CVGIP: GRAPHICAL MODELS AND IMAGE PROCESSING
Vol. 55, No. 6, November, pp. 428–446, 1993

Image Encoding, Labeling, and Reconstruction from Differential Geometry*

ERHARDT BARTH,† TERRY CAELLI, AND CHRISTOPH ZETZSCHE‡

Computer Science Department, The University of Melbourne, Parkville, Victoria, 3052 Australia

Received July 3, 1991; revised June 4, 1992; accepted June 23, 1993

In this paper we consider how the representation of images as surfaces, and their characterizations via surface differential forms, can be related to the concept of redundancy in the intensity signal. In contrast to common approaches, the basic surface types (planar, parabolic, elliptic/hyperbolic) are not seen as equal-priority classes, but as corresponding to different degrees of redundancy. This leads to a new approach to image representation and region labeling based upon generalized curvature measures. Furthermore, we employ different reconstruction algorithms to show that elliptic surface patches carry the significant information in natural images. Based upon deterministic and stochastic relaxation techniques, these algorithms allow one to reconstruct the original image from (i) "elliptic intensities" only and (ii) curvature measures which are zero for nonelliptic regions. © 1993 Academic Press, Inc.

1. INTRODUCTION

Low-level visual routines for the characterization of images are generally focused upon the fundamental problem of efficiently encoding the image, that is, the capture of salient or nonredundant image information that can be used for higher level pattern or object recognition. In image processing, representations via transform techniques, scale-space or even linear prediction [1] have been used to encode and segment images.

In recent years, partly due to the development of range imaging and partly due to the interest in inferring geometric properties of object surfaces from intensity or range data, differential geometry has become a popular method for representing depth or image data. For example, a large number of object recognition systems characterize and segment image range data via input surface curvatures in the initial encoding stages (see [2–5]). In these cases, Gaussian (K) and mean (H) curvatures are used to determine the eight well-known local surface types. Haralick [6] used different differential operators to label each pixel of an intensity image with one of ten topographic labels.

We have recently developed an alternative conceptual framework which is based on "intrinsic signal dimensionality" and differs from the common approaches in two ways [7–9]. First, we do not think of the basic surface types as equally important classes, but claim them to be ruled by a hierarchical order which is determined by their significance. This results in a classification of signals into 0D (planar), 1D (parabolic) and 2D (elliptic/hyperbolic) regions corresponding to the different degrees of predictability. Second, we do not use the classical measures H and K, but measures which are more basic and reliable from a signal processing point of view. This signal processing type of approach has also led us to the development of a Fourier (spatial frequency) description of the curvature operators which allows to overcome the limitations imposed by the differential calculus.

The main point of this paper is to show the redundancy of developable surface patches (planar and parabolic regions) or, equivalently, to demonstrate the predominant significance of curved (2D) regions for the representation of relevant image information. (A more detailed discussion of the redundancy concept in the context of image data compression can be found in [10]. Some hints and examples are given in Section 2.3.) In general, the relevance of a certain type of image features or a certain representation can only be judged with respect to a specific task, that is, one can evaluate the amount of data, computational effort, etc., needed to achieve a certain performance.

Our "criterion for relevance" is more general. It is based on the reconstruction of the image on the sole basis of the suggested "essential" features. Our position can be related to standard arguments concerning the decorrelating and energy concentration properties of the Karhunen–Loève transform, where the irrelevance of

* Study supported by grants from the Daimler-Benz Foundation and Deutsche Forschungsgemeinschaft DFG Re 337/7-1.

† Present address: Institut fuer Medizinische Psychologie, Goethestrasse 31, 80336 Muenchen, Germany.

‡ Visiting Scholar from Lehrstuhl fuer Nachrichtentechnik, Technische Universitaet Muenchen, Arcisstrasse 21, 80290 Muenchen, Germany.

1049-9652/93 $5.00
Copyright © 1993 by Academic Press, Inc.
All rights of reproduction in any form reserved.

coefficients is demonstrated by showing them to be unnecessary for the achievement of a sufficient signal to noise ratio of the reconstructed image [1]. In analogy, we want to show that the planar and parabolic regions are not essential for a satisfactory reconstruction of the image. This is done in two different ways: in Section 3 by reconstructing the image from original intensity values at image positions which are classified as curved (2D regions) and, in Section 4, by dealing with the harder inverse problem of reconstruction from curvature measures.

To attain these ends we first consider some fundamental aspects of what may be termed "geometric signal processing"—in this case the use of filter theory to generalize fundamental measures in differential geometry and compute them in a robust manner.

2. COMPUTING CURVATURES OR THEIR EQUIVALENTS

2.1. Classical Measures

In classical differential geometry a surface is defined, in general, by the parametric representation

$$\mathbf{S}(u, v) = \{X(u, v), Y(u, v), Z(u, v)\}, \tag{1}$$

where (u, v) correspond to the surface (index) intrinsic parameters and (X, Y, Z) to real-valued functions of (u, v) which result in three-dimensional loci of surface points. The fundamental differential operators used in differential geometry to (uniquely) determine its shape, locally, are

$$\{\mathbf{S}_u, \mathbf{S}_v, \mathbf{S}_{uv}, \mathbf{S}_{uu}, \mathbf{S}_{vv}\}, \tag{2}$$

where $\mathbf{S}_z$ refers to partial differentiation of $\mathbf{S}$ with respect to z.

For our application images are special types of surfaces: viewer-dependent, insofar as the intensity (or depth) is a function (f) of the viewer's stance. That is, for projection plane coordinates (x, y), f is a function of x, y, giving ($x = u$, $y = v$ in (1)):

$$\mathbf{f}(x, y) = \{x, y, f(x, y)\}. \tag{3}$$

Such surfaces are termed "Monge" patches [11] and, sometimes, the "2 1/2-D image" when referring to depth maps [12]. The fundamental theorem of surfaces not only shows that the five differential operators defined in Eq. (2) uniquely define a surface up to rigid motions, but that surface curvatures are readily derived from these measures. Specifically, these operators determine two matrices which constitute the coefficients of the surface first and second fundamental forms, which, for Monge patches (3), are (the matrix notations are chosen to illustrate common structures of the measures introduced below)

$$(g_{ij}) = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} 1 + f_x^2 & f_x f_y \\ f_x f_y & 1 + f_y^2 \end{pmatrix} \tag{4}$$

$$(b_{ij}) = \begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix} \cdot \mathbf{N},$$

$$\mathbf{N} \equiv \text{surface normal.} \tag{5}$$

That is,

$$(b_{ij}) = \frac{1}{\sqrt{1 + f_x^2 + f_y^2}} \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}. \tag{6}$$

The principle curvatures correspond to the eigenvalues of

$$(b_{ij})(g_{ij})^{-1}. \tag{7}$$

Equally, the eigenvalues of the Hessian of $f(x, y)$ provide unnormalized principal curvature estimates and are adequate for the classification of region types in terms of their curvature sign.

For k_1, k_2 corresponding to the eigenvalues of (7), mean (H) and Gaussian (K) curvatures are given as

$$H = \frac{1}{2}(k_1 + k_2) = \frac{1}{2}\,\text{trace}(b_{ij})(g_{ij})^{-1} \tag{8}$$

and

$$K = k_1 \cdot k_2 = \frac{\det(b_{ij})}{\det(g_{ij})}. \tag{9}$$

Similar to Eqs. (8) and (9) we obtain the Laplacian form,

$$\Delta f = \frac{1}{2}\,\text{trace}(h_{ij}) = \frac{1}{2}(f_{xx} + f_{yy}) = \frac{1}{2}(d_1 + d_2), \tag{10}$$

and the determinant of the Hessian,

$$D = \det(h_{ij}) = f_{xx} f_{yy} - f_{xy}^2 = d_1 d_2, \tag{11}$$

where d_1, d_2 are the eigenvalues of the Hessian

$$(h_{ij}) = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{pmatrix}. \tag{12}$$

2.2. Geometrical Signal Processing

Differential geometry uses the calculus for a quantitative description of surface properties. Therefore the basic measures are first and higher order derivatives (traditionally computed by finite difference, or convolution-based procedures). However, by using Fourier methods for a frequency domain design of the operators we have been able to provide more stable and more general curvature measures.

We first note that Eq. (11) can be rewritten as

$$D = \frac{1}{4}(f_{xx} + f_{yy})^2 - \frac{1}{4}(f_{xx} - f_{yy})^2 - f_{xy}^2 = (\Delta f)^2 - \varepsilon^2, \tag{13}$$

with eccentricity ε defined as

$$\begin{aligned}\varepsilon^2 &= \frac{1}{4}(f_{xx} - f_{yy})^2 + f_{xy}^2 = f_{xy}^2 + f_{x'y'}^2 \\ &= \frac{1}{4}(f_{xx} - f_{yy})^2 + \frac{1}{4}(f_{x'x'} - f_{y'y'})^2.\end{aligned}$$

The (x', y') coordinates are rotated 45° relative to (x, y) and are introduced to illustrate that the eccentricity can result from bandpass filters, as discussed below. The eccentricity ε is a measure of "elongatedness", that is, it is equal to zero for circular symmetric patches and equal to the Laplacian (Δf) for parabolic regions.

Interestingly enough, the eigenvalues d_1, d_2 can be computed using the same Laplacian (Δf) and eccentricity (ε) operators (the proof is straightforward):

$$d_{1,2} = \Delta f +/- \varepsilon. \tag{14}$$

To obtain operators with different selectivities for positive and negative elliptic patches we introduce the clipping operation:

$$d^+ = \begin{cases} d, & d \geq 0 \\ 0, & d < 0 \end{cases}; \quad d^- = \begin{cases} 0, & d \geq 0 \\ -d, & d < 0 \end{cases}. \tag{15}$$

As shown in Table 1, the clipped-eigenvalues operator

$$c(x, y) = d_2^+(x, y) - d_1^-(x, y) \tag{16}$$

is equal to the minimum (with respect to orientation) second-order derivative within elliptic regions and zero elsewhere.

Remember, that we are concerned with differentiating between curved (2D) and not curved image regions. From Table 1 it should be clear that this can be done by using the Laplacian and the eccentricity as basic measures. In the following we present possible implementations and generalizations of this operation from a signal processing point of view.

Even- and odd-order partial derivatives correspond, via the convolution and differentiation theorems [1], to even (real) and odd (imaginary) transfer functions of the form $(iu)^n$, or $(iv)^n$, where (u, v) are the (spectral) spatial frequency coordinates and n is the order of differentiation.

Introducing the filter functions (in polar frequency coordinates ρ, θ, where $A(\rho)$ corresponds to the radial filter tuning function),

$$\begin{aligned} C_n &= (i)^n A(\rho)\cos(n\Theta) \\ S_n &= (i)^n A(\rho)\sin(n\Theta) \end{aligned} \tag{17}$$

with their corresponding image domain convolution kernels $c_n(x, y)$, $s_n(x, y)$, we can define a generalized eccentricity (ε_n) as

$$\varepsilon_n^2 = (c_n(x, y) * f(x, y))^2 + (s_n(x, y) * f(x, y))^2, \tag{18}$$

with "*" denoting convolution with respect to x and y. Note that for $A(\rho) = (2\pi\rho)^2$ and $n = 2$, ε_n in Eq. (18) corresponds to ε in (13) and for $n = 0$ to Δf in (10). The gradient is defined by ε_n with $n = 1$ and $A(\rho) = 2\pi\rho$. In practice we usually introduce a Gaussian blurring which

TABLE 1
Basic Surface Types

Signal classes	Basic surface types	H	K	$k_{\max}$	$k_{\min}$	$d_{\max}$	$d_{\min}$	D	Δf	$d_2^+ - d_1^-$
2D	+ Elliptic (pit)	>0	>0	>0	>0	>0	>0	>0	>0; >ε	>0; =$d_{\min}$
	− Elliptic (peak)	<0	>0	<0	<0	<0	<0	>0	<0; <ε	<0; =$d_{\min}$
	Hyperbolic (saddle ridge)	<0	<0	<0	>0	≠0 Opposite sign		<0	$\lvert\Delta f\rvert < \varepsilon$	=0
	Hyperbolic (saddle valley)	>0	<0	>0	<0			<0		=0
	Hyperbolic (minimal surface)	=0	<0	$k_{\max} = -k_{\min}$				<0		=0
1D	+ Parabolic (valley)	>0	=0	>0	=0	>0	=0	=0	>0; =ε	=0
	− Parabolic (ridge)	<0	=0	<0	=0	<0	=0	=0	<0; = −ε	=0
0D	Planar	=0	=0	=0	=0	=0	=0	=0	=0	=0

Note. Traditionally H and K are used as initial estimates for region labelling (Besl and Jain [3]). Instead, we use the simpler and more robust Δf and ε (or ε_n), since we are not interested in differentiating within hyperbolic patches. Last column illustrates that the clipped-eigenvalues operator (16), used for reconstruction, depicts the elliptic patches, being equal to the minimum eigenvalue of the Hessian. This is because for elliptic patches d_1 and d_2 have the same sign and thus either d_2^+ or d_1^- is equal to zero. The 2D, 1D, 0D classification is discussed in detail in Zetzsche and Barth [7].

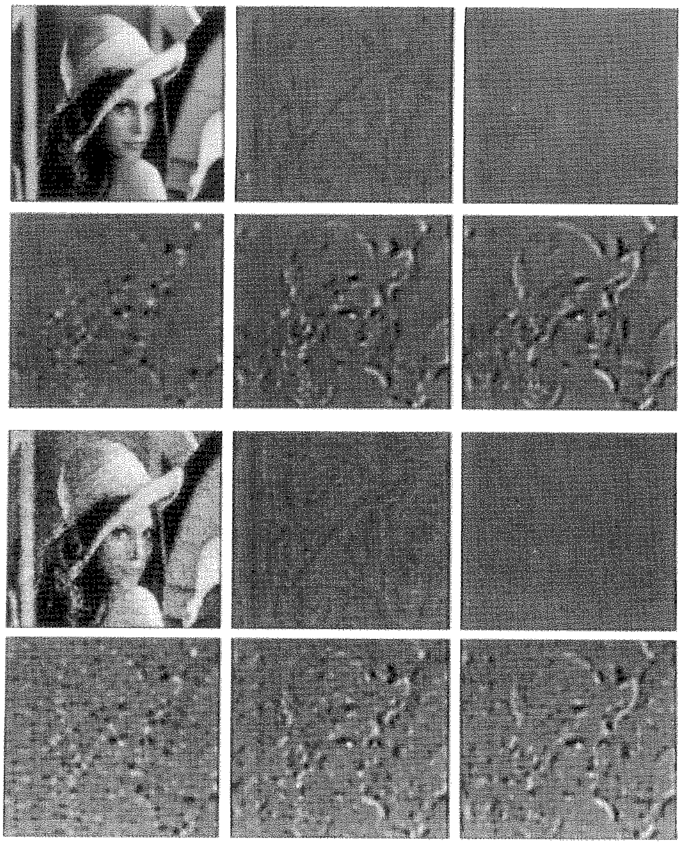

FIG. 2.1. Traditional geometric measures H and K compared to the more robust clipped eigenvalues (defined in Eq. 16). Results are illustrated as gray-level images with "gray" for zero, "bright" for positive, and "dark" for negative values of the corresponding measures. First row: mean (in the middle) and Gaussian curvature (on the right) of the natural intensity surface shown on the left. Second, row: output of the clipped eigenvalues operator (Eq. (16)) for $n = 2, 4, 6$ (from left to right) and $A(\rho) = (2\pi\rho)^2 \exp(-\pi\rho^2/4\sigma_f^2)$ including a Gaussian blurring with a space constant $\sigma = 1.8$ pixels ($\sigma = 1/4\sigma_f$). The same operations are repeated below for a noisy input (25% uniformly distributed additive noise).

changes, for example, the second-order radial profile to $A(\rho) = (2\pi\rho)^2 \exp(-\pi\rho^2/4\sigma_f^2)$.

Figure 2.1 illustrates the benefits of using the c operator (defined in Eqs. (16) to (18) with ε_n replacing ε) for the processing of natural images. It also demonstrates the effect of increasing n in Eq. (17). Since these basic operations are used in the following sections on reconstruction, we devote the remainder of this section to a brief discussion of the differences between our 2D operators and the traditional curvatures. Note that, by definition, 2D operators and classical curvatures have, in a strict mathematical sense, the same classification properties with respect to the basic signal classes as defined in Table 1. We have argued elsewhere that the traditional geometric approach can be seen as a specific realization of more general principles for the nonlinear processing of multidimensional signals [8–10]. Here we just give some motivation for increasing the parameter n in Eq. (17) and for using curvature operators which are not normalized.

From Eq. (13) one can understand ε_2 as resulting from a nonlinear interaction between oriented filters (l_{xx}, l_{yy}, $l_{x'x}$, $l_{y'y'}$) which have an orientation bandwidth of 90°. Increasing n reduces the bandwidth, for example, to 30° for $n = 6$. This is reasonable since it introduces additional integration in the process of differentiation, thus providing a more robust representation (the corresponding convolution kernel is elongated orthogonal to the direction in which it differentiates). Also, a reduced bandwidth is in accordance with models of the biological visual system [8]. It is important to note, however, that for the c operator the integration is implicitely performed in direction of the minimum curvature and is therefore fundamentally different from the simple and unspecific kind of integration one can obtain by increasing the support of the derivative operators or by Gaussian smothing of the input function.

The main differences between K and D, and between H and Δf, are due to the division by the first fundamental form $det\ (g_{ij})$ (see Eqs. (8) to (11)). This is essentially a normalization of even-symmetric measures (like l_{xx}, ε_2, ε_4, corresponding to real-valued filter functions) to odd-symmetric measures (like l_x, ε_1, ε_3, corresponding to imaginary filter functions). (An example of how the even/odd relationship can be understood as a 2D-phase property is given in [8]) There are two specific points we want to make here:

(i) The signs of K and D are identical and the signs of H and Δf differ only within hyperbolic regions (see Table 1). In practical applications, however, testing for zero and sign involves thresholding. Thus, the classification, for example based on the signs of K versus D, will differ due to the different noise sensitivities and tuning properties of the measures involved (see Fig. 2.1).

(ii) Usually, mean and Gaussian curvatures (H and K) are used because they are invariant to rigid motion of the surface. However, different visual tasks involve different invariance properties which are often contradictory and difficult to define.

For example, it has been claimed that the same procedures could be used to segment range and intensity images [3] although the desired invariance is completely different for the two cases. While the segmentation of range images should be invariant to rigid motion of the objects involved, i.e., to rotations and translations of the (x, y, f) (see Eq. (3)) coordinates, the segmentation of intensity images is supposed to be invariant to illumination, i.e., to multiplicative distorsion, and to rotations, translations and dilations, but of the (x, y) coordinates only. These, quite different, desired invariance properties will interfere with noise and thresholding effects, and this interference will be more critical for normalized measures, like H and K (see Fig. 2.1), because of their ex-

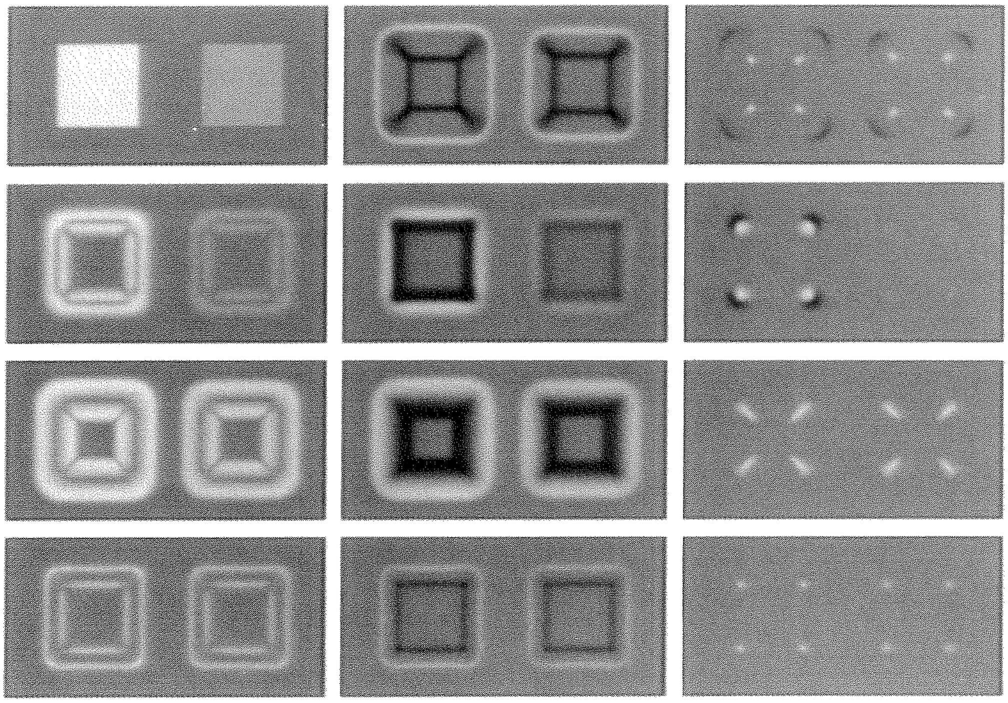

FIG. 2.2. Normalization and contrast invariance. First row: input (on the left) representing squares of different contrast. The geometric measures H (middle) and K (right) are invariant to the change in contrast (same operators as in Fig. 2.1). Second row: ε_2 (Eq. (18), $n = 2$, $A(\rho) = (2\pi\rho)^2 \exp(-\pi\rho^2/4\sigma_f^2)$); Laplacian (Eq. (10)); determinant of the Hessian (Eq. (13)). Third row: same measures as second row but divided by the gradient ε_1 (Eq. (18), $n = 1$, $A(\rho) = 2\pi\rho$). Fourth row: same as third row but using ε_1 with $A(\rho) = (2\pi\rho)^2 \exp(-\pi\rho^2/4\sigma_f^2)$. $\sigma = 2.6$ pixels.

treme dynamics. Think, for example, of range data from differently sized spheres with radii from 1 to 100. The Gaussian curvature of the corresponding inputs will then vary in a range from 1 to 10,000. Similar problems will occur with curved contours in intensity images. Using the standard geometrical operators, only sharp corners can be labelled, with sufficient reliability, as "curved," whereas smoothly curved contours, which would be clearly classified as "curved" by any human observer, will be labelled as straight (parabolic) by the technical system.

We do not claim to have a solution for all these problems, but it should be clear from the above discussion that no simple optimal solution will probably exist. For intensity images we suggest to perform either no normalization at all, which may be the best solution anyway, or to use an odd ε_n (Eq. (18), with $n = 1$ or even $n = 3, 5, \ldots$) for normalization. Figure 2.2 illustrates how Laplacian (Δf), eccentricity (ε), and curvature measures like $\mathfrak{c}$ in Eq. (16) can be normalized in two different ways to achieve contrast invariance.

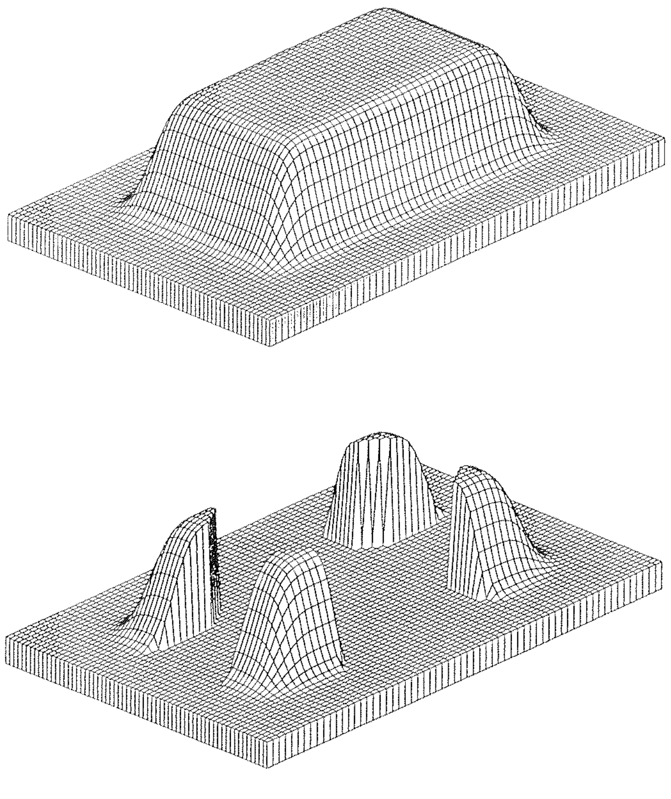

FIG. 2.3. A surface corresponding to the blurred image of a rectangle (top). The 2D regions (corners) shown below illustrate that the missing surface patches are determined by the given boundary conditions.

2.3. Curvature and Redundancy

2.3.1. *Redundancy from a Geometrically Extrinsic Point of View*

Figure 2.3 illustrates that developable surface patches are usually determined by boundary conditions. Cutting out the 2D patches from a rectangle surface one can formulate the task of joining the corners without producing new 2D regions. The suitable material for this type of work would be paper, and whatever one tries to do without creasing the paper will lead to the original surface. We try to reveal this type of redundancy in Section 3.

2.3.2. *Redundancy from a Geometrically Intrinsic Point of View*

Consider, for example, the problem of fitting a net to a given surface. If the surface is purely 0D and 1D, the fitting is straightforward. For 2D regions on the surface we have to change the local structure of the net such that the changed metric leads to a Gaussian curvature equal to the curvature of the surface. These changes will then contain essential information about the "shape" of the surface. In spite of this, there is no general mathematical proof for these "deviations from the flat metric" being sufficient for determining the embedding (see Section 4). Nevertheless, it should be clear that 0D and 1D regions, being flat, will enforce less "shape constraints" than 2D regions.

2.3.3. *Relationships to Other Concepts of Redundancy*

Another way to characterize the redundancy of 1D regions is to say that one can find a rotation of the (x, y) coordinates such that in the new coordinates (x', y') we have

$$f(x, y) = f_1(x') \quad \text{for 1D regions.} \tag{19}$$

That is, 1D regions are fully characterized by a one-dimensional function. Note, however, that there might be restricted variations within 2D regions also, for example, for umbilic points (where $K > 0$) or points on ruled surfaces which are not developable (where $K < 0$).

Once we can assume to know the orientation of (x', y') (the determination of which will always involve a nonlinear operation) we can consider linear prediction (or predictive compression) techniques were a given pixel's (x, y) intensity $(f(x, y))$ is modeled by

$$f(x, y) = \sum_{u,v \in R(x,y)} a_{uv} f(x + u, y + v), \tag{20}$$

where $R(x, y)$ corresponds to the "support" region about (x, y) used to predict the intensity at (x, y). From this perspective the 0D, 1D, and 2D signal classes correspond to different prediction coefficients and, in particular, developable surfaces or surfaces with zero Gaussian curvature always have subsets of a_{uv} coefficients which index collinear positions (u, v) where $f(x + u, y + v)$ is a linear function of (u, v) (they can be isometrically "ruled"). Obversely, if there does not exist a locus of points $(x + u, y + v)$, including (x, y), which has such an intensity function, then the point (x, y) must have nonzero Gaussian curvature. Such points have the least redundancy in the sense that the coefficients of (20) which can predict $f(x, y)$ are not linearly related to the shift position vector components (u, v).

Also, from the perspective of locally approximating images by some standard functions, e.g., splines, the number of free parameters will increase with the 0D, 1D, and 2D hierarchy. In this sense, from an image coding perspective, it would be reasonable to gradually allocate resources for the encoding of the different region types—a mechanism implicit to some advanced coding schemes (see [10]).

However, this is different from saying that *all* information is contained in 2D regions and, as a further step, that images can be recovered from only (2D) curvature measures. These, more radical, points of view are supported by the reconstruction results presented in the next sections.

Due to the fact that 2D regions are rare events in natural images [13], a 2D representation has low entropy. (Examples for this, and a comparison with the Laplacian pyramid are given in [14]; see also Figs. 3.3 and 3.4.) Thus, curvature extraction and reconstruction have an obvious potential for image coding, an issue beyond the scope of this paper. (The importance of curvature features is discussed from an image-coding perspective in [10].) Nevertheless, the relationship between what we call redundancy in a geometric and heuristic sense and the statistical concept of redundancy is not clear (particularly as the operations are extended to multiple scales). Note, however, that there is no a priori reason for one specific mathematical theory occupying the intuitive notion of redundancy.

3. IMAGE RECONSTRUCTION BASED ON LABELING AND PARTIAL DATA

Our first reconstruction paradigm depends upon how the partial data are selected and leads to the issue of region labeling based upon geometric characteristics of surfaces.

3.1. Region Labeling

Most applications involve a two-stage segmentation procedure. That is, the image (range or intensity) is initially labeled for surface type. Second, such labels are merged according to model data either in the form of known relationships between model parts (for example, bounds on interpart angles, etc.; see [4]) or by fitting surface parts to specific parametric surface types (B-splines, quadratic surfaces, etc.; see [15]). Such techniques have proved to be useful as early low-level procedures for object recognition systems. However, to this stage we know of no use of differential geometry to represent images, per se, from the perspective of developing efficient encoding and region labelling procedures.

Based on the measures discussed in Section 2 we used different algorithms to label images into six or less region types, that is, to assign one of the following labels to each pixel: positive elliptic (pit), negative elliptic (peak), hyperbolic (saddle), positive parabolic (valley), negative parabolic (ridge), and planar.

3.1.1. *Thresholding*

If we do not differentiate between "saddle ridge" and "saddle valley," it is clear from Table 1 that we can use Δf (10) and ε_n (18) as the basic measures for the following labeling algorithm, which assigns one of six labels (l_i) to each pixel:

$$\begin{array}{lll}
\text{if} & \{((\Delta f)^2 + \varepsilon_n^2) < h\} & \text{then } l_6 \text{ (planar)} \\
\text{else if} & \{\Delta f > \alpha_1\ \varepsilon_n\} & \text{then } l_1 \text{ (pit)} \\
\text{else if} & \{-\Delta f > \alpha_1\ \varepsilon_n\} & \text{then } l_2 \text{ (peak)} \\
\text{else if} & \{\varepsilon_n > \alpha_2 |\Delta f|\} & \text{then } l_3 \text{ (saddle)} \\
\text{else if} & \{\Delta f > 0\} & \text{then } l_4 \text{ (valley)} \\
\text{else} & & \qquad l_5 \text{ (ridge).}
\end{array} \tag{21}$$

The threshold h discriminates between "planar" and some kind of variation, while α_1, α_2 (both ≥ 1) determine the boundary between curved ($K \neq 0$) and parabolic regions (see Fig. 3.1). Theoretically, the algorithm is equivalent to the more traditional way of using the signs of H and K (see Table 1), but, in practice, the results will differ due to the thresholding.

As discussed in Section 2, Laplacian and eccentricity can be normalized (see Fig. 2.2) if a contrast invariant labeling is required. Results of applying the labeling (21) to intensity data are shown in Fig. 3.2.

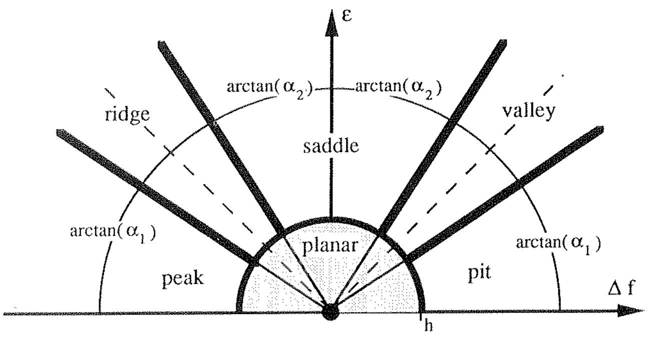

FIG. 3.1. Discrimination lines in the (ε, Δf) plane corresponding to the algorithm (21). (Other configurations may be conceived, for example, rectangular regions for "ridge," "valley," and/or "planar.")

3.1.2. *Relaxation Labeling*

We have extended classical relaxation labeling procedures [1] to image segmentation. Starting with geometric measures as initial probabilities for a certain label, consensus is reached between labels, neighbors, and scales.

The algorithms are presented in their most general form. However, depending on the application, the global structure can easily be reduced, for example, if the scale compatibility is not required or if less than six labels are sufficient.

To each pixel we assign the probability vector $p_i^t(x, y)$, denoting the probabilities for pixel (x, y) being labeled l_i (see Algorithm (21)) after t iterations.

We use the following geometric measures, based on Δf and ε_n (similar to (21)) to define the initial probabilities:

$$\begin{array}{ll}
m_1 = (\Delta f - \varepsilon_n)^+, & m_4 = (\Delta f)^-, \\
m_2 = (\Delta f + \varepsilon_n)^-, & m_5 = (\Delta f)^+, \\
m_3 = (|\Delta f| - \varepsilon_n)^-, & m_6 = \max\limits_{x,y,i} \{m_i(x, y)\}.
\end{array} \tag{22}$$

Here, "$(\,)^-$," "$(\,)^+$" correspond to the clipping operations defined in Eq. (15). First, the m_i's are normalized over (x, y):

$$m_{ni}(x, y) = \frac{m_i(x, y)}{\max_{x,y}\{m_i(x, y)\}}. \tag{23}$$

Finally, the initial probabilities are obtained as

$$p_i^0(x, y) = \frac{m_{ni}(x, y)}{\sum_{i=1}^N \ m_{ni}(x, y)} \tag{24}$$

to satisfy

$$\sum_{i=1}^{N} p_i^0 = 1, \quad N = \text{number of labels}. \tag{25}$$

The initial probabilities in Eq. (22) capture the local surface geometry. They are computed once (since we do not update the image) using the filters (17) and then merged as below. After a number of iterations, typically less than 10, the pixels are labeled according to the highest probability:

$$p_{\max} = \max_i \ p_i(x, y). \tag{26}$$

In contrast to earlier approaches [3], compatibility is not accomplished after thresholding the geometric measures and then merging the initial labels. Instead, the relaxation labeling algorithm will allow the initial measures to contribute to the final result until, in the end, a decision is

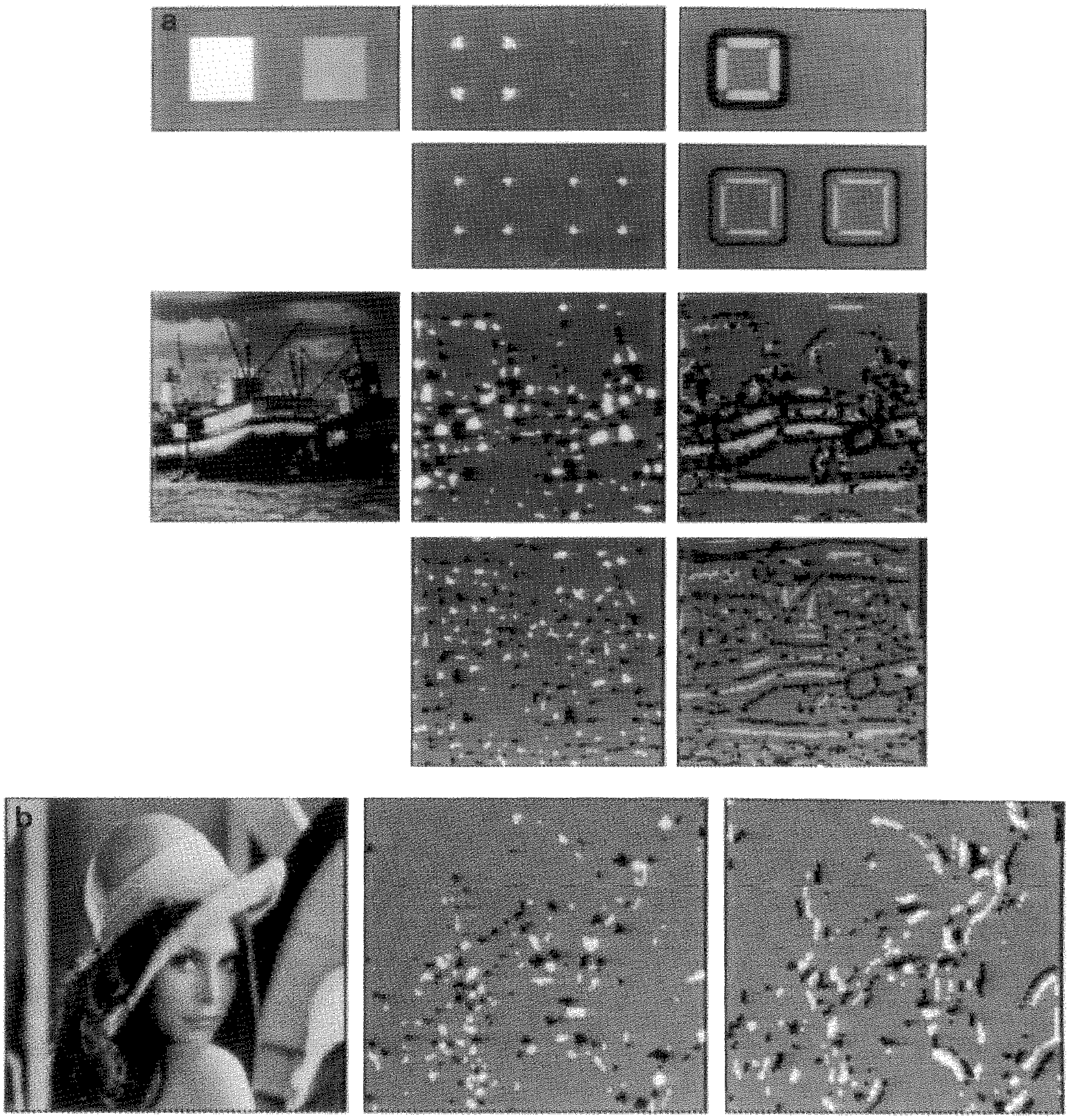

FIG. 3.2. (a) Threshold labeling, as defined by Eq. (21) and Fig. 3.1, for geometrical and natural images, illustrating the effect of normalization. First row: input (on the left); in the middle: negative (bright) and positive (dark) elliptic regions; on the right: hyperbolic (black), $-$parabolic (light gray) and $+$parabolic (dark gray) patches. Second row: the same labeling as above but using normalized measures (Δf and ε_2, both divided by $\varepsilon_1 + 0.001$, for both ε_2 and ε_1 the radial profile is $A(\rho) = (2\pi\rho)^2 \exp(-\pi\rho^2/4\sigma_i^2)$, with $\sigma = 2.6$). The same procedure is repeated below for the "boats" image. Both labeling results were obtained with the same thresholds: $h = 3$, $\alpha_1 = 1.4$, and $\alpha_2 = 1.6$ (see Fig. 3.1). (b) Threshold labeling, as in (a) illustrating the effect of increasing n in Eq. (18). Only negative (bright) and positive (dark) elliptic patches are shown for Δf and ε_2 (in the middle, with $h = 5$, $\alpha_1 = 1.5$, and $\alpha_2 = 2$) and Δf and ε_6 (on the right, with $h = 5$, $\alpha_1 = 2$, and $\alpha_2 = 2$). In both cases we included a Gaussian blurring with $\sigma = 1.8$ pixels.

made by assigning the label corresponding to the highest probability.

The complete relaxation labelling scheme has three components, label, space, and scale compatibility, which can be used separately or in different combinations.

Starting with the probabilities defined in Eq. (24) we use the updating rule

$$p_i^{t+1}(x, y) = \mathcal{N}\{p_i^t(x, y)(1 + q_i^t(x, y))\}, \tag{27}$$

with

$$q_i^t(x, y) = \sum_{j=1}^{N} c_l(i, j)p_j^t(x, y) + \sum_{k=1}^{N} c_n(i, k)n_k^t(x, y) + \sum_{h=1}^{N} c_s(i, h)s_h^t(x, y), \tag{28}$$

where N is the number of labels. C_l, C_n, C_s denote the

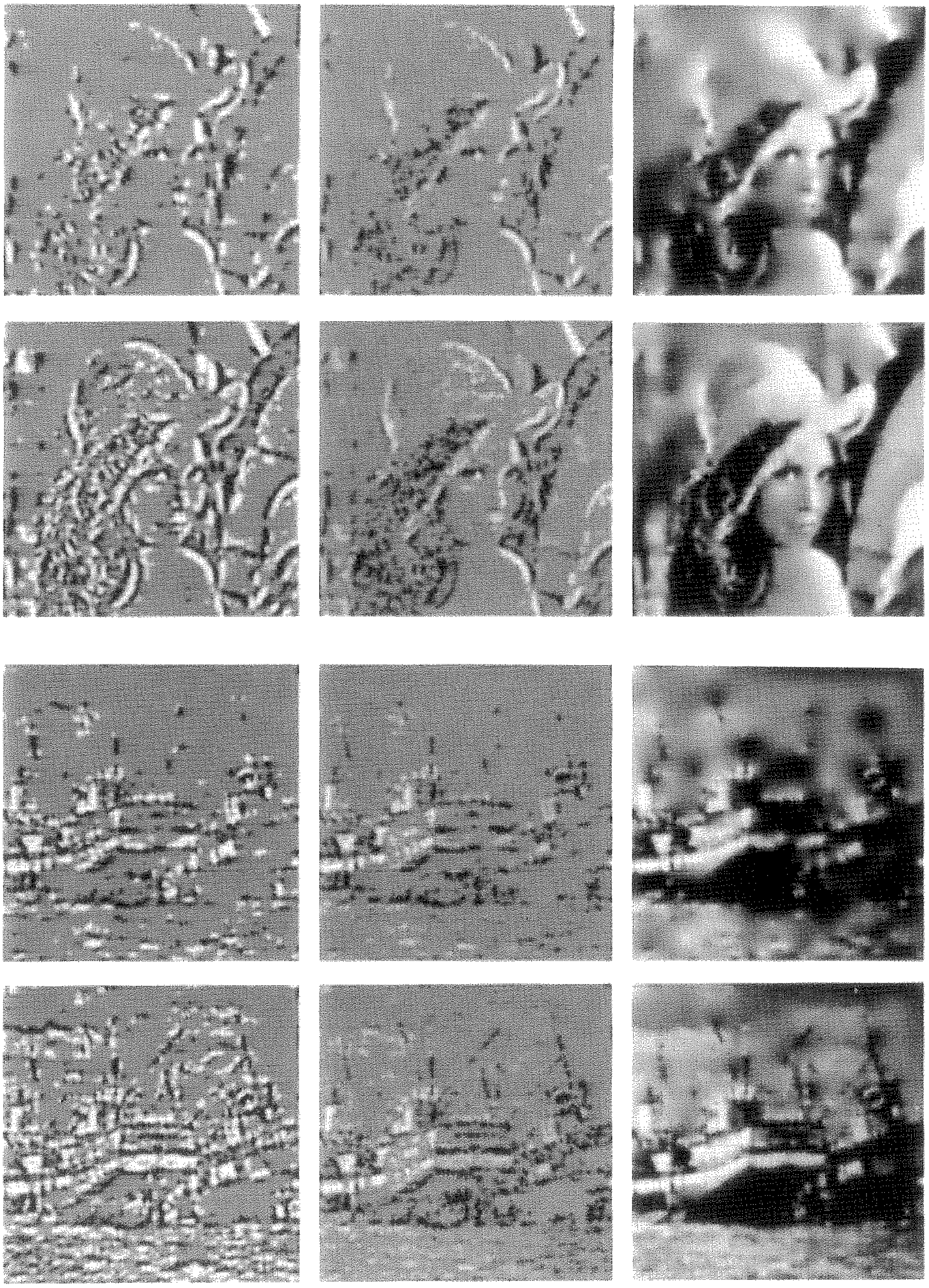

FIG. 3.3. Reconstruction results obtained by spreading elliptic intensities for the "lady" (see Fig. 3.2b) and "boats" (see Fig. 3.2a) images. The labeling on the left (bright and dark for elliptic patches) was used to select the corresponding intensities shown in the middle. This selection reduces the data to 12% (plus 3.2% boundary values) in the example shown in the first row (low entropy reconstruction) and to 22% (+3.2%) in the second row. We have been able to reconstruct the images shown on the right from this restricted set of original pixel intensities using algorithm A2 (7 iterations) followed by A1 (14 iterations). Note that the distinction between + and − elliptic has been done for illustration purpose only. The labeling (+ elliptic, − elliptic, others) was obtained after 7 iterations of the relaxation labeling procedure described in Section 3.1.2 using the following compatibility matrices ($N = 3$):

first row (low entropy),

$$c_l(i, j) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 0.07 \end{pmatrix}, \quad c_n(i, k) = \begin{pmatrix} 0.15 & 0.1 & 0.1 \\ 0.1 & 0.15 & 0.1 \\ 0.1 & 0.1 & 0.15 \end{pmatrix}, \quad c_s(i, h) = 2c_l(i, j);$$

label-, space-, and scale-transition matrices, and n_k, s_h are the neighbor and scale-probability vectors as described below. Normalization $\mathcal{N}$ is achieved by replacing p_i with

$$p_i \Big/ \sum_{i=1}^{N} p_i, \quad i = 1 \ldots N, \tag{29}$$

after each iteration. The algorithm defined by Eqs. (27) and (28) can be seen as a modification of classical relaxation labeling procedures [1].

The updating is meant to achieve a consensus defined by the weights in the transition matrices, which are problem dependent and can be chosen in various ways. This issue is now discussed for each matrix (compatibility component).

3.1.2.1. *Label Compatibility.* We have chosen the probabilities defined in Eq. (24) such that initially all pixels have a highest probability of being planar. Thus the label-compatibility component of the updating procedure has to allow higher order label probabilities to reduce the probability of lower-order labels. Specifically, the weights $c_l(i,j)$ are chosen such that high probabilities $p_{4,5}^t$ (parabolic) will reduce the probabilities p_6^t (planar), and, further, $p_{1,2,3}^t$ (curved) have a chance to win over $p_{4,5,6}^t$. This allows us not only to perform a non-trivial thresholding (curved discrimination lines in Fig. 3.1) in an intuitive way, but also to satisfy other constraints, over space and scale, in parallel. The hierarchy described above will usually require a triangular matrix C_l. Therefore, we can replace N by i in the first sum of Eq. (28).

At this stage the weights $c_l(i,j)$ have been chosen heuristically. Negative values denote incompatibility, for example $c_l(1,4) = -1$ will result in a high probability p_1^t (pit) decreasing the probability p_4^t (valley). However, we suggest that the compatibility coefficients (c_l's) can be optimized, for example, by learning in a neural network, to satisfy global constraints, like obtaining representations with the highest information content at a given entropy. Referring to Section 3.2, this would lead to c_l's where the image can be optimally reconstructed from data with the lowest possible entropy (see Figs. 3.3 and 3.4).

3.1.2.2. *Space Compatibility.* There are various ways to make the labels at a given pixel consistent with the labels at its neighbors. Our basic measures defined in Section 2 already exhibit a low sensitivity to noise, compared to, for example, H and K (see Fig. 2.1). Hence it proved to be sufficient to reduce the neighborhood interaction to only one $N \times N$ transition matrix, by defining the neighbor-probability vector:

$$\begin{aligned} n_i^t(x,y) = \mathcal{N}\{p_i^t(x-1,y) + p_i^t(x+1,y) + p_i^t(x,y-1) \\ + p_i^t(x,y+1)\}, \quad i = 1, \ldots, N. \end{aligned} \tag{30}$$

Note that the sum in Eq. (30) can be extended to eight neighbors or reduced to one neighbor without changing the relaxation labeling procedure defined by Eqs. (27) and (28).

3.1.2.3. *Scale Compatibility.* Space transition enables us to implement constraints like the existence of low probabilities for "peaks" in "valley" regions and the general types of relationships between regions. We now want to make the labeling consistent with the persistence of features at lower scales. For example, we would like to support a pixel to be labeled "peak" or "pit" if the region is elliptic at lower scales.

We have chosen the scale probability vector in Eq. (28) as

$$s_i^t(x,y) = \mathcal{N}\left\{\sum_{l=1}^{L} w_l p_i^0(x,y,s_l)\right\}. \tag{31}$$

This corresponds to a probability vector with each component being a weighted sum over different scales. Therefore, Laplacian and eccentricity are computed on L (typically 1 to 4) lower scales and transformed into a set of L initial probability vectors: $p_i^0(x,y,s_l)$, $l = 1, \ldots, L$ according to Eqs. (22) to (25). Their weighted (w_l's) sum is then renormalized to satisfy the condition (25). Note that

$$s_i^t = s_i^{t-1} = s_i^{t+1} = s_i^0. \tag{32}$$

That is, lower scale probabilities are not updated.

second row,

$$c_l(i,j) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 0 \end{pmatrix}, \quad c_n(i,k) = \begin{pmatrix} 0.1 & 0.07 & 0.07 \\ 0.07 & 0.1 & 0.07 \\ 0.07 & 0.07 & 0.1 \end{pmatrix}, \quad c_s(i,h) = 2c_l(i,j).$$

The initial measures for the scale compatibility vector in Eq. (31) (see also Eq. (22)) were computed on subsampled lower scales ($L = 3$ and all scale weights $w_l = 1$; $\sigma = 1, 2, 4, 8$ pixels) and then expanded using blockwise interpolation.

The same procedure is used for the "boats" image below.

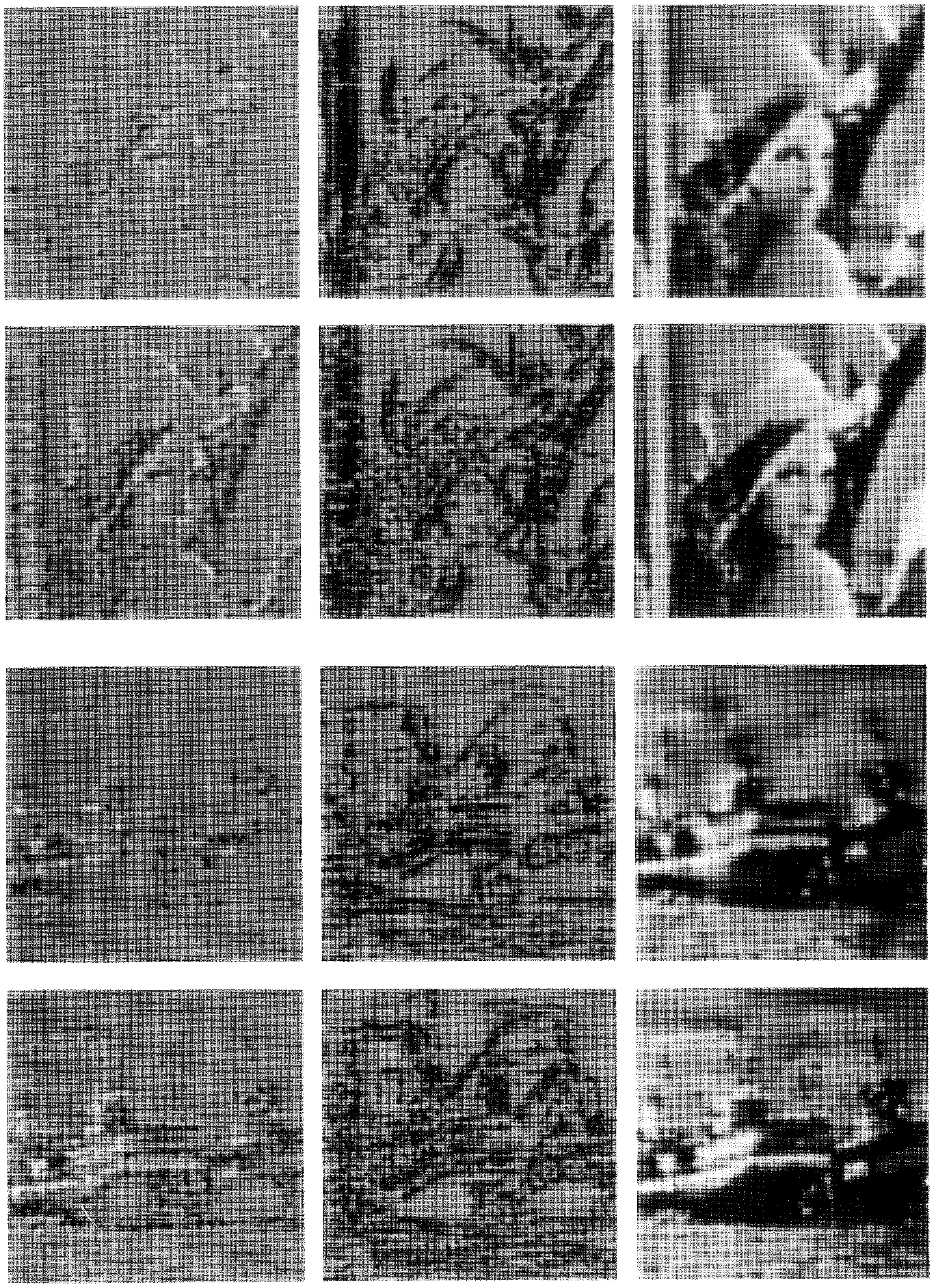

FIG. 3.4. Reconstruction results obtained by labeling elliptic and parabolic patches and spreading elliptic intensities. The partial data on the left (intensities at pixel locations labelled elliptic) were obtained as in Fig. 3.3. The intensities are first spread through the parabolic patches, shown in the middle, according to Algorithm A3 (28 iterations). The new resulting image intensities were then kept fixed, together with the initial intensities shown on the left, and treated like elliptic intensities in Algorithm A1 (42 iterations). Compared to Fig. 3.3 we need additional information about the locations of parabolic pixels, but, on the other hand, the number of original intensities can be further reduced to 3.9% (+3.2%) in the low entropy case (first row) and to 15% (+3.2%) in the second row. The labeling (elliptic, parabolic, others) resulted after 9 iterations with

$$c_l(i, j) = \begin{pmatrix} .5 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 0.01 \end{pmatrix}, \quad c_n(i, k) = \begin{pmatrix} 0.05 & 1.5 & -0.1 \\ -0.1 & 0.5 & -0.1 \\ 0 & -0.1 & 0.1 \end{pmatrix}, \quad c_s(i, h) = c_l(i, j),$$

with the only difference that $c_s(3,3) = 0.15$. The coefficients used for the higher entropy labeling (second row) differ from the above (low entropy)

In conclusion, the region labeling depends on some geometric measures defined by Eq. (22) and up to $3 \cdot N^2$ compatibility coefficients. However, the maximum number of $108 + L$ weights, including the w_l's in Eq. (31), will not be needed for most of the applications. For example, in Section 3.2 we use a segmentation into elliptic and nonelliptic regions ($N = 2$) and reconstruct the image from "elliptic data" only.

Although region labeling is not the main concern of this paper, the results presented in Figs. 3.3 and 3.4 together with the corresponding reconstructed images suggest that our method, initially designed for reconstruction, can be used to segment images into parts which are more interpretable and unique compared to previous labelling methods based upon differential geometry.

3.2. Reconstruction from Partial (2D) Data

In a first stage, we assume that we have the *original* intensity information only at those image positions which are classified as elliptic (partial data). This is to be contrasted with our simulations presented in Section 4, where only curvature information is used.

The simplest approach to the problem is a modification of the classical Gauss–Seidel (sequential) and Jacobi (parallel) relaxation procedures, employing successive overrelaxation insofar as we retain the intensity image values in elliptic regions. Clearly, then, the boundary of such elliptical patches are spread through other patches in accord with spreading (relaxation) conditions which vary as a function of the region types.

By treating "elliptic intensities" like sparse data in traditional reconstruction procedures [16, 17] our first algorithm (A1) is defined as

(A1.1) All elliptical pixels retain their intensity values:

$$r^{t+1}(x, y) = f_{\text{elliptic}}(x, y). \tag{33}$$

(A1.2) All other patches are averaged to simulate soap bubble surfaces, that is, the intensity $r^t(x, y)$ is replaced by

$$r^{t+1}(x, y) = \frac{1}{4}(r^t(x-1, y) + r^t(x+1, y) + r^t(x, y-1) + r^t(x, y+1)). \tag{34}$$

(Modifications are necessary at boundaries. In applications like those illustrated in Figs. 3.3 and 3.4 boundaries were treated like elliptic patches, that is, the original intensity image values were kept fixed.)

However, since soap bubbles cannot develop sufficient rigidity to fill in between elliptic patches, we introduce additional rules, leading to Algorithm A2. Starting with an image labeled elliptic and nonelliptic we have (A2):

(A2.1) All elliptical pixels retain their intensity values Eq. (33).
All relabelled pixels retain the intensity values assigned in A2.2.
(A2.2) For all nonelliptic pixels:
if {it is a neighbor of elliptical pixels or relabeled pixels}
then {replace it by the average of only these neighbors and relabel it}
else {averaging (Eq. (34))}.

Results shown in Fig. 3.3 have been obtained by using (A2) followed by (A1). As illustrated in Fig. 3.4, the amount of data from which we reconstruct can be further reduced by, in addition, using the parabolic patches as in Algorithm (A3):

(A3.1) All elliptical pixels retain their intensity values (Eq. (33)).
All relabeled pixels retain the intensity values assigned in A3.2.
(A3.2) For parabolic pixels:
if {it is a neighbor of elliptical pixels or relabeled pixels}
then {replace it by the average of only these neighbors and relabel it}
else {do A3.3}.

coefficients only in

$$c_s(3,3) = 0.1 \quad \text{and} \quad c_n(i, k) = \begin{pmatrix} 0.03 & 2.4 & -0.06 \\ -0.06 & 0.3 & -0.06 \\ 0 & -0.06 & 0.06 \end{pmatrix}.$$

In addition to the previous example the initial measures were divided by $\varepsilon_1 + 3$ (partial normalization). Again, we used $L = 3$ lower scales and the weights $w_l = 1, 0.5, 0.25$ (from high to low scales).

For the "boats" image we used 4.4% (+3.2%) and 16% (+3.2%), respectively, of the original image intensities resulting from the same labeling parameters as above, with the only difference that $c_l(3,3) = 0.02$ in the low entropy case.

(A3.3) All other patches are averaged to simulate soap bubble surfaces (Eq. (34)).

Note, that, in order to use (A3), images have to be segmented in 3 regions: elliptic ($l_1 \equiv l_2$ in Eq. (21), parabolic ($l_4 \equiv l_5$), and hyperbolic/planar ($l_3 \equiv l_6$). Thus, we reconstruct from elliptical intensities and by knowing the parabolic image positions but not their intensities. Although we use less intensity data, reconstruction improves, especially for long straight patches (see Fig. 3.4).

Finally, we should mentioned that results obtained with the above algorithms depend on the labeling procedures. As already suggested in Section 3.1 the reconstruction result could be used to optimize or, at least, improve the labeling. This could then lead to coding strategies where images can be reconstructed from low entropy data. Since 2D features seem to play an important role in human vision [7, 8], one can expect such coding procedures to be optimal with respect to subjective image quality.

4. IMAGE RECONSTRUCTION BASED ON CURVATURE MEASURES

Though curvature measures provide useful ways of defining local surface geometries invariant to rigid motion they are, when taken alone, not sufficient to reconstruct a surface. Indeed, the fundamental theorem of (surface) differential geometry shows that a surface is uniquely defined and can be reconstructed up to an unknown rigid motion from the first- and second-order fundamental form coefficients: E, F, G, L, M, N (Eqs. (4) and (5)) (for Monge patches f_x, f_y, f_{xx}, f_{yy}, f_{xy}); see, for example, Do Carmo [11] or Spivak [18]. This inverse problem is equivalent to solving the Gauss–Weingarten differential equations and involves the generation of geodesic curves in 3D (which correspond to surface loci). Numerical algorithms for such procedures are well-known and have been applied, for example, to the construction of color space [19] the perceived geometry of color distances.

Our approach to reconstruction differs from such sequential or contour tracking schemes and is based upon parallel relaxation procedures and, more importantly, on the use of partial information about the surface. That is, we have investigated the extent to which images can be reconstructed from 2D regions ($K \neq 0$, or, even more restricted, $K > O$; see Table 1).

In differential geometry it has been known since Minkowski (see [2, 20–22]) that convex surfaces are uniquely defined by their Gaussian curvature values. For Monge-patch surfaces (Eq. (3)) the question of how the determinant D (Eq. (11)) determines the surface is related to the solution of the Monge–Ampere differential equation (see [23, 24]). Further, our problem is related to the more general question of how the metric on a manifold determines the possible embeddings of the manifold into a higher dimensional space. This is because the Gaussian curvature K, Eq. (9), is an intrinsic property of surfaces (*theorema egregium*) and thus it is only for 2D regions that the metric of the surface differs from the Euclidean (flat) metric. The vast mathematical literature on this issue shows that the problem of reconstructing from 2D information is under constraint in general, but solutions are possible under certain conditions [21, 23, 24].

Thus, our approach to reconstruction from curvature measures is more related to this mathematical inverse problem than to image restoration from sparse or noisy data [17] or the integration of information from multiple sources [16].

Similar problems have been addressed in the context of reconstructing images from partial Fourier magnitude or phase [25]. Although phase or magnitude information alone is, in general, not sufficient to uniquely specify an image, it was possible to (a) synthesize a signal in a more-or-less straightforward way from partial information that captures "much of the intelligibility of the signal" (by combining the known information with an "average assumption" about the remainder) (Oppenheim *et al.* [25, p. 1413]) and (b) develop conditions under which exact reconstruction is possible. Much work was devoted to (b), e.g., for developing efficient algorithms [26]. We are more interested in a general algorithm of type (a) and expect further research related to (b). However, although our curvature approach poses similar problems, the nature of the partial information is quite different. Whereas the necessity for reconstructing from partial Fourier-domain information arises from a variety of practical problems, we use reconstruction to motivate curvature extraction.

We employed stochastic relaxation techniques, particularly a parallel version of simulated annealing, which was first applied to optimization by Kirkpatrick *et al.* [27] and is "regarded as a general engine for non-convex problems" [17]. Our aim was to design a general algorithm capable of revealing the amount and type of information contained in different representations.

One graphical way of illustrating this approach is to start with white noise (see Fig. 4.1) and ask the question as to how to update pixels to result in a desired image using only the neighborhood constraints defined with respect to different curvature types. The algorithm, described in some detail in the Appendix, minimizes a cost function (energy) which is generally defined as

$$E_b = (\xi(r^t(x, y)) - \xi(f(x, y)))^2, \qquad (35)$$

where $r^t\,(x, y)$ denotes the actual intensity value at pixel (x, y) after t iterations (relaxation image) and is expected to converge to the original data $f(x, y)$. This should hap-

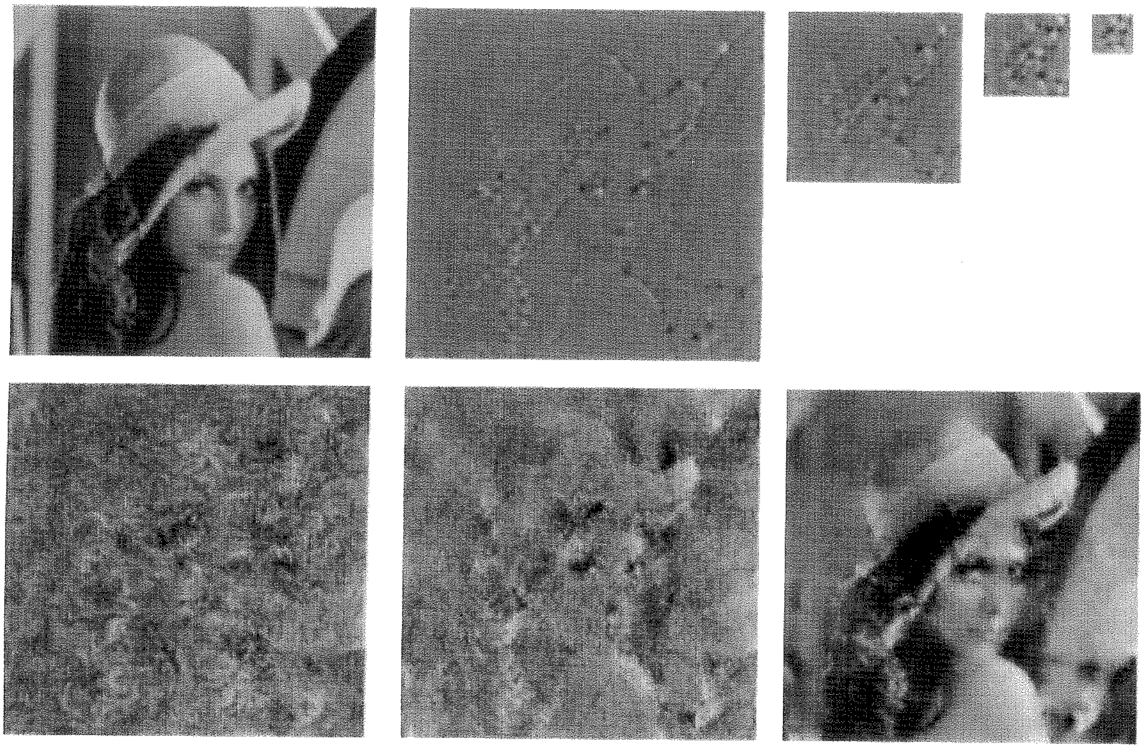

FIG. 4.1. Reconstruction from multiscale curvature information using stochastic relaxation. The clipped eigenvalues of the original image are computed on four subsampled scales and shown as grey level images as in previous figures. The original image was represented by 128 × 128 intensity values. Thus, the input for Algorithm A4 consists of 128 × 128 plus 64 × 64 plus 32 × 32 plus 16 × 16 values of c (see Eq. (16), $n = 2$; scale parameter $\sigma = 1, 2, 4, 8$ pixels). We start from white noise and present results obtained after 200, 400, and 2000 iterations corresponding to different "cooling" stages (see the text and the Appendix for details).

pen under local constraints which propagate during the relaxation process. ξ, the representation operator, is applied to the relaxation image $r^t(x, y)$, and $\xi(f(x, y))$ is the representation from which we reconstruct (we know only $\xi(f(x, y))$ but not the original $f(x, y)$). In our applications the operator ξ extracts local (surface) properties and thus the enforcement of local constraints becomes equivalent to the problem of minimizing E.

However, when reconstructing from only 2D information, results can be considerably improved by using a multiscale representation. In this case the energy can be defined as

$$E_s = \sum_{l=1}^{L} w_l(\xi_l(r^t(x, y)) - \xi_l(f(x, y)))^2, \qquad (36)$$

where l is the scale index and w_l are scale-weights; that is, the local properties are computed on L different scales. In our applications the lower scales were subsampled according to the sampling theorem of Fourier analysis making the algorithm less than two times slower when running on four scales. We have not systematically tested the influence of lower scales. However, in some applications we obtained good results from only one, or even no, additional scale.

Figure 4.1 shows results for ξ corresponding to c (defined in Eqs. (16) and (17), with $n = 2$) and $L = 4$; that is, we constrained the relaxation image r^t (initially white noise) to have the same clipped eigenvalues (Eq. (16)) as the original (see the Appendix for details).

We would like to think of the reconstruction problem as a process of modeling a pellicle (membrane). In Section 3.2 the pellicle was defined by deterministic relaxation algorithms in order to fit partial data. When minimizing cost functions like those defined in Eq. (35), the pellicle has no properties other than those derived as curvature measures from the original image (only even-symmetrical local properties).

We therefore make some additional assumptions (about the world); that is, we think of a pellicle which has some general local properties and we try to "impregnate" this pellicle with curvature information which is, still, the only information we have about the actual image. In other words, additional energy terms are included in Eq. (35) to support the reconstruction process (see the Appendix for details). As in Section 3.2, we have to fill in regions where the curvature is zero. Since the corresponding planar and parabolic patches are developable they would be modelled as "paper surfaces." The problem is that an isometric sheet (paper) is rigid enough to

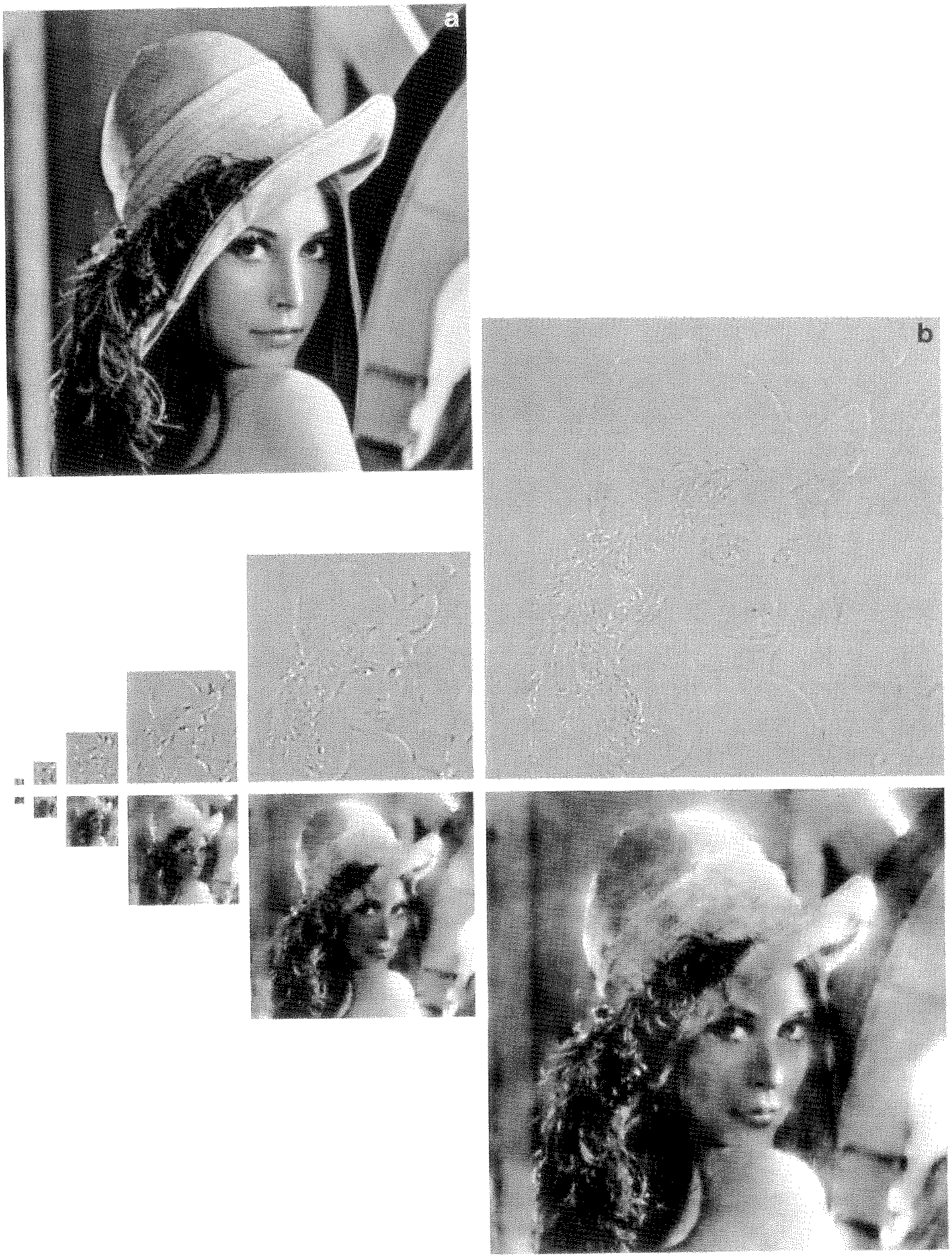

FIG. 4.2. Successive reconstruction from multiscale curvature information using additional constraints and a deterministic acceptance criterion (see the Appendix for details). The original "lady" (a) and "boats" images (c) are represented by 512 × 512 intensity values from which we compute a subsampled curvature-scale space, that is, 512 × 512 plus 128 × 128 plus . . . 16 × 16 values of the clipped eigenvalues operator ϵ (see Eq. (16), $n = 6$; $\sigma = 1$ pixel on the finest scale). As described in the Appendix, reconstruction starts at each scale (with the exception of the lowest, where it starts with a constant-gray-value image) with the expanded version of the next lower scale, resulting in a total of 137 increments per pixel, on average for a 512 × 512 image (see the Appendix for details). For the 128 × 128 image, however, only 111 iterations were needed compared to 2000 iterations of the stochastic relaxation algorithm in Fig. 4.3. A threshold of 1% of the maximum was applied to the clipped eigenvalues, that is, we allowed a dynamic of 1 to 100 for the curvature measures $\xi_l\,(f(x, y))$. However, results are seriously affected only for thresholds greater than 7%.

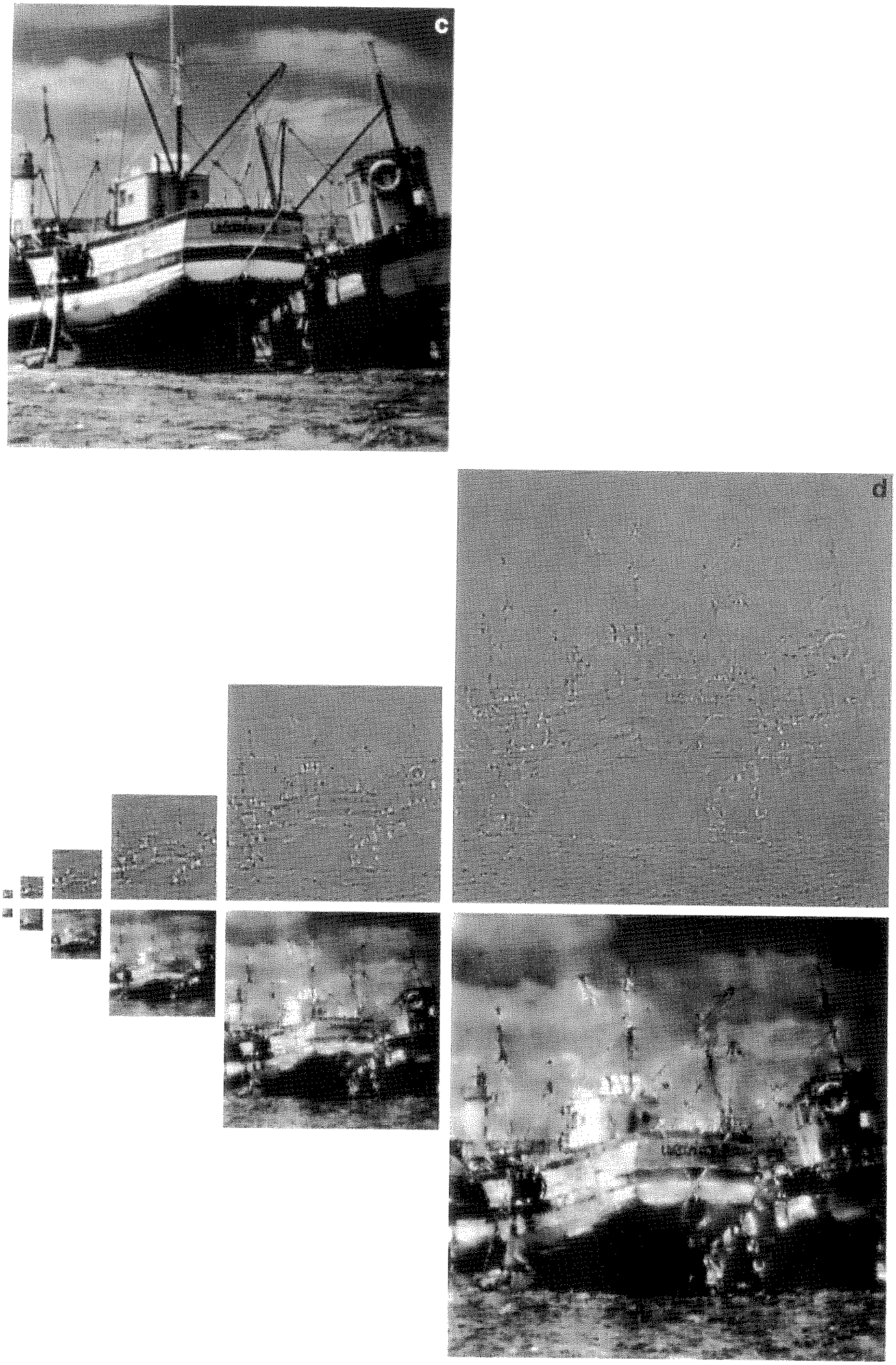

FIG. 4.2—*Continued*

bridge parabolic and planar regions, but, on the other hand, will crease as soon as we try to bend it into an elliptically shaped surface. However, we obtained good results (illustrated in Fig. 4.2) by defining an additional energy term,

$$E_c = (\nabla_{\varphi(t)} r^t(x, y))^2, \tag{37}$$

where $\nabla_{\varphi(t)}$ performs a first-order derivative in direction φ, and φ changes during the relaxation process (see the Appendix for details). Results do not critically depend on the type of continuity constraints. However, well-known energy terms like those for the membrane or plate [17, 28] lead to a higher degree of blurring and somehow inhibit the pellicle as it tries to "blow up."

Surprisingly, it turned out that reconstruction results can be improved by replacing the multiscale energy E_s (Eq. 36) by

$$E_f = \sum_{l=1}^{L} w_l(\xi(r^t(x, y)) - \xi_l(f(x, y)))^2. \tag{38}$$

E_f inforces a certain persistence of elliptic features over scale ("fractal" energy), thus providing an influence of lower scales, although the lower scales are not evaluated in the relaxation process. However, the computational effort is much lower for E_f compared to E_s.

The algorithm illustrated in Fig. 4.2 successively minimizes a cost function defined as the weighted sum of E_f (Eq. (38)) and E_c (Eq. (37)). It turned out that this type of cost function (energy) can be minimized using a deterministic acceptance criterion (see the Appendix for details). From these simulations, it can be seen that reconstruction under these conditions is possible, though further work is required in optimization of the algorithms and parameter states.

5. CONCLUSION

In this paper we have explored how differential geometry and filter theory can be used to capture significant image regions. We have noted that zero-Gaussian curvature surfaces, being developable, always have a direction through which intensities (or depth) are constant and so are not intrinsically "two-dimensional." Further, we have developed new methods for computing such surface characteristics, for example, based on the clipped eigenvalues of the Hessian, instead of the more traditional mean and Gaussian curvature measures. The generalization of the basic Laplacian and eccentricity operators has given us more robust representations and, as initial measures for a relaxation labeling procedure, reasonable region classifications.

A further benefit of this approach for determining important regions of images is that the operators themselves are inherently locally adaptive in contrast to traditional adaptive filtering methods where adaptation is obtained by providing additional "side information" or by gradual approximations over extended spatial regions (see also [10]). The clipped eigenvalues operator (Eq. (16)), for example, can be thought as performing a second order derivative in the direction of the minimum second order derivative. That is, the linear operator "adapts" to the local surface geometry. Also, 2D operators not only provide an efficient representation, but are relevant to biological vision and can be used as prerequisites for certain, for example, topological, invariants [7, 8]. We have only briefly mentioned the possible connections between curvature-scale space and fractals, when modeling the properties of the "reconstruction pellicle" in Section 4. Yet, we suppose that self-similarity is related to the presence of elliptic features over scale and that the fractal dimension could be measured in curvature-scale space [29].

We introduced the 0D, 1D, and 2D signal classes, showing that 2D features are the most significant insofar as we have been able to reconstruct a variety of images from only retaining the 2D regions. Surprisingly, reconstruction also reveals a redundancy of hyperbolic patches in intensity images, which is due to the fact that these regions are usually well approximated by minimal or ruled surfaces.

From the geometric point of view we have shown that the "embedding" (shape) is much more determined by intrinsic curvature properties than one would expect from classical differential geometry. One reason for this is the fact that we use filters on different scales, whereas the geometric results are based on infinitesimal calculus. Second, differential geometry, per se, cannot incorporate relaxation as a method of revealing possible embeddings.

In both labeling and reconstruction phases we have used relaxation techniques which operate simultaneously over space and scale. We suggest that reconstruction algorithms, like those presented in Sections 3 and 4, could be accepted as a general criterion, that is, as a method for determining the usefulness of certain image features and the efficiency of labeling procedures via the achievable quality of the reconstructed image.

Thus, reconstruction can help to find efficient representations, and, thereby, avoid feeding higher order processes with irrelevant or redundant information.

APPENDIX

The stochastic relaxation algorithm based on simulated annealing (and illustrated in Fig. 4.1) has the following basic structure:

A4.1. Compute the initial "energy" E_1 (Eq. (36)) for all pixels.

A4.2. Update all pixel values.

A4.3. Compute the new energy E_2 and the difference $\Delta E = E_2 - E_1$ (cost function).

A4.4. Decide wether to accept the new pixel value, or to restore the initial state, using the Metropolis algorithm, that is,

$$\text{accept if } \{\Delta E < 0\} \quad \text{or} \quad \left\{r < \exp\left(\frac{-\Delta E}{c}\right)\right\}, \tag{39}$$

where r is a random number between 0 and 1 and c is the actual "temperature" (control parameter).

A4.5. Change the value of the control parameter c (cooling or heating) and restart with A4.1 unless a stop criteria is reached.

(A special treatment is necessary at boundaries. In the examples illustrated in Figs. 4.1 and 4.2, the boundary energy was defined as $(f(x, y) - r(x, y))^2$. However, results do not critically depend on the boundary conditions).

We adopted a parallel algorithm, as it is more efficient in our implementations of the energy computation where the filter operations involved are realized via a fast Fourier transform of the image. Due to the growing popularity of simulated annealing and Boltzmann machines, there is a large literature on how to optimize the cooling schedule to obtain best convergence of such an algorithm (see [30]).

In our application it proved to be efficient to start at a low temperature c_0 and to increase the control parameter until about 80% of the changes are accepted. After that, we started the cooling using the same type of updating rules (Eq. (37)) as with heating.

In most applications the updating (A4.2) was defined as

$$r^{t+1}(x, y) = r^t(x, y) +/- i_0, \tag{40}$$

where the sign of the increment i_0 was chosen randomly with equal probability for "+" or "−." $r^t(x, y)$ denotes the actual intensity value at pixel (x, y) after t iterations (relaxation image) and is expected to converge to the original data $f(x, y)$.

However, running A4 as heating, we used a modified acceptance criterion:

$$\text{accept if } \left\{r < \exp\left(\frac{-|\Delta E|}{c}\right)\right\}. \tag{41}$$

Dropping the first branch of the original Metropolis algorithm (Eq. (39)), we ensure that the number of accepted transitions is a purely statistical measure which reflects the actual temperature state without influence from a "downhill component." In (A4.5) the new value of the control parameter at "time" $t + 1$ was usually

$$c^{t+1} = \alpha c^t \tag{42}$$

with $\alpha \geq 1$ (typically between 1.5 and 2) for heating and $\alpha \leq 1$ (0.8 to 0.95) for cooling.

For obtaining the results illustrated in Fig. 4.1, we started with white noise and "heated" with $\alpha = 1.5$ until 80% of the changes were accepted. Then, the steps (A4.1) to (A4.4) were repeated 20 times (unless the number of accepted transitions was less than 45% = stop criterion) for each value of the control parameter before cooling with $\alpha = 0.9$. All pixels were incremented at each iteration as defined in Eq. (40) with $i_0 = 2$. The energy (E_b, $\xi \equiv c$, $n = 2$) was computed on subsampled lower scales which were then added (Eq. (36)) using blockwise interpolation.

The algorithm illustrated in Fig. 4.2 is defined as

A5.1. Start with a constant grey-value image on the lowest scale (here 16 by 16 pixels)

A5.2. Do M iterations of steps (A4.1) to (A4.4) using a deterministic acceptance criterion, that is: accept if $\{\Delta E < 0\}$, the energy E being a (weighted) sum of E_f (Eq. 38) and w_c E_c (Eq. (37)).

A5.3. Decrement the previous increment, as in Eq. (42) with $\alpha = 0.9$, starting with an initial increment i_0, unless a final increment i_f is reached (stop criterion).

A5.4. Expand the relaxation image (e.g., from 16 × 16 to 32 × 32 intensity values) using any type of interpolation, and then go back to A5.2.

The continuity energy E_c (Eq. (37)) was computed by finite differences to approximate a first order derivative which rotates clockwise (with t) and stops at every "hour." It was defined as the squared difference to one of the four neighbors ($(x, y - 1)$, $(x + 1, y)$, $(x, y + 1)$, and $(x - 1, y)$ for 0, 3, 6, and 9 o'clock, respectively) or as the weighted sum of squared differences to two neighbours depending on the direction φ (Eq. (37)).

At each stage (scale, see Fig. 4.2) of the reconstruction we used a "fractal" energy E_f ($L = 1, \ldots 4$, see below) where the lower scale terms were not extrapolated, but added only on the corresponding coarse grid, their influence on the fine grid being ensured by E_c (Eq. (37)).

The parameters used for the different scales were:

Scale	M	i_0	i_f	I	w_1	w_2	w_3	w_c
1 (16 by 16)	24	10	5	168	0	0	0	0.25
2 (32 by 32)	12	8	4	84	0.8	0	0	0.25
3 (64 by 64)	12	6	3	84	0.8	0.75	0	0.25
4 (128 by 128)	12	4	2	84	0.7	0.45	0.15	0.15
5 (256 by 256)	12	4	2	84	0.7	0.45	0.15	0.15
6 (512 by 512)	12	5	2	108	0.7	0.45	0.15	0.15

I is the total number of iterations from i_0 to i_f, and w_1 to w_3 are the energy weights for lower scales in Eq. (38).

ACKNOWLEDGMENTS

We thank Mario Ferraro, Ingo Rentschler, and the reviewers for valuable suggestions.

REFERENCES

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing,* Academic Press, New York, 1982.
2. P. J. Besl and R. C. Jain, Invariant surface characteristics for 3D object recognition in range images, *Comput. Vision Graphics Image Process.* **33,** 1986, 33–80.
3. P. J. Besl and R. C. Jain, Segmentation through variable-order surface fitting., *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-10**(2), 1988, 167–192.
4. A. Jain and R. Hoffman, Evidence-based recognition of objects, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-10**(6), 1988, 783–801.
5. T. J. Fan, G. Medioni, and R. Neviata, Recognizing 3-D objects using surface descriptions, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-11,** 1989, 1140–1157.
6. R. M. Haralick, L. T. Watson, and T. J. Laffey, The topographic primal sketch, *Int. J. Robotic Res.* **2**(1), 1983, 50–72.
7. C. Zetzsche and E. Barth, Fundamental limits of linear filters in the visual processing of two-dimensional signals, *Vision. Res.* **30,** 1990, 1111–1117.
8. C. Zetzsche and E. Barth, Image surface predicates and the neural encoding of two-dimensional signal variations, *Human Vision Electron. Imaging Models Methods Appl.* **1249,** 1990, 160–177.
9. C. Zetzsche and E. Barth, Direct detection of flow discontinuities by 3D curvature operators, *Pattern Recognition Lett.* **12,** 1991, 771–779.
10. C. Zetzsche, E. Barth, and B. Wegmann, The importance of intrinsically two-dimensional image features in biological vision and picture coding, in *Visual Factors in Electronic Image Communication* (Watson, Ed.), MIT Press, 1993, in press.
11. M. P. Do Carmo, *Differential Geometry of Curves and Surfaces,* Prentice–Hall, Engelwood Cliffs, NJ, 1976.
12. D. Marr and E. Hildreth, Theory of edge detection, *Proc. R. Soc. London Ser. B* **202,** 1980, 181–207.
13. B. Wegmann and C. Zetzsche, Statistical dependence between orientation filter outputs used in an human vision based image code, *Visual Comm. Image Process.* **1360,** 1990, 909–923.
14. E. Barth, T. Caelli, and C. Zetzsche, Efficient visual representation and reconstruction from generalized curvature measures., *Geom. Methods Comput. Vision* **1570,** 1991, 86–95.
15. R. Hoffman and A. Jain, Segmentation and classification of range images, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-9**(5), 1987, 608–620.
16. D. Terzopoulos, Multilevel computational processes for visual surface reconstruction, *Comput. Vision Graphics Image Process.* **24,** 1983, 52–96.
17. A. Blake and A. Zisserman, *Visual Reconstruction,* MIT Press, Cambridge, MA/London, 1987.
18. M. Spivak, *A Comprehensive Introduction to Differential Geometry,* Publish or Perish, Boston, MA, 1975.
19. G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae,* Wiley, New York, 1982.
20. Chern, A proof of the uniqueness of Minkowski's problem for convex surfaces, *Am. J. Math.* **79,** 1957, 949–950.
21. N. W. Efimov, *Flaechenverbiegung im Grossen,* Academie, Berlin, 1957.
22. B. K. P. Horn, Extended Gaussian images, *Proc. IEEE* **72**(12), 1984, 1671–1686.
23. A. V. Pogorelev, *Monge–Ampere Equations of the Elliptic Type,* Noordhoff, Groningen, 1964.
24. D. Gilberg and N. Trudinger, *Elliptic Partial Differential Equations of Second Order,* Springer, Berlin, 1983.
25. A. V. Oppenheim, J. S. Lim, and S. R. Curtis, Signal synthesis and reconstruction from partial Fourier-domain information, *J. Opt. Soc. Am.* **73,** 1983, 1413–1420.
26. M. H. Hayes, The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-30,** 1982, 140–154.
27. S. Kirkpatrick, C. D. Gelatt, and M. P. Vechi, Optimization by simulated annealing, *Science* **220**(4598), 1983, 671–680.
28. W. E. L. Grimson, An implementation of a computational theory of visual surface interpolation, *Comput. Vision Graphics Image Process.* **22,** 1982, 39–69.
29. E. Barth, C. Zetzsche, M. Ferraro, and I. Rentschler, Fractal properties from 2D-curvature on multiple scales. *Geometric Methods in Computer Vision II,* SPIE **2031,** 1993, 87–99.
30. E. H. L. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines,* Wiley, Chichester, NY, 1989.