curved ridge or ravine.

(3) They can be computed efficiently.

(4) They are well conditioned, so the computation will be numerically stable.

In order to estimate the required partial derivatives we perform a weighted least squares fit with a 2-dimensional surface f in a neighborhood of each pixel. It is required that the function f be continuous throughout the neighborhood and have continuous first and second-order partial derivatives with respect to r and c at the center pixel. The basis functions comprised of tensor products of 1-dimensional generalized splines, described below, have all of the properties discussed.

Consider a 9×9 window centered at the origin with pixel coordinates at the integer lattice points. Using the notation of deBoor [3] the knot sequence

$$t = (-4.5, -4.5, -4.5, -1, -1, 1, 1, 4.5, 4.5, 4.5)$$

defines 7 independent generalized B-splines of order 3,

$$B_1(x), B_2(x), \ldots, B_7(x).$$

The $B_i(x)$ are continuous piecewise quadratic functions with breakpoints at -4.5, -1, 1, and 4.5. The crucial characteristic is that the B_i are continuous but not continuously differentiable at the interior breakpoints -1 and 1. The 2-dimensional basis functions are then given by

$$\phi_{ij}(r,c) = B_i(r) \otimes B_j(c), \qquad i,j = 1,2,\ldots,7,$$

where $\otimes$ is the symbol for tensor product. There are 49 basis functions, providing 49 degrees of freedom, to fit the 81 pixel values in the 9×9 window. The inner product with respect to which the best weighted least squares approximation is computed is

$$\langle f, g \rangle = \sum_{r,c} w_{rc} * f(r,c) * g(r,c),$$

with both r and c being summed from -4 to $+4$, and the weights w_{rc} for $\max\{|r|, |c|\} = 0,1,2,3,4$ are 1, 1, e^{-1}, e^{-2}, and e^{-3}, respectively. These weights require some explanation. Since the purpose of the fit is to understand the nature of the underlying digital image intensity surface at 0, pixels near 0 are more significant than those far away. Thus the weights decrease exponentially away from 0. The apparent anomaly is that the weight at 0 is relatively small, reflecting that *consistency* at 0 is more important than the actual measured value at 0. Intuitively, the effect of this inner product is to use an approximation over a 3×3 window, somewhat modified by a 5×5 window approximation, and very slightly modified by 7×7 and 9×9 window approximations.

The needed partials, evaluated at local coordinates (r, c), can be calculated

$$\begin{aligned} \partial f/\partial r &= \sum \phi_{ij} * B_i'(r) * B_j(c) \\ \partial f/\partial c &= \sum \phi_{ij} * B_i(r) * B_j'(c) \\ \partial^2 f/\partial r^2 &= \sum \phi_{ij} * B_i''(r) * B_j(c) \\ \partial^2 f/\partial c^2 &= \sum \phi_{ij} * B_i(r) * B_j''(c) \\ \partial^2 f/\partial r\,\partial c &= \sum \phi_{ij} * B_i'(r) * B_j'(c). \end{aligned}$$

The summations are done over both i and j, going from 3 to 5. The summation is done only from 3 to 5 because none of the other basis functions affect the calculation when we are in the interval $|r| \leq 1$, $|c| \leq 1$ due to the placement of the knots at -1 and 1.

To find a zero-crossing of the first directional derivative within a pixel's area using the generalized spline approach is more complex than locating them using the local polynomial facet model. An iterative root-finding technique must be used since there is no simple closed form solution. In order to locate the zero-crossing in the direction β, we constrain r and c by

$$r = \rho * \sin\beta \qquad \text{and} \qquad c = \rho * \cos\beta.$$

Since the direction β is known, locating the zero-crossing amounts to finding the value ρ, where $f_\beta'(\rho \sin\beta, \rho\cos\beta) = 0$. The root finding can be done efficiently and stably using the method of Shampine and Allen [22], which is a combination of bisection and the secant rule. The code we used is a modification of the code ZEROIN, which is completely explained and documented in the text by Shampine and Allen. Finding a zero-crossing of the second directional derivative can be done in an analogous manner.

4.2. Discrete Cosine Transformation

A very popular set of basis functions used in image processing is the *discrete cosine basis*. It was known to Chebyshev [2], and applied to image processing by Ahmed, Natarajan, and Rao [1]. Efficient computation of the 2-D discrete cosine transformation is described in Kamanger and Rao [13]. The 2-dimensional basis functions are tensor products of 1-dimensional basis functions, and are orthogonal with respect to a discrete inner product.

On the interval [0, L] take the points

$$x_j = \frac{(2 * j + 1) * L}{2 * (n + 1)}, \qquad j = 0, 1, \ldots, n,$$

and the discrete inner product

$$\langle f, g\rangle = \sum f(x_j) * g(x_j),$$

with the summation going over j from 0 to n. Then the cosine basis functions

$$1, \cos(\pi x/L), \cos(2\pi x/L), \ldots, \cos(n\pi x/L)$$

are orthogonal with respect to the above inner product [11]. On the square [0, L] × [0, L] the appropriate inner product is

$$\langle f, g\rangle = \sum f(x_i, y_j) * g(x_i, y_j)$$

with respect to which the tensor products

$$\phi_{r,s} = \cos(r\pi x/L) \otimes \cos(s\pi y/L)$$

are also orthogonal. When the basis functions involved are orthogonal, solving the normal equations for a least squares approximation reduces to the trivial problem of solving a diagonal linear system of equations. The function

$$\sum \alpha_{r,s} * \phi_{r,c}$$

which minimizes

$$\left\langle \sum \alpha_{r,s}\phi_{r,s} - g, \sum \alpha_{r,s}\phi_{r,s} - g \right\rangle = \sum \left[\sum \alpha_{r,s}\phi_{r,s}(x_i, y_j) - g(x_i, y_j) \right]^2,$$

where $g(x_i, y_j)$ is the image grey tone at pixel coordinates (x_i, y_j), is given simply by

$$\phi^* = \sum \alpha^*_{r,s} * \phi_{r,s},$$
$$\alpha^*_{r,s} = \frac{\langle g, \phi_{r,s}\rangle}{\langle \phi_{r,s}, \phi_{r,s}\rangle}.$$

For the present situation, $n = 8$, $x_j = -4.5 + (2j + 1)/2$, and $j = 0, 1, 2, \dots, n$;

$$\phi_{k,m}(x, y) = \cos(k\pi(x + 4.5)/9) \otimes \cos(m\pi(y + 4.5)/9),$$

where $k, m = 0, 1, \dots, 6$.

The calculation of the derivatives and the location of the zero-crossings are done in the same way as for the generalized splines. Note that (if 49 basis functions are used) all 49 terms must be used since *all* of the basis functions have support within the center pixel's area. This makes the zero-crossing calculations very expensive compared to the spline or polynomial basis.

5.0. SUMMARY OF THE TOPOGRAPHIC CLASSIFICATION SCHEME

The scheme is a parallel process for topographic classification of every pixel which can be done in one pass through the image. At each pixel of the image the following four steps need to be performed:

(1) Calculate the fitting coefficients of a 2-dimensional surface using the chosen basis functions in an n by n neighborhood around the pixel. (*Note*: The neighborhood size is not arbitrary. It is as small as possible to permit a least squares fit.)

(2) Use the coefficients calculated in step 1 to find the gradient, gradient magnitude, and the eigenvalues and eigenvectors of the Hessian at the center of the pixel's neighborhood (0, 0).

(3) Search in the direction of the eigenvectors calculated in step 2 for a zero-crossing of the first directional derivative within the pixel's area. (If the eigenvalues of the Hessian are equal and nonzero, then search in the Newton direction).

(4) Recompute the gradient, gradient magnitude, and values of second directional derivative extrema at each zero-crossing. Then apply the labeling scheme as described in Subsections 3.1, 3.2, 3.3, and 3.4.

6.0. RESULTS

In this section we show the results of the topographic primal sketch on several test images, comparing the results obtained from the different surface estimation schemes.

6.1. Comparison of Residual Error

The root-mean squared error (RMS) of a least squares fit f is defined as

$$\mathrm{RMS} \overset{N}{\underset{r,c}{=}} \left[\sum (g(r,c) - f(r,c))^2 \Big/ N\right]^{1/2},$$

where $g(r, c)$ = observed image grey tone at (r, c),

$f(r, c)$ = estimated image grey tone at (r, c), and

N = numer of data points.

TABLE 5

Image	RMS of Spline	RMS of DCT
Chair	2.398	2.756
Machine Parts	2.560	2.938
Two cells	0.686	0.800
Surface of Revolution	0.323	0.448
Step Edge	0.771	0.945

The RMS can be thought of as the average error between the estimated and the observed image grey tone at each pixel. In Table 5 we show an *informal* comparison of the RMS of the generalized splines vs. the discrete cosines. For the DCT in this comparison we use 49 basis functions, the same as the generalized spline. The RMS reported here is the average RMS of each 9 × 9 neighborhood in the specified image. The images used were the chair (Fig. 1), machine parts (Fig. 2), two cells (Fig. 3), the surface of revolution (Fig. 4a), and a 1-dimensional step edge consisting of a flat, a ramp with slope 20, and then another flat. The first three images in the table are real images and the last two were generated. In all cases, the spline fit had lower residual error than the DCT.

In Table 6 the DCT is compared with the local cubic facet fit [14]. Again a 9 × 9 neighborhood was used for both techniques. For the DCT only 10 basis functions

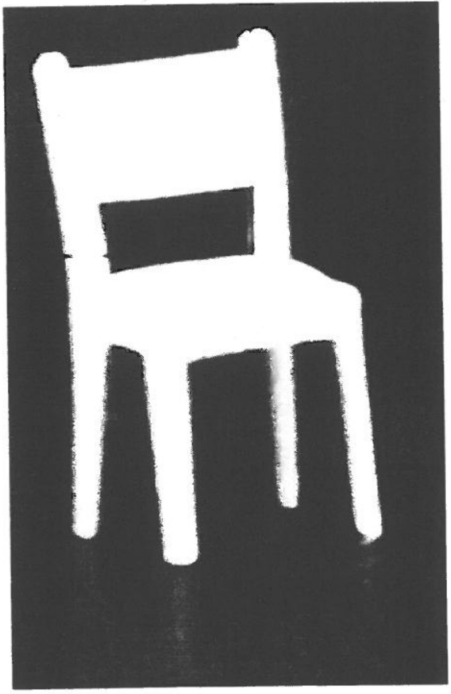

FIG. 1. Chair image.

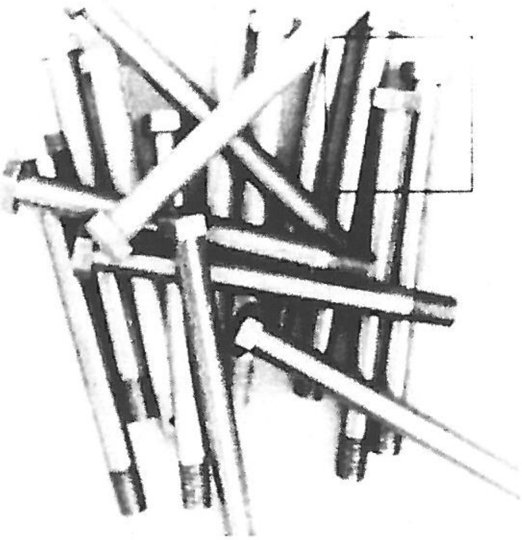

FIG. 2. Machine parts image.

($\phi_{k,m}$ with $k + m \leq 3$) were used so that it would compare directly to the cubic facet. The DCT has lower residual error on all images except the surface of revolution. In Table 7 we show the results of the spline, DCT with 10 coefficients, DCT with 49 coefficients, and the local cubic facet (10 coefficients). The RMS of the spline is only about $\frac{1}{3}$ of that of the DCT (with 10 coefficients) or the cubic facet. It is fair to compare the spline fit with the 10 coefficient DCT fit and the cubic fit because we actually use only 9 coefficients when doing the topographic labeling with the generalized spline method. The reason why only 9 coefficients are needed is discussed in Subsection 4.1.

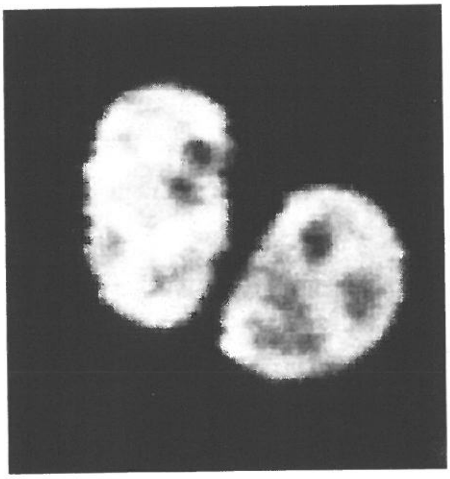

FIG. 3. Two cells image.

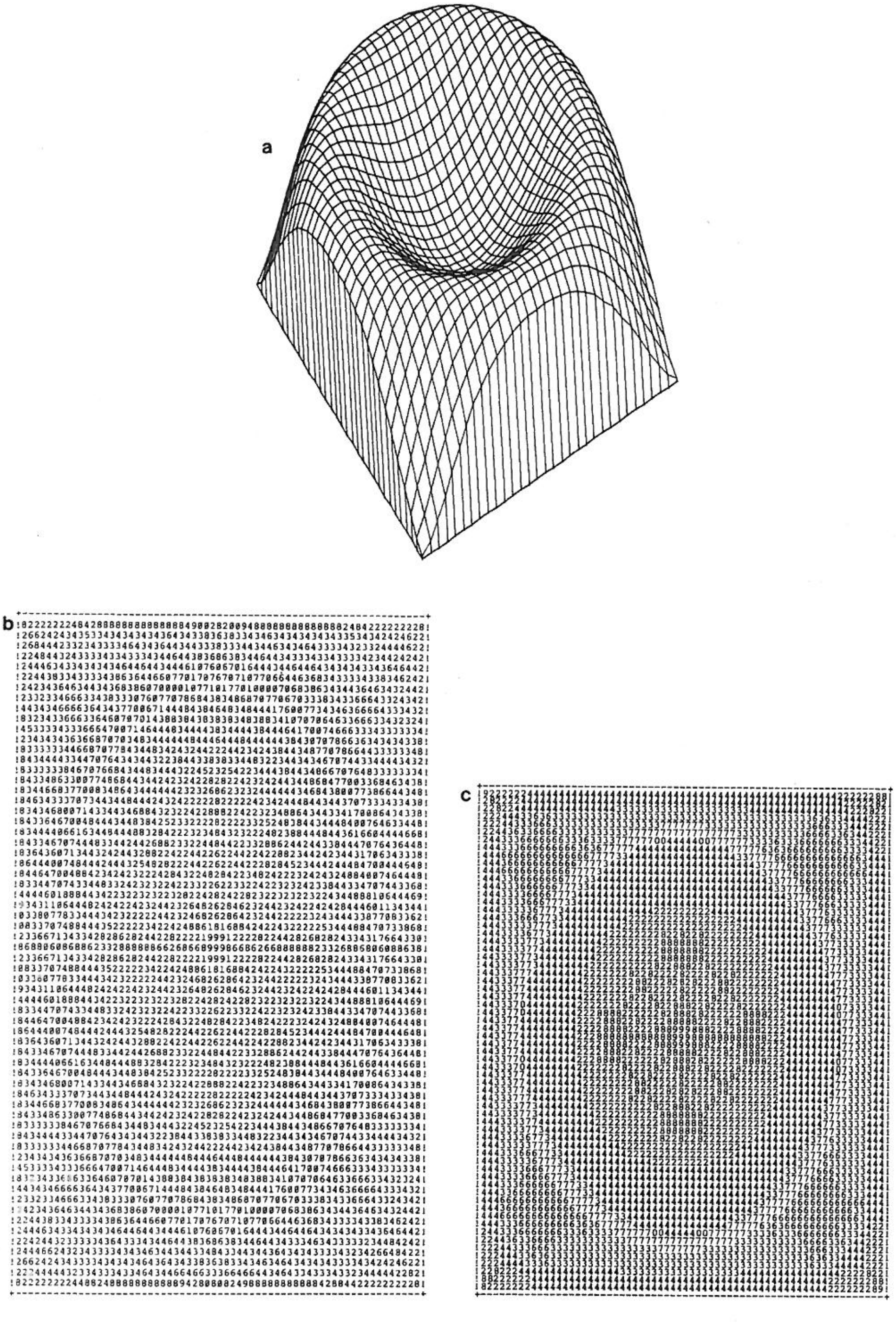

FIG. 4. (a) Surface of revolution plot; (b) labeling of surface of revolution based on generalized splines; (c) labeling of surface of revolution based on DCT (10 coefficients). The numbers correspond to: 1 = flat, 2 = convex hill, 3 = concave hill, 4 = saddle hill, 5 = slope, 6 = ridge, 7 = peak, 8 = ravine, 9 = pit, 0 = saddle, * = inflection.

TABLE 6

Image	RMS of Cubic Facet	RMS of DCT
Chair	7.253	6.796
Machine Parts	7.799	7.363
Two cells	2.872	2.659
Surface of Revolution	1.115	1.417
Step Edge	2.478	2.213

TABLE 7

Image	RMS of Spline	RMS of DCT—49	RMS of DCT—10	RMS of Cubic Facet
Chair	2.398	2.756	6.796	7.253
Machine parts	2.560	2.938	7.363	7.799
Two cells	0.686	0.800	2.659	2.872
Surface of Revolution	0.323	0.448	1.417	1.115
Step Edge (height = 5)	0.771	0.945	2.213	2.478
Step Edge (height = 3)	0.771	0.654	2.618	2.917
Ridge–ravine surface	0.000	19.695	70.744	0.000

6.2. Ridge-Ravine Surface

A ridge–ravine surface of size 64 × 64 having no noise can be generated by the equation

$$f(r,c) = c * r^2.$$

Taking the origin at image coordinates (32, 32), the surface plot is illustrated in Fig. 5a. This is a very interesting surface. A ridge line running along the column axis gradually decreases until it becomes a flat, then the curvature changes, and it turns into a ravine line. The results of the classifier are shown in Fig. 5b (generalized splines) and c (DCT). Each number in the figure represents the label assigned the pixel by the classifier. As expected, the results from the spline approach are perfect since the surface was generated from a low order polynomial. It correctly classifies the ridge and ravine lines, the flat at the center, and the saddle hillsides. On the other hand, the DCT has some difficulty fitting this surface as seen from the residual error in Table 7. It incorrectly labels the set of pixels which occur near the center of the surface. The center pixel in the image is labeled a ridge, instead of a flat as it should be.

6.3. Surface of Revolution

A surface of revolution of size 64 × 64 with no noise can be generated by the equation

$$f(r,c) = k * \mathrm{SIN}(0.5 * (r^2 + c^2))$$

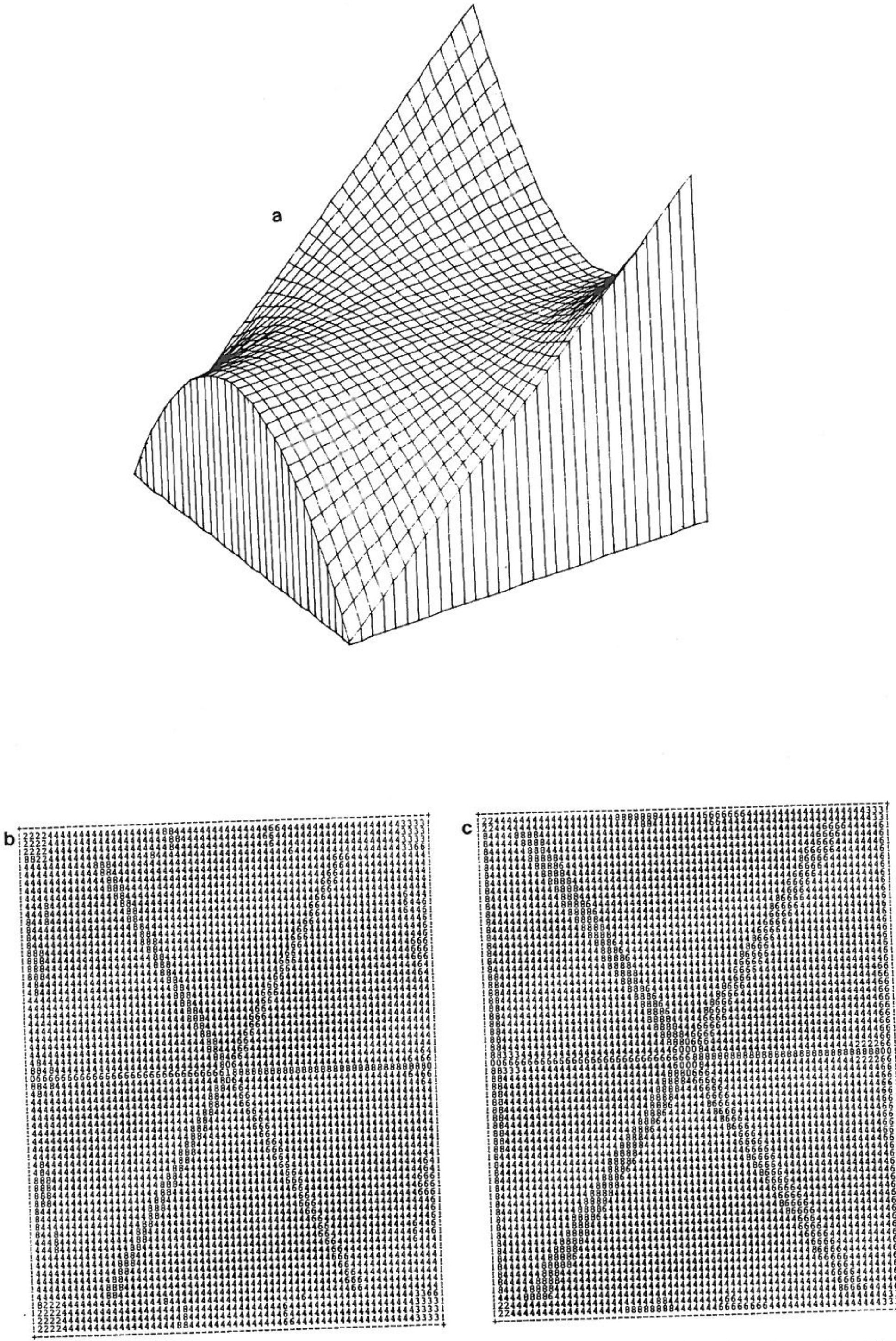

FIG. 5. (a) Ridge–ravine surface plot; (b) labeling of ridge–ravine surface based on generalized splines; (c) labeling of ridge–ravine surface based on DCT (10 coefficients). The numbers correspond as in Figs. 4.

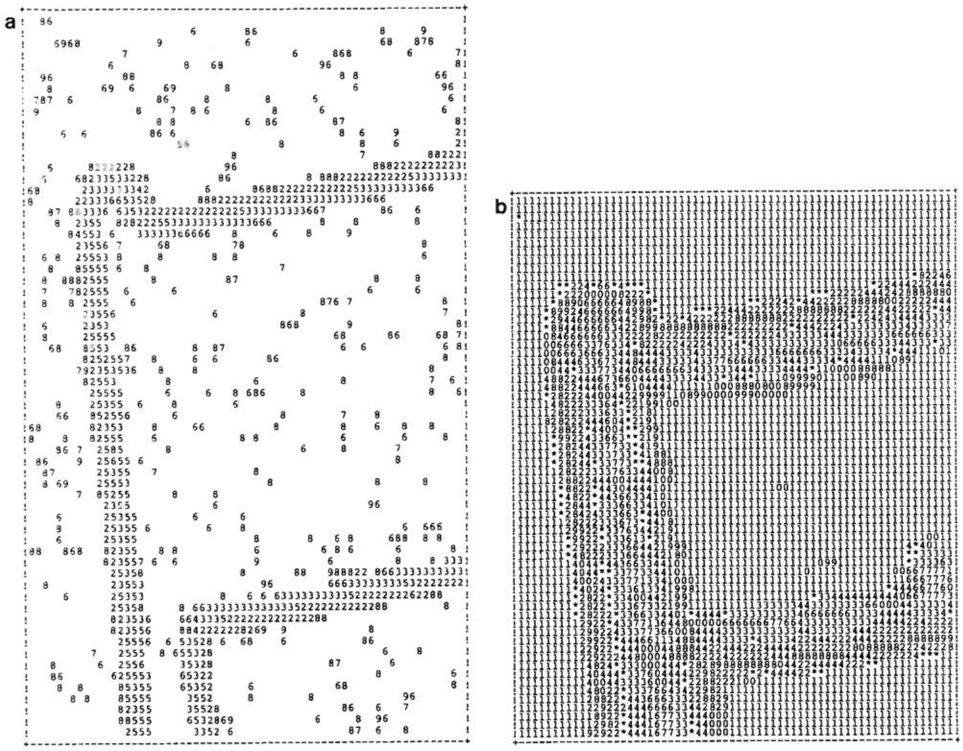

FIG. 6. (a) Labeling of top left corner of chair based on generalized splines; (b) labeling of top left corner of chair based on DCT (10 coefficients). The numbers correspond as in Figs. 4.

with origin at image coordinates (32, 32). The surface plot is illustrated in Fig. 4a. The labeling produced using the generalized splines gave very poor results (see Fig. 4b). The labeling from the DCT is much better (see Fig. 4c). A 3 × 3 set of pits were found at the center of the surface. Also, the local surface fits are very poor on this surface of revolution. This leads to some unexpected results, such as the peaks found on the rim of the surface where the ridges and ravines come together. The pixels labeled saddle on the image occur at locations where both a ridge and ravine were detected within the same pixel.

Notice that the labelings produced are not perfectly symmetric as one would expect on a radially symmetric surface. The reason for this is that the surface estimation is done with rectangular windows which produces different approximations at the same radial distance from the axis of revolution, and hence radially unsymmetric labeling. Symmetric labeling would be produced by using a circular window, but choosing a particular window shape requires a priori knowledge of the nature of the image surface.

6.4. Real Images

In this section we show the results of the classifier on two real images. The results on the top left corner of a chair image are illustrated in Figs. 6a and b. The results on the image of the two cells are illustrated in Figs. 7a and b. The various nonflat

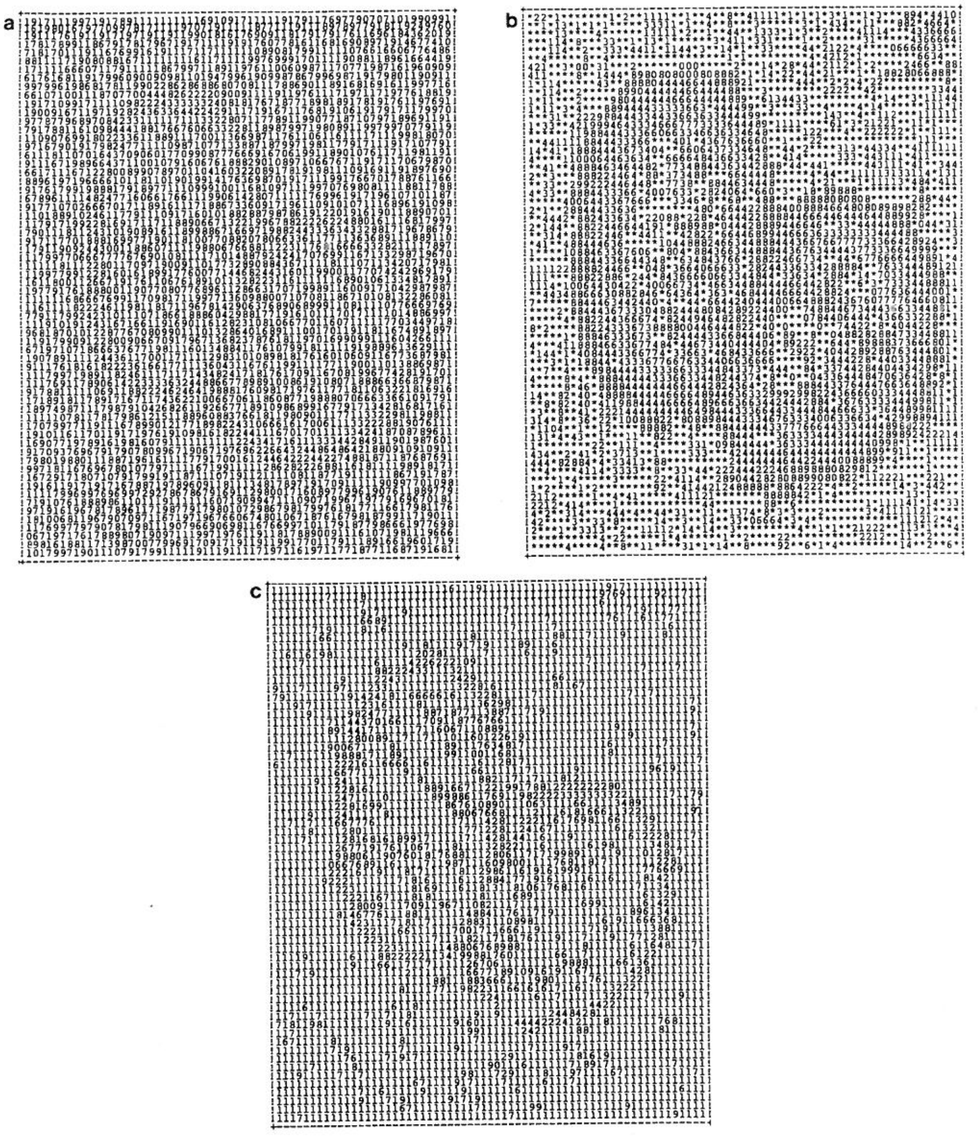

FIG. 7. (a) Labeling of two cells based on generalized splines; (b) labeling of two cells based on DCT (10 coefficients); (c) Fig. 7a with larger eigenvalue threshold. The numbers correspond as in Figs. 4.

labels in the backgrounds of the images are caused by very slight dips and rises in the surface fits. These may be cleaned up by requiring the eigenvalues to be above a certain threshold to be considered nonzero. (See Fig. 7c.)

7.0. CONCLUSIONS

In this paper we have given a precise mathematical description of the various topographical structures which occur in a digital image. Our set of topographical categories is invariant under grey tone monotonically increasing transformations and consists of {peak, pit, ridge, ravine, saddle, flat, and hillside}, with hillside being broken down further into subcategories {inflection point, slope, convex hill, concave hill, and saddle hill}. The hillside subcategories are not invariant under the monotonic transformations.

The topographic label assigned a pixel is based on the pixel's first and second-order directional derivatives. Generalized splines and the discrete cosine basis are used to estimate the directional derivatives of the underlying grey tone intensity surface. The calculation of the extrema of the second directional derivative can be done efficiently and stably by forming the Hessian matrix and calculating its eigenvalues and their associated eigenvectors. Strict local 1-dimensional extrema (such as pit, peak, ridge, ravine, and saddle) are found by searching for a zero-crossing of the first directional derivative in the directions of extreme second directional derivative (the eigenvectors of the Hessian). We have also identified another direction of interest, the Newton direction, which points toward the extremum of a quadratic surface. The classification scheme was found to give satisfactory results on a number of test images.

Research with the topographic primal sketch needs to be done in four areas: better basis functions, use of fitting error, solving the ridge (ravine) continuum problem, and grouping of the topographic structures.

Basis functions worth considering include piecewise trigonometric polynomials, algebraic polynomials of order higher than cubic, splines with variable knots, and 2-dimensional functions not obtained as tensor products. The spline basis functions have a significantly smaller RMS error, yet the labeling based on them may be much poorer than the DCT based labeling (Fig. 4). A small RMS fitting error does not guarantee a good topographic labeling. It still remains unclear what properties are desirable for basis functions. The DCT (both in RMS error and final labeling) is not sufficiently better than the local cubic facet to warrant the extra expense. In terms of RMS error and computational efficiency, the generalized spline basis is best, but the surface of revolution example indicates a definite weakness.

Use needs to be made of fitting error when trying to decide into which class a pixel falls. Noise causes the fitting error to increase and increased fitting error increases the uncertainty of the labeling. Also, global knowledge of how the topographic structures fit together could be used to undo the misclassification error due to noise. The way the neighborhood size affects the surface fitting error and the classification scheme was investigated in Haralick, Watson, and Laffey [9], but needs further study.

The zero-crossing portion of the algorithm is reasonably rigorous, and ostensibly justified despite its substantial cost. However, Fig. 8 casts doubt on the entire philosophy of calculating zero-crossings. If thresholds are going to be "tuned" to the image, they might as well be tuned using only information at the center of the pixels. Whether or not information at zero crossings is really required, and whether good topographic labels can be obtained without "tuning" the thresholds, will require considerable further research.

The ridge (ravine) continuum problem needs to be solved. It may be that there is no way to distinguish between a true ridge and a ridge continuum by only using the values of partial derivatives at a point. To solve the problem may require complete use of the partial derivatives in a local area about the pixel.

Most important is the use of these topographic categories in a general computer vision system. Techniques need to be developed for grouping and assembling topographically labeled pixels together to form the primitive structures involved in

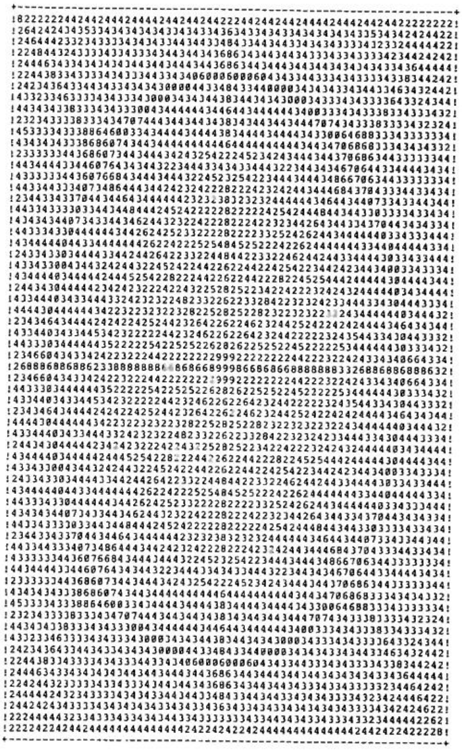

FIG. 8. Figure 4b with center pixel labeling and larger thresholds. The numbers correspond as in Figs. 4.

higher level matching and correspondence processes. How good can the stereo correspondence or the frame-to-frame time varying image correspondence problem be accomplished using the primitive structures in the topographic primal sketch? How well can the topographic sketch be used in undoing the confounding effects of shading and shadowing? How well will the primitive structures in the topographic sketch perform in the 2-dimensional to 3-dimensional object matching process?

REFERENCES

1. N. Ahmed, T. Natarajan, and K. R. Rao, Discrete cosine transform, *IEEE Trans. Comput.* **C-23**, 1974, 90–93.
2. P. L. Chebyshev, Sur l'interpolation par la methode des moindres carres, *Oeuvres* **1**, 1859, 473–498; reprint, Chelsea, New York, 1962.
3. C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
4. R. W. Ehrich and J. P. Foith, Topology and semantics of intensity arrays, in *Computer Vision Systems*, pp. 111–128, Academic Press, New York, 1978.
5. G. C. Grender, TOPO III: A Fortran program for terrain analysis, *Comput. Geosci.* **2**, 1976, 195–209.
6. R. M. Haralick, The digital edge, in *Proceedings of the 1981 Conference on Pattern Recognition and Image Processing*, Dallas, Texas, 1981, pp. 285–294.
7. R. M. Haralick, Edge and region analysis for digital image data, *Comput. Graphics Image Process.* **12**, 1980, 60–73.
8. R. M. Haralick, Zero-crossing of second directional derivative edge operator, in *SPIE Proceedings on Robot Vision*, Arlington, Virginia, 1982.
9. R. M. Haralick, L. T. Watson, and T. J. Laffey, The topographic primal sketch, *Int. J. Robotics of Research* **2**, 1983, 50–72.

10. S. Hsu, J. L. Mundy, and P. R. Beaudet, Web representation of image data, in *The Fourth International Joint Conference on Pattern Recognition*, Kyoto, Japan, 1978, pp. 675–680.

11. E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Wiley, New York, 1966.

12. E. G. Johnston and A. Rosenfeld, Digital detection of pits, peaks, ridges, and ravines, *IEEE Trans. Systems, Man, Cybernetics* **SMC-5**, 1975, 472–480.

13. F. A. Kamanger and K. R. Rao, Fast algorithms for the 2-D discrete cosine transform, *IEEE Trans. Comput.* **C-31**, No. 9, 1982, 899–906.

14. T. J. Laffey, R. M. Haralick, and L. T. Watson, Topographic classification of digital image intensity surfaces, in *Proc. IEEE Workshop on Computer Vision: Theory and Control*, Rindge, New Hampshire, August, 1982.

15. H. C. Lee and K. S. Fu, The GLGS image representation and its application to preliminary segmentation and pre-attentive visual search, in *Proceedings of the 1981 Conference on Pattern Recognition and Image Processing*, Dallas, Texas, pp. 256–261.

16. D. Marr, Early processing of visual information, *Philos. Trans. Roy. Soc. London, Ser. B* **275**, 1976, 483–524.

17. D. Marr, Visual information processing: The structure and creation of visual representations, *Philos. Trans. Roy. Soc. London, Ser. B* **290**, 1980, 199–218.

18. K. Paton, Picture description using Legendre polynomials, *Comput. Graphics Image Process.* **4**, 1975, 40–54.

19. T. K. Peucker and E. G. Johnson, *Detection of Surface-specific Points by Local Parallel Processing of Discrete Terrain Elevation Data*, Computer Science Center, University of Maryland Technical Report No. 206, November 1972.

20. T. K. Peucker and D. H. Douglas, Detection of surface-specific points by local parallel processing of discrete terrain elevation data, *Comput. Graphics Image Process.* **4**, 1975, 375–387.

21. H. Rutishauser, Jacobi method for real symmetric matrix, in *Handbook for Automatic Computation, Vol. II, Linear Algebra* (Wilkinson and Reinsch, Eds.), Springer-Verlag, New York, 1971.

22. L. F. Shampine and R. C. Allen, *Numerical Computing: An Introduction*, Saunder, Philadelphia, 1973.

23. G. Strang, *Linear Algebra and Its Applications*, 2nd ed., pp. 243–249, Academic Press, New York, 1980.

24. J. Toriwaki and T. Fukumura, Extraction of structural information from grey pictures, *Comput. Graphics Image Process.* **7**, 1978, 30–51.