International Journal of Computer Vision, 3, 73–102 (1989)

Constructing Simple Stable Descriptions for Image Partitioning*

YVAN G. LECLERC
Artificial Intelligence Center (Room EK247), SRI International, Menlo Park, CA 94025

*Support for this work was provided by the Defense Advanced Research Projects Agency under contract MDA903-86-C-0084.

Abstract

A new formulation of the image partitioning problem is presented: construct a complete and stable description of an image—in terms of a specified descriptive language—that is simplest in the sense of being shortest. We show that a descriptive language limited to a low-order polynomial description of the intensity variation within each region and a chain-code-like description of the region boundaries yields intuitively satisfying partitions for a wide class of images.

The advantage of this formulation is that it can be extended to deal with subsequent steps of the image understanding problem (or to deal with other attributes, such as texture) in a natural way by augmenting the descriptive language. Experiments performed on a variety of both real and synthetic images demonstrate the superior performance of this approach over partitioning techniques based on clustering vectors of local image attributes and standard edge-detection techniques.

1 Introduction

The partitioning problem is one of the most important unsolved problems in computer vision. In its broadest sense, the task is to delineate regions in an image that correspond to semantic entities in the scene, such as objects and/or coherent physical processes. We shall refer to this as the *scene partitioning problem.* In this sense, the scene partitioning problem is almost isomorphic to the entire image understanding problem and probably cannot be solved unless a solution to the image understanding problem in its entirety is achieved as well.

In the narrower sense used here, the partitioning problem is to delineate regions that have, to a certain degree, coherent attributes in the image. We will refer to this as the *image partitioning problem.* It is an important problem because, on the whole, objects and coherent physical processes in the scene project into regions with coherent image attributes. Thus, the image partitioning problem can be viewed as a first approximation to the scene partitioning problem, and hence a critical first step in solving the image understanding problem. Crucial to the utility of image partitioning in subsequent steps, of course, is the precise definition of "coherent image attributes."

Until recently, most image partitioning techniques took *coherent* image property to mean *homogeneous* image property. That is, most partitioning techniques were designed to identify regions that are homogeneous in some set of local image attributes, such as intensity, color, and texture [8, 13, 21], or to detect the boundary between regions with the assumption that the attributes were locally constant on either side of the boundary [5, 11].

Although *homogeneous* regions are a useful description for a certain class of images, there is a much wider class that is more usefully described as having *piecewise-smooth* image attributes, that is, attributes that are almost everywhere continuous and differentiable up to some specified

low order [1, 4, 10, 12, 14, 16, 17, 19, 20, 26, 27]. For example, images of objects with piecewise-smooth surfaces and albedos are well described in this manner. (For simplicity, we shall restrict our discussion henceforth to a single image attribute, namely, intensity, although the partitioning technique developed here is directly applicable to any attribute that can be represented as either a sparse or dense scalar "image.")

Before we can address the problem of computing a piecewise- smooth description of a real image, we must first define precisely what we mean by such a description. This is nontrivial because real images are spatially discrete and quantized, so that the standard definitions of continuity and smoothness do not apply. The solution proposed here is to model a real image as the *corruption* of an *underlying piecewise-smooth image* and to define this underlying piecewise-smooth image as the desired description.

Intuitively, the underlying piecewise-smooth image is meant to model the image we would have obtained if we had used a perfect pin-hole camera, and if the scene had actually been composed of objects with piecewise-smooth surfaces and albedos. The corruption is meant to model both deviations from this idealized piecewise-smooth model of the scene and corruptions inherent in image sensors. In particular, we model the corruption as convolution with a known point-spread function (to model the point-spread function of the lens of a real camera), followed by sampling, quantization, and the addition of white noise (whose variance is unknown and which might also vary in a piecewise-smooth fashion). The white noise is an approximate model of both the deviations from the piece-smooth model due to small-scale texturing of the objects (which is why we assume that the variance is not uniform) and sensor noise. We will refer to the difference between the real image and the underlying image (after convolution, sampling, and quantization) as the residuals.

Unfortunately, complete information about the underlying image is necessarily lost because of the discrete spatial sampling and intensity quantization. Hence, one can generally hypothesize an infinite number of underlying images that can be corrupted to produce the same real image. The stochastic component of the corruption, of course, only makes the ambiguity worse.

The basic problem addressed here, therefore, is to define criteria by which we can select a unique underlying image for a given real image, and to specify a computationally efficient algorithm for finding this image. We shall call the unique underlying image and its *associated corruption* the "best" description of the real image (the associated corruption is the one that maps the underlying image to the real image).

The formalism in which we pose the problem of finding the best description is that of finding the simplest description of an image, in terms of a specified descriptive language, that is both stable and complete. The formalism is defined in detail in the next section, but, roughly speaking, we take *simplest* to mean shortest description length, *stable* to mean that minor perturbations in the viewing conditions or descriptive language parameters should not alter critical aspects of the description, and *complete* to mean that it should be possible, given the description, to reconstruct the image exactly. The principal components of any solution thus include the specification of the descriptive language and a computationally feasible procedure for selecting a best (i.e., simplest, stable and complete) description.

Our principal contributions in this paper are (1) a formal set of criteria for defining a best description that is applicable not only to image partitioning, but also to much of computer vision; (2) the specification of a descriptive language for piecewise-smooth image partitioning that is very simple, yet yields intuitively satisfying partitions, largely avoiding the gross errors typical of local techniques, even for images with a nontrivial amount of texturing; and (3) a computationally feasible procedure that not only finds the simplest description, but also provides a measure of the stability of the description with respect to perturbations in the image and/or language parameters. It is my hope that the procedure developed here is general enough to be useful for subsequent steps in the computer vision problem.

2 General Framework

The general framework of our approach can be described intuitively as constructing the best description of an image in some specified descriptive language. The choice of descriptive language and what is meant by "best" is, of course, strongly task dependent. However, not all choices are reasonable. We argue that the following criteria are important, and, perhaps, even necessary constraints on the choice of language and the interpretation of best. These criteria constitute the foundation on which rests the specific image partitioning technique developed here and which, we hope, will be the basis for future work on subsequent steps of the computer vision problem, ultimately leading to complete three-dimensional descriptions of a scene and to recognition of the objects contained therein.

2.1 The Criteria

The first criterion is a constraint on the descriptive language alone, namely, that the descriptive language must be *complete.* That is, all descriptions in the language must exactly determine a single image. Thus, what one usually describes as "noise" must be included as part of the descriptive language. Note that completeness means that a given description yields only one image; however, there may be many descriptions of a given image.

The second criterion is a constraint on both the language and the definition of best, namely, *computational feasibility.* This means that the best description of an image (or at least something very close to it) must be constructible in a reasonable amount of time.

Of crucial importance to any system that purports to find the best description of an image is the ability to determine when the image (or, more generally, some portion thereof) lies outside the range of the descriptive language. This leads to two further criteria, which must be satisfied for all (or at least a very large fraction) of the images for which the language is appropriate. Failure to satisfy either of these criteria is a strong indication that the language is inappropriate.

The third criterion, then, is that the best description of an image must be *stable.* The simplest definition of this criterion is: a description of an image is stable when it is unaffected by certain changes in that image. This cannot be used here, however, because descriptions are complete. Hence, *any* change in the image necessarily causes *some* change in the description. Instead, we say that the best description is stable when *some specified portion* of the best description is unaffected by *certain specified classes* of image changes.

The fourth and final criterion is that the best description of an image must be *efficient.* A weak form of this criterion is that the best description must be shorter than the image itself, as suggested by Georgeff and Wallace [9]. A stronger form, which can be defined here only approximately, is that the complexity of the description should not exceed the complexity one would expect for the given image.

2.2 Motivation for the Completeness and Stability Criteria

Although the above criteria are used here only for the development of a specific image partitioning technique, their motivation is much more general. For this reason, and because of its intuitive nature, we use the following example as motivation for the completeness and stability criteria. The final criterion, simplicity of description, is treated in the following subsection.

Consider a complete three-dimensional description of a scene, including a complete camera model, that has been computed from a single image. Clearly, we should expect the volumetric (three-dimensional shape) portion of the description to almost always remain the same, given a new image of the same scene differing only slightly in, say, lighting, surface coloration, or viewpoint. In other words, we should expect the volumetric portion of the description to be stable with respect to the class of image changes corre-

sponding to the aforementioned scene changes, but not, for example, to those corresponding to changes in the shapes of objects.[1]

When supplied with a single image, we cannot test directly for stability by analyzing another image of the scene with slightly different characteristics. (Indeed, for changes like surface coloration, this is not feasible even if we had the opportunity to take a new picture!) Instead, we are left with a somewhat weaker but still crucial expectation: that the volumetric portion of the description should remain the same, given a synthetically generated image derived from a new description—one in which the portions of the new description relating to, say, lighting, surface coloration, or viewpoint are slightly different from the first. This expectation is precisely an example of the stability criterion defined above. Furthermore, we now see the motivation for the completeness criterion: without it, we could not have generated a unique new image from the modified description and hence could not have tested for stability.

Now, it is clearly infeasible to test directly for stability in the above fashion because that would entail generating and analyzing a large number of synthetic images. Instead, one must demand that the language and the algorithm that computes the best description be designed in such a way that together they guarantee stability for the class of images for which the language is designed. Or, at least, they should allow for a computationally inexpensive determination of some measure of stability of the description (perhaps on a part-by-part basis).

To summarize in the abstract, the motivation for stability and completeness is that typical images are the product of a complex combination of many independent (or quasiindependent) processes (i.e., physical processes such as illumination, as well as concrete objects), and our descriptions of images should reflect this. That is, the description should take into account all of the different processes, describing each of them as independently as possible so that changes in one process in the scene are reflected only in changes in the description of that one process.

The above criteria impose constraints on the type of descriptive languages we should strive for, but are not generally sufficient to determine a unique description in a given language for a given image. The criterion we have adopted for the purpose of determining a unique description is that of simplicity, formally called the *minimum-description-length* (MDL) criterion [24]. As we shall see, this criterion is related to the *maximum-likelihood* and *maximum-a-posteriori* (MAP) criteria, but is a more natural criterion when prior probabilities are not well defined.

The use of the MDL criterion is a significantly more general approach than that of regularization theory [22]. Regularization theory deals with so-called *ill-posed* problems (inverse problems that do not have a unique solution) by adding a measure of the solution's *smoothness.* In the MDL approach, smoothness is only one of many possible measures of simplicity.

3 Motivation for the Simplicity Criterion

The idea that simpler descriptions are better than more complex ones is a strongly intuitive notion that was first enunciated as Occam's razor, which counsels us "not to multiply entities beyond necessity." It reflects not only the intuition that simpler descriptions are better because they are easire to use in many ways, but also the body of scientific and personal experience that tells us there is almost always a simpler description of a set of observations than their mere tabulation.

There are two important assumptions behind this notion. The first assumption is that the data are observations of an underlying structured process, and that we could describe these observations in a much simpler fashion by describing them in terms of that process. The second assumption is that the simpler the description, the more likely we are to be describing that underlying process or, at least as far as the observations are concerned, something equivalent to that process.

However, the idea that simpler is better is quite

[1] In effect, Binford [2] calls stability with respect to change in viewpoint the "assumption of general position." In this sense, general position is a special case of our notion of stability.

vague: what exactly does it mean for one description to be simpler than another? One possible answer is that the number of degrees of freedom, or of distinct and independent variables in the description, should be the measure of simplicity. Take, for example, the classical curve-fitting problem, in which one is presented with an ordered set of numerical observations that can purportedly be described as points along some mathematically defined curve. The simplest description, then, should be the one that requires the fewest parameters to define the curve. But, even for such a simple problem, one immediately sees that the definition, as stated, is still somewhat vague.

First, the number of parameters required to define a curve depends very much on the vocabulary of curves one brings to bear. For example, if the observations were actually equally spaced points on a quadratic curve, but one attempted to describe them as the sum of sinusoids (as in a discrete Fourier transform), one would require as many parameters as there are observations. However, a polynomial representation would require only six parameters (three specifying the number of observations, spacing, and order of the polynomial; and three specifying the coefficients of the polynomial), independently of the number of observations. Thus, one would be inclined to say that the polynomial description is the simpler of the two for these observations.

However, if one is allowed to use any possible mathematical curve, one must first specify which of the infinite classes of curves the parameters refer to (polynomials versus sinusoids versus . . .). That is, we must first specify the *language* in which the description is expressed. Since this clearly requires an infinite number of parameters, one is left with the inescapable conclusion that the vocabulary of curves (or, more generally, the language in which the description is expressed) must be restricted in some sense, or else more parameters than observations will always be needed.

A second fundamental problem posed by this definition of simplicity is that almost all phenomena, and hence observations of them, have an inherent stochastic component. At the very least, the observations will be corrupted in some stochastic manner, even if the underlying phenomenon is purely deterministic. Thus, for our curve-fitting example, even if we could specify the underlying curve with a few variables, we would still need to describe the point-by-point deviations from the curve (either directly or in some appropriate parameter space) to obtain a complete description, and this would require at least as many variables as observations! Again we are left with more variables than observations.

The information-theoretic answer to this quandary is to reduce the idea of an independent variable to its simplest form: a bit. The measure of simplicity then becomes the number of bits in the description that some computationally effective procedure can use to reproduce the observations. This is the *minimum-description-length* (MDL) criterion mentioned above. This criterion, of course, demands prior specification of the computationally effective procedure, which is equivalent to specifying the language in which the description is expressed. Thus, in this formalism, the notion of simplicity is a relative one that depends strongly on the choice of descriptive language.

The necessity of providing an a priori descriptive language is a very important and fundamental point. It means that, for a finite number of observations, there is no such thing as an absolute measure of the simplicity of a description; simplicity is inescapably a function of one's prior assumptions.

For example, suppose we assume that the underlying process generating the observations in our curve-fitting problem is the sum of a polynomial (of unknown order) and zero-mean white noise (of unknown variance), and that we wish to find the polynomial with the smallest number of nonzero coefficients compatible with this model. A good descriptive language might then have two components: the first to specify the number of nonzero coefficients and each of their values; the second to specify the variance and point-by-point values of the added white noise. The curve-fitting problem then becomes that of finding the simplest description (the one with the fewest bits) such that the two components add up to the given observations exactly.

One natural choice for the first component is to

assign a fixed number of bits for the specification of the order and for each nonzero coefficient of the polynomial. (The number of bits required is a function of the logarithm of the number of observations, their range, and their precision.) Thus, for this choice of language, polynomials of lower order are simpler to describe than those of higher order.

Since there are provably optimal languages for describing stochastic processes such as white noise, such a language is the natural choice for the second component.[2] With this optimal language, the number of bits required for the second component is roughly proportional to the number of observations times the variance of the point-by-point values.

Thus, with the above descriptive language, there is a natural trade-off between the complexity of the deterministic component (the number of nonzero coefficients) and the complexity of the stochastic component (the variance of the noise): a smaller number of nonzero coefficients reduces the complexity of the first component, but increases the variance of the noise and thus also increases the complexity of the second component; conversely, a larger number of nonzero coefficients increases the complexity of the first component while reducing that of the second.

Rissanen [23] has shown that such a scheme not only produces intuitively pleasing results for observations of real-world processes (when the underlying process is actually unknown), but it is also a MAP solution for a particular class of prior distributions on the parameters of the polynomials.

4 Relationship Between the MAP and MDL Criteria

The advantage of the MDL approach, as I see it, is the uniform manner in which one can combine purely stochastic models (such as white noise) with deterministic models (such as the polynomials above). The approach is very general, and, it is hoped, can be used in subsequent stages of the solution to the computer vision problem.

To see this advantage, and to see the relationship between the MAP and MDL criteria, let us first consider the more traditional MAP criterion. In the abstract, the criterion is to choose the i^{th} model M_i that maximizes the conditional probability of the model, given the data: $P(M_i|D)$. An application of Bayes' rule yields

$$P(M_i|D) = \frac{P(D|M_i)PM_i)}{P(D)}$$

Since $P(D)$ is constant, the MAP strategy is to choose the M_i that maximizes

$$P(D|M_i)P(M_i)$$

For the image partitioning problem, M_i corresponds to the i^{th} piecewise-smooth image (in the set of all possible underlying images), and $D|\mathrm{M_i}$ is the associated corruption.

The first term, $P(D|M_i)$ is straightforward to compute. It is the conditional probability of obtaining the real image, given the underlying image and our model for the corruption. This is simply the probability that the residuals were produced by a white-noise process. Since the noise is uncorrelated, and if we assume, for the purposes of this discussion, that the variance of the noise is known and spatially invariant, this is simply the product of the probabilities of the point-by-point residuals.

The second term, $P(M_i)$, is not as straightforward because it requires specification of the prior probabilities of the piecewise-smooth images. The simplest specification of the prior probabilities is that they are all the same, i.e., $P(M_i)$ is a constant. This leads to the simpler *maximum-likelihood* strategy of choosing the M_i that maximizes $P(D|M_i)$. Unfortunately, the set of piecewise-smooth images is so rich that there is an infinite number of underlying images for which $P(D|M_i)$ is arbitrarily close to one. Thus, the maximum-likelihood strategy is inadequate, and we must find a way of specifying the prior probabilities.

How, then, are we to decide if one piecewise-smooth image is more probable, a priori, than another? We cannot estimate the distribution empirically because the set of all possible piecewise-

[2]An optimal descriptive language is one that minimizes the average number of bits of descriptions per bit of input. This will be discussed in detail shortly.

smooth images (or equivalently, the set of all possible scenes from which these images are derived) is much too large. We are therefore left with simply defining the prior probabilities in order to meet some criterion or other that we choose. For example, we could attempt to define the prior probabilities in such a manner that the MAP criterion selects the smoothest underlying image with the fewest discontinuities for which the residuals are indistinguishable from white noise. As we shall see, this is equivalent to defining a descriptive language and using the MDL criterion.

To see this, let us denote the language for describing the models M_i as $\mathscr{L}_m(M_i)$, the language for describing the data, given a model, as $\mathscr{L}_c(D|M_i)$, and the number of bits in a description as $|\cdot|$. The MDL strategy, then, is to choose the M_i that minimizes

$$|\mathscr{L}_c(D|M_i)| + |\mathscr{L}_m(M_i)|$$

When we know the prior probabilities of the thing we are describing, information theory tells us that we can design an optimal descriptive language that minimizes the expected number of bits per description [23]. For such optimal languages, the number of bits in the description equals the negative base-two logarithm of the probability of the thing being described. For example, if we knew the prior probabilities of the models above, and if $\mathscr{L}_m^*$ is the optimal descriptive language, then

$$|\mathscr{L}_m^*(M_i)| = -\log_2 P(M_i)$$

Similarly,

$$|\mathscr{L}_c^*(D|M_i)| = -\log_2 P(D|M_i)$$

and the MDL strategy can be rewritten as choosing the M_i that minimizes

$$-\log_2 P(D|M_i) - \log_2 P(M_i)$$

This is equivalent to the MAP strategy of maximizing $P(D|M_i)P(M_i)$.

Thus, we see that, by choosing optimal descriptive languages for given prior probabilities, the MDL strategy is equivalent to the MAP strategy. *Conversely,* if one assumes the prior probabilities implicitly specified by the given descriptive languages, the MAP strategy is equivalent to the MDL strategy. The choice of strategies depends on whether it is easier or more natural to specify a descriptive language directly or to specify prior probabilities.

For the image partitioning problem, it is more natural to specify a descriptive language for the underlying piecewise-smooth images (namely, the discontinuities and low-order derivatives of the underlying image), but it is more natural to specify the prior probabilities of the residuals (since the unknown component is a well-understood stochastic process, namely, white noise). Consequently, our strategy is to choose the M_i that minimizes

$$-\log_2 P(D|M_i) + |\mathscr{L}_m(M_i)|$$

That is, we choose the M_i that minimizes the number of bits required to describe the residuals (as defined by the statistical distribution of the residuals, $P(D|M_i)$) plus the number of bits required to describe the underlying image (in terms of discontinuities and low-order derivatives).

It is not necessary to actually describe either the residuals or the models in their optimal languages. All we need do is compute the number of bits it would have taken to describe them had we actually used the optimal languages. This is in fact what we do for the image partitioning algorithm described in the following sections.

5 The Piecewise-Constant Case

An an introduction to the more general piecewise-smooth image partitioning problem, and as a tutorial on the steps involved in solving the more general problem, let us consider the simpler case in which a real image is the sum of an underlying piecewise-constant image and white noise with known variance.

We denote the real $n \times m$ image by the vector $\mathbf{z}$ indexed by $i \in I = 1, \ldots, nm$. Using a single index like this significantly simplifies the notation for this and the more general piecewise-smooth case. One can think of i either as an integer representing the i^{th} pixel in the image for some ordering of the pixels, or as a vector belonging to the set $\{1, \ldots, n\} \times \{1, \ldots, m\}$.

For computational reasons, we represent the underlying image $u(x,y)$ by a grid of square 1×1

elements, with each element centered at the coordinate (x_i,y_i) of the i^{th} pixel in the real image. The 1×1 square centered at (x_i,y_i) is the *spatial domain* $\mathcal{S}_i$ of the i^{th} element, and the value of the element is u_i. Thus,

$$u(x,y) = u_i \quad \forall\,(x,y)\in \mathcal{S}_i, \quad i \in I$$

and the underlying image is completely represented by the vector $\mathbf{u} = \{u_i,\ i \in I\}$.

Similarly, we represent the noise by the vector $\mathbf{r}$. Thus, the statement that the real image is the sum of the underlying image and the noise can be written as

$$\mathbf{z} = \mathbf{u} + \mathbf{r} \tag{1}$$

A consequence of this choice of representations is that discontinuities in the underlying image can occur only along the vertical and horizontal boundaries between the grid elements. One advantage of this is that the underlying image is uniquely specified when there is no noise (namely, $\mathbf{u} = \mathbf{z}$). However, a more sophisticated representation in which elements have variable shape is also possible. This is an excellent avenue for future research.

Using the above definitions, the problem of finding the simplest description is therefore:

$$(\mathbf{u}^*, \mathbf{r}^*) = \min_{(\mathbf{u},\mathbf{r}):\mathbf{z}=\mathbf{u}+\mathbf{r}} |\mathcal{L}_u(\mathbf{u})| + |\mathcal{L}_r(\mathbf{r})|$$

where $\mathcal{L}_u$ and $\mathcal{L}_r$ denote the languages used to describe $\mathbf{u}$ and $\mathbf{r}$. From equation (1), the equivalent problem is

$$\mathbf{u}^* = \min_{\mathbf{u}} |\mathcal{L}_u(\mathbf{u})| + |\mathcal{L}_r(\mathbf{z} - \mathbf{u})|$$

There are two steps involved in solving this problem. First, we must define the languages $\mathcal{L}_u$ and $\mathcal{L}_r$. Second, we must specify a computationally feasible procedure for finding $\mathbf{u}^*$ and for determining the stability of the solution.

5.1 Defining Descriptive Languages

The first task, then, is to define a language for describing the underlying piecewise-constant image $\mathbf{u}$. By definition, $\mathbf{u}$ is composed of regions of constant intensity. Thus, for each region, we need specify only the shape and position of the region boundaries and the constant intensity within the region. Clearly, if we had strong prior expectations about the shape of these region boundaries (e.g., we might know that they are composed of long straight-line segments only), or about relationships among regions, then we could use these prior expectations to define the language for describing the regions. This is a topic I hope to explore in future work. For this paper, however, a very simple yet general-purpose language is used.

Specifically, the region boundaries are described by a chain code of unit-length line segments located between adjacent elements; each line segment corresponds to the boundary between adjacent square grid elements. The number of bits required to describe each region is thus proportional to the number of elements in the chain plus a constant to specify the constant intensity and the first element of the chain. The total number of bits required to specify the underlying image is thus proportional to the number of regions plus the total length of the region boundaries.

Since region boundaries occur only when spatially adjacent elements of $\mathbf{u}$ are different, their total length can be determined locally by counting all adjacent elements (u_i, u_j) that have a nonzero difference and dividing by 2 (since region boundaries will be counted twice this way). Thus, the total length of the region boundaries is

$$\frac{1}{2}\sum_{i\in I}\ \sum_{j\in N_i} (1 - \delta(u_i - u_j))$$

where

N_i = the set of 4(or8)-connected neighbors of the i^{th} element

$\delta(x)$ = the Kronecker delta = $\begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$

When the regions are relatively large, a good approximation to the number of bits required to describe $\mathbf{u}$ is thus

$$|\mathcal{L}_u(\mathbf{u})| \approx \frac{b}{2}\sum_{i\in I}\ \sum_{j\in N_i} (1 - \delta(u_i - u_j)) \tag{2}$$

where b is the sum of (1) the number of bits required to encode each element in the chain code and (2) the number of bits required to encode the constant intensity and starting element, divided by the average region-boundary length. For example, for 4- connected elements, there are only 3 possible directions for each new element of the

chain code. Each chain code element can thus be encoded by using somewhere between $\log_2 3 (\approx 1.585)$ and two bits (the lower limit can only be achieved when encoding infinitely long chain codes). Thus, for 4-connected elements, b should be at least as large as $\log_2 3$, but not much more than two.

The disadvantage of 4-connected elements is that diagonal boundaries are more expensive to encode than horizontal or vertical boundaries of the same Euclidean length. This rotational asymmetry can be approximately countered by using 8-connected elements and weighting the diagonal discontinuities by $1/\sqrt{2}$. The cost of encoding a region boundary is then more closely proportional to the Euclidean length of the boundary, which is rotationally invariant. Eight-connected elements with this weighting scheme are used for all of the examples here. For simplicity, however, the notation of equation (2) is retained.

As for describing the noise, recall that the fewest bits required to describe data generated by a stochastic process is the negative base-two logarithm of the probability of observing that data. Since we assume the noise to be uncorrelated,

$$|\mathcal{Y}_r(\mathbf{r})| \equiv -\log_2 P(\mathbf{r}) = -\log_2 \prod_{i \in I} P(r_i)$$
$$= -\sum_{i \in I} log_2 P(r_i)$$

Furthermore, we assume the noise to be quantized white noise, where the elements are drawn from a normal distribution and then quantized to the nearest q, the precision of the pixels in the real image. Thus,

$$P(r_i) = \int_{|r_i|q}^{|r_i|q} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-x^2}{2\sigma^2}\right) dx$$
$$\approx \frac{q}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-r_i^2}{2\sigma^2}\right)$$

when $q < \sigma$ (3)

and

$$-\log_2 P(\mathbf{r}) \approx nmc + a \sum_{i \in I} \left(\frac{r_i}{\sigma}\right)^2 \quad (4)$$

where

$$a = \frac{1}{2 \log 2}$$

$$c = \frac{1}{\log 2}\left(\frac{1}{2}\log 2\pi + \log \rho - \log q\right) \quad (5)$$

Thus, for $\mathbf{u}$ and $\mathbf{r}$ satisfying Equation 1, an approximation to the total number of bits required to describe $\mathbf{u}$ and $\mathbf{r}$ is

$$|\mathcal{Y}_u(\mathbf{u})| + |\mathcal{Y}_r(\mathbf{r})| \approx nmc + L(\mathbf{u})$$

where

$$L(\mathbf{u}) = a \sum_{i \in I} \left(\frac{z_i - u_i}{\sigma}\right)^2$$
$$+ \frac{b}{2} \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(u_i - u_j)) \quad (6)$$

Dropping the additive constant, the minimization problem can thus be written as

$$\mathbf{u}^* = \min_u L(\mathbf{u})$$

To re-emphasize the origins of this function, note that the term on the right of L depends on the equality of adjacent elements only. Thus, every $\mathbf{u}$ can be characterized by the regions of contiguous equal-valued elements in the image. For a given set $R = \{R_r\}$ of such regions, where, by definition

$$\bigcup_r R_r = I$$

and u_r is the intensity within region R_r, $L(\mathbf{u})$ can be written as

$$L(R) = a \sum_r \sum_{i \in R_r} \left(\frac{z_i - u_r}{\sigma}\right)^2$$
$$+ b \sum_r \text{(length of boundary of } R_r)$$

Thus, for a fixed set of regions, $L(R)$ is a quadratic with the unique global minimum

$$u_r^* = \frac{\Sigma_{i \in R_r} z_i}{\Sigma_{i \in R_r} 1} \quad \forall\, r$$

That is, the intensity of the ideal image within each region equals the mean of the real image within that region, as we would expect for a problem involving white noise.

5.2 Defining a Computationally Feasible Procedure

The simplest, direct way of finding the global minimum of $L(R)$ is to search through all possible sets of regions, calculating the cost for each set, and choosing the set with the smallest cost. Unfortunately, the number of possible sets of regions grows exponentially with the number of elements of $\mathbf{u}$, rendering such a search completely infeasible. Even dynamic programming-like algorithms require at least the evaluation of the cost for every possible simple region, which is an exponential in nm when n and m are greater than 1, again rendering such a search computationally infeasible.

Furthermore, because of the Kronecker delta term, $L(\mathbf{u})$ has many local minima. Thus, standard descent-based optimization techniques are useless. Also, the simulated-annealing style of algorithms exemplified in Geman and Geman [8] are inappropriate, because the time complexity is much too high for this type of function [3]. Intuitively, the reason that stochastic gradient-descent algorithms are inappropriate for this particular objective function is that the function has extremely narrow (in fact, infinitesimally narrow) valleys, so that even stochastic sampling of the surface provides no guidance for the search.

Instead, I have devised an algorithm that yields something very close or equal to the optimal solution for a large class of inputs. It belongs to a class of optimization techniques generally called continuation methods [6, 29]. This algorithm is similar in spirit to the algorithm described in Blake and Zisserman [4] as the "graduated non-convexity," or GNC algorithm.

As used here, a continuation method embeds the objective function in a family of functions $L(\mathbf{u}, s)$ for which there is a single local minimum at some large s, and for which the number and position of the local minima converge to those of $L(\mathbf{u})$ as s approaches zero. The steps of the continuation method are straightforward. First, find the unique local minimum $\mathbf{u}^0$ of $L(\mathbf{u}, s^0)$ for some sufficiently large s^0. Then, track the local minimum in $\mathbf{u}$ as a decreasing function of s, as follows. For $s^{t+1} = s^t$, let $\mathbf{u}^{t+1}$ by the result of taking a single step of a descent algorithm, as applied to the objective function $L(\mathbf{u}, s^{t+1})$ started at $\mathbf{u} = \mathbf{u}^t$. When the descent algorithm converges, let $s^{t+1} = rs^t$ for some $0 < r < 1$, and repeat until s^t is sufficiently small. For an ideal embedding, there will be no bifurcations along this path, and the value of $\mathbf{u}^t$ for a sufficiently large t (and hence a sufficiently small s^t) will be close or equal to the global minimum of $L(\mathbf{u})$.

The specific embedding used here replaces $\delta(u_i - u_j)$ with an exponential,

$$\delta(u_i - u_j) \rightarrow e_{i,j}(\mathbf{u}, s) \equiv \exp\left[- \frac{(u_i - u_j)^2}{(s\sigma)^2} \right]$$

so that

$$L(\mathbf{u}, s) = a \sum_{i \in I} \left(\frac{z_i - u_i}{\sigma} \right)^2 + \frac{b}{2} \sum_{i \in I} \sum_{j \in N_i} (1 - e_{i,j}(\mathbf{u}, s)) \qquad (7)$$

This is an appropriate embedding because

$$\lim_{s \to 0} e_{i,j}(\mathbf{u}, s) = \delta(u_i - u_j)$$

so that

$$\lim_{s \to 0} L(\mathbf{u}, s) = L(\mathbf{u})$$

and hence the local minima of $L(\mathbf{u}, s)$ approach the local minima of $L(\mathbf{u})$. Furthermore, there exists a unique local minimum of $L(\mathbf{u}, s)$ for sufficiently large s, namely $\mathbf{u} = \mathbf{z}$. This is so because (1) $L(\mathbf{u}, s) \geqslant 0 \; \forall \, \mathbf{u}$; (2) $\mathbf{u} = \mathbf{z}$ is the unique point for which the first summation of equation (7) is identically zero; and (3) the second summation vanishes for arbitrarily large s when $\mathbf{u}$ is bounded. Thus, for s approaching infinity, $\mathbf{u} = \mathbf{z}$ is the unique point for which $L(\mathbf{u}, s) = 0$, the unique local (and global) minimum.

Intuitively, the exponential term introduces broad valleys when s is large, and converges to the narrow valleys in the limit as s goes to zero. Thus, the continuation method creates a kind of "scale space" representation of the objective function $L(\mathbf{u})$ (in analogy to Witkin's scale-space representation of a signal [28]) and tracks a local minimum from the coarsest scale (where there is only one local minimum) to the finest scale (where there are many). A complete discussion of the bifurcations that may occur along the path of the local minimum, and a direct comparison of the results of this continuation method with the global optimum when n or m is one (where a dynamic-programming solution is feasible) is presented in [15]. Briefly, the comparison shows

that for sufficiently large signal-to-noise ratios (where "sufficiently large" is a function of the distance between discontinuities), the continuation method always finds the global minimum. In any case, the encoding cost found by the method is almost always within 2–3 percent of the cost at the global minimum. Experimental results for larger images are discussed in the next section.

Although any iterative descent algorithm can be used for the continuation method (see, for example, the wide variety described in Luenberger's excellent book [18]), the following algorithm has proven to be quite efficient for the objective function examined here.

By definition, local minima of $L(\mathbf{u}, s)$ occur when, for all $i \in I$,

$$\frac{\partial L(\mathbf{u}, s)}{\partial u_i} = \frac{2a}{\sigma^2}(u_i - z_i) + \frac{2b}{(s\sigma)^2}\sum_{j \in N_i} e_{i,j}(\mathbf{u}, s)(u_i - u_j) = 0 \tag{8}$$

which can be written in vector notation as

$$\nabla L(\mathbf{u}, s) \equiv \frac{\partial L(\mathbf{u}, s)}{\partial \mathbf{u}} = \mathbf{b} + \mathbf{A}(\mathbf{u}, s)\mathbf{u} = \mathbf{0} \tag{9}$$

where

$$a_{i,i}(\mathbf{u}, s) = \frac{2a}{\sigma^2} + \frac{2b}{(s\sigma)^2}\sum_{j \in N_i} e_{i,j}(\mathbf{u}, s)$$

$$a_{i,j}(\mathbf{u}, s) = \begin{cases} \dfrac{-2b}{(s\sigma)^2} e_{i,j}(\mathbf{u}, s) & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases}$$

$$b_i = \frac{-2az_i}{\sigma^2}$$

At each step of the iterative descent algorithm, we linearize the above set of equations by setting $s^{t+1} = s^t$ and fixing $\mathbf{A}^t \equiv \mathbf{A}(\mathbf{u}^t, s^{t+1})$. Since $\mathbf{A}^t$ is diagonally dominant, a Gauss-Seidel iterate can be used to provide a step in the direction of the solution:

$$u_i^{t+1} = \frac{-1}{a_{i,i}^t}\left(b_i + \sum_{j \neq i} a_{i,j}^t u_j^t\right)$$

$$= \frac{z_i + \dfrac{b}{a(s^{t+1})^2}\sum_{j \in N_i} e_{ij}\, u_j^t}{1 + \dfrac{b}{a(s^{t+1})^2}\sum_{j \in N_i} e_{i,j}^t} \quad \forall i \in I \tag{10}$$

where

$$e_{i,j}^t \equiv e_{i,j}(\mathbf{u}^t, s^{t+1})$$

The above is repeated until $|u_i^{t+1} - u_i^t|$ is sufficiently small (less than $0.1s^{t+1}\sigma$) for all i; only one or two iterations are typically required to achieve this accuracy. Once convergence has been achieved, s is decreased ($s^{t+1} = rs^t, 0 < r < 1$), and everything repeated until s^{t+1} is sufficiently close to zero. Ideally, r should be arbitrarily close to one to guarantee that the correct local minimum is tracked. Also, the closer r is to one, the closer the starting point $\mathbf{u}^t$ will be to a local minimum after decreasing s, and hence the fewer Gauss-Seidel iterations will be required. However, making r closer to one increases the number of times s must be decreased to achieve a given small value. A good compromise between accuracy and computation time, as we shall see below, is $r = 0.95$.

When the *interaction strength* $e_{i,j}^t$ falls below $1/e$ (i.e., when $|u_i^t - u_j^t| < s^{t+1}\sigma$), we say that a (tentative) discontinuity between adjacent elements has been found at time t. The discontinuity is called tentative because it is possible (though relatively rare) for the interaction strength to oscillate a few times before converging to a stable value. The word "tentative" will be dropped unless ambiguity would result. The first value of s^{t+1} for which this occurs is called the *stability*, $s_{i,j}$, of the discontinuity (we shall shortly see why).

5.3 Discussion

To arrive at an intuitive understanding of both the objective function and the continuation method, recall from equation (8) that at a local minimum (i.e., for those times t at which the descent algorithm has converged), $\mathbf{u}$ satisfies

$$u_i - z_i + \frac{b}{a(s^{t+1})^2}\sum_{j \in N_i} e_{i,j}^t(u_i - u_j) = 0$$

When $|u_i^t - u_j^t| \ll s^{t+1}\sigma$ for all i, then $e_{ij}^t \approx 1$ for all i, and the above equation is the discrete form of the Euler-Lagrange equation

$$u(\mathbf{x}) - z(\mathbf{x}) - \lambda^2 \nabla^2 u(\mathbf{x}) = 0$$

for $\lambda^2 = b/a(s^{t+1})^2$. Far from the boundaries of the image, the Green's function (or impulse response function) for this equation is [4]

$$G(\mathbf{x}, \mathbf{x}', \lambda) = \frac{1}{2\pi\lambda^2} K_o \left(\frac{|\mathbf{x} - \mathbf{x}'|}{\lambda} \right)$$

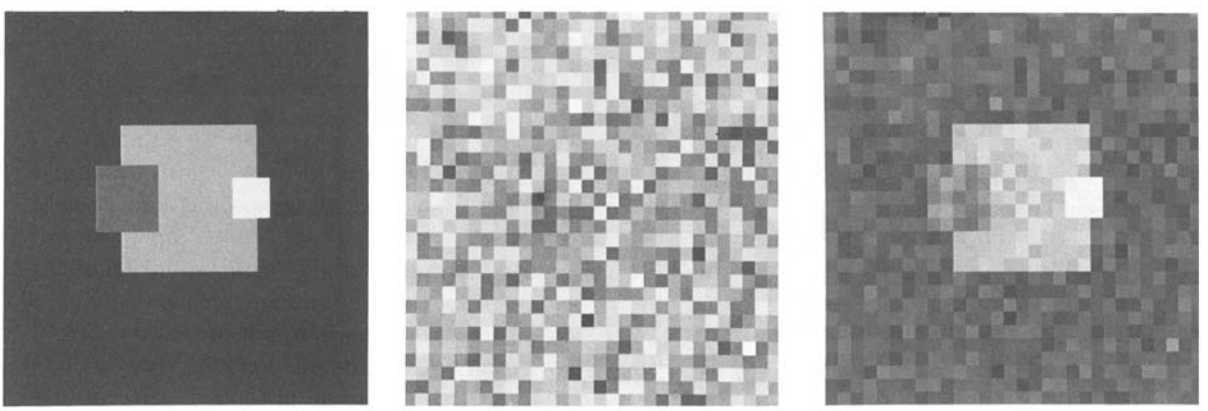
Thus, far from the image boundaries, local maxima in L correspond approximately to the convolution of the image data with $G(\mathbf{x},\mathbf{x}',\lambda)$, which is a strictly positive circularly symmetric function whose spatial scale is inversely proportional to s^t. Hence, as s^t becomes smaller, the spatial scale of G increases, and $\mathbf{u}^t$ becomes smoother.

Near those elements where the interaction strength is significantly less than one (i.e., near the discontinuities in $\mathbf{u}^t$), the functional form of the Green's function is complex, but it can be determined numerically from equation (9) as a column of $(\mathbf{A}^t)^{-1}$. As before, the spatial scale of the Green's function is inversely proportional to s^{t+1}. However, the function is not circularly symmetric; instead it is "adapted" to the interaction strengths, so that smoothing does not directly "spill across" discontinuities.

In other words, we can view the continuation method as a kind of adaptive smoothing of the real image. At first, the spatial scale of the smoothing filter is small and the filter is spatially invariant. As the iterations proceed, the spatial scale increases, but the filter adapts itself to discontinuities found at previous iterations; the adaptation being to not smooth across these discontinuities.

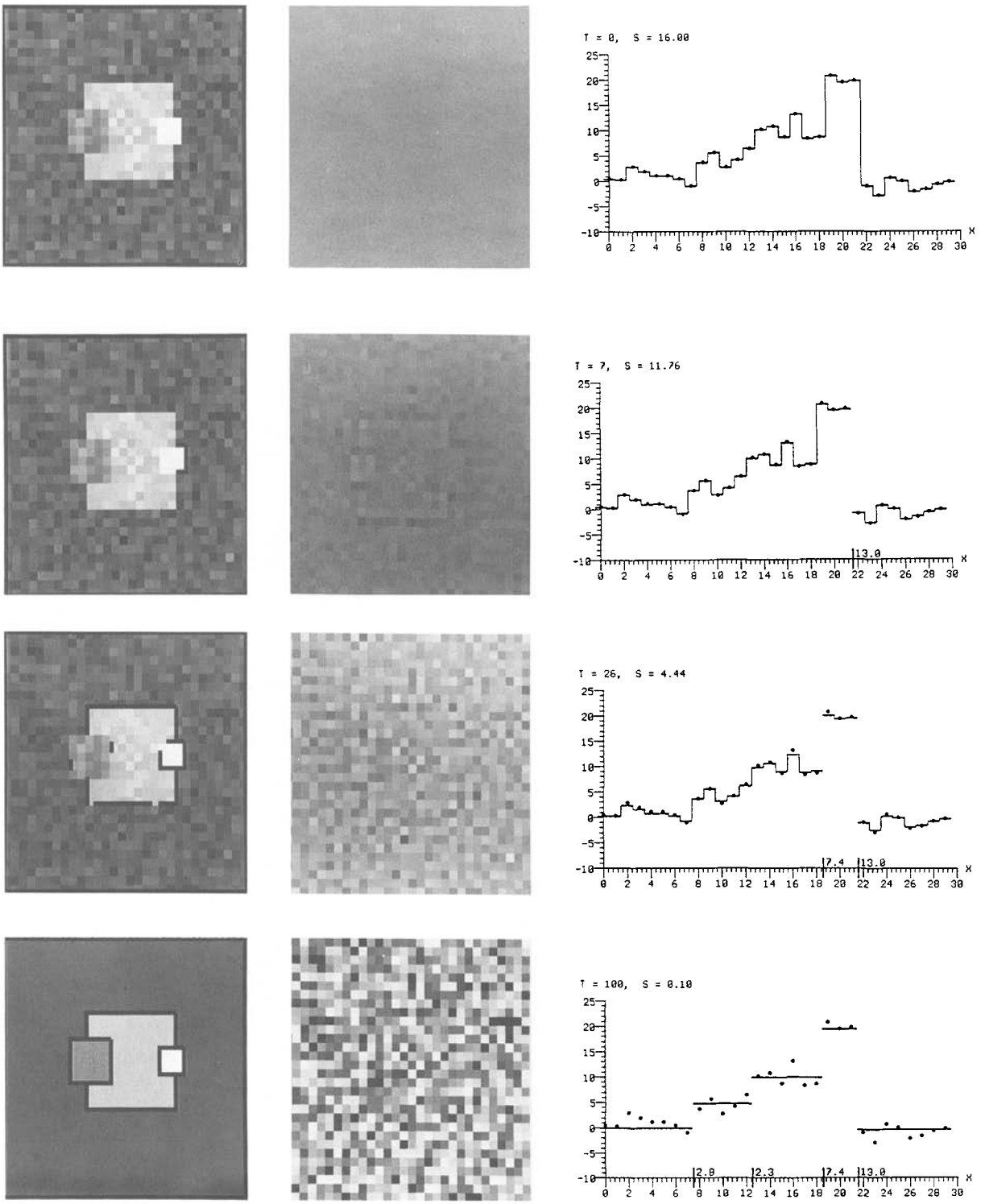
To see the above points graphically, consider an application of the continuation method to a synthetic piecewise-constant image (figure 1), of which several steps are illustrated in figure 2. Note how $\mathbf{u}^t$ becomes smoother as s^t decreases, except at discontinuities. Also observe that the stability of a discontinuity is a function of both the local contrast in the image and the size of the two regions to which the discontinuity belongs. For example, the stability of the boundary between the right square and the background, as well as most of the boundary between the middle square and the boundary, is approximately equal to the ratio of the local contrast to σ. Thus, when the contrast is sufficiently large relative to σ, or when the boundary is between large regions, the stability measure corresponds approximately to a local measure of the signal-to-noise ratio.

For smaller regions, or when the contrast is less, the stability measure can be significantly lower than the local signal-to-noise ratio. In fact, discontinuities can disappear entirely when the signal-to-noise ratio is sufficiently small for a given region size. In a sense, then, the stability



(a) (b) (c)

Fig. 1. A piecewise-constant function embedded in white noise. (a) The actual underlying piecewise function. The background, left, middle, and right squares have intensity 0.0, 5.0, 10.0, and 20.0, respectively. (b) White noise with $\sigma = 1.5$. (c) The sum of (a) and (b).



(a)

Fig. 2. (a) Several steps of the continuation-method. Each step shows the current estimate of the underlying image, the residuals, and a graph of the center row.

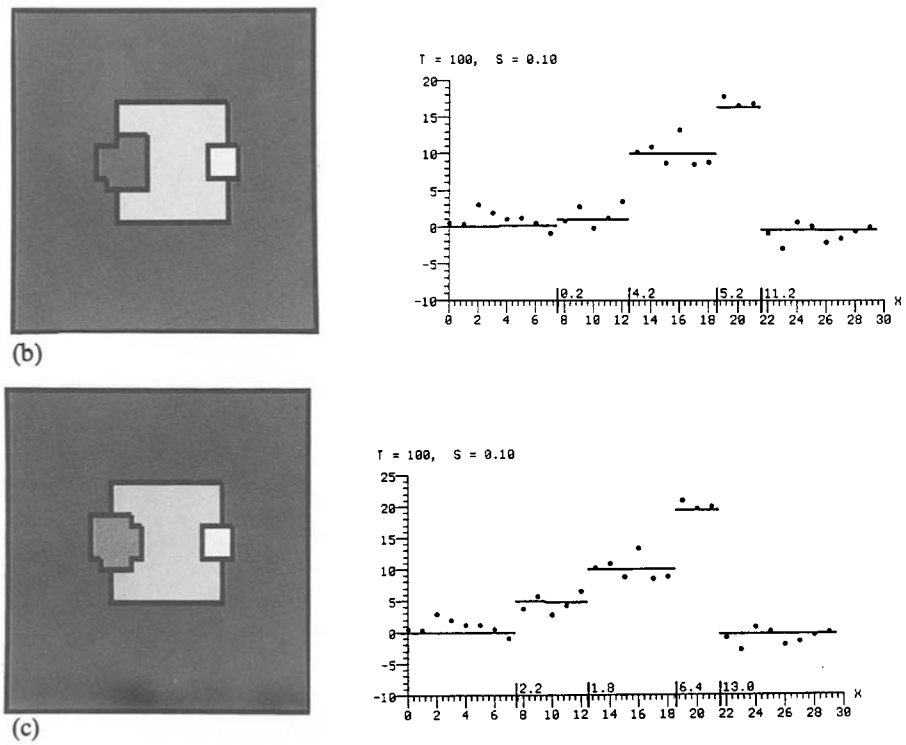


Fig. 2. (b) Reducing the intensity of the left and right squares alters only the less stable boundaries. (c) Tripling the encoding cost of a discontinuity similarly alters only the less-stable boundaries.

measure corresponds to the ease with which a discontinuity can be found. By and large, this seems to correspond to the perceptual ease of seeing the discontinuity for this class of images (modulo global-grouping processes such as subjective contours).

A second and more important point is that s_{ij} is also a good measure of the stability of a discontinuity with respect to changes in: the image, the parameters of the objective function, and the parameters of the continuation method. For example, reducing the intensity of the left and right squares by 3.0 (figure 2b) alters the less stable boundary between the left square and the background, but the more stable boundary of the right square remains unaffected. Tripling parameter b in L (figure 2c) has a similar effect.

This style of local, parallel, and iterative algorithm is ideally suited to massively parallel computer architectures, or even to special-purpose VLSI hardware, because it requires communication between neighboring elements only. Furthermore, this algorithm (and all of the generalizations presented below) can be pipelined for real-time applications. That is, each step in the computation involving a decrease in s can be performed by a separate layer of parallel processors in a chain of such layers. Thus, each layer l computes u^l given u^{l-1} and $\mathbf{z}^{l-1}$ from the preceding layer, and layer 0 computes $\mathbf{u}^0$ as a function of the time-varying input image. Although the total time to process a single image remains the same in such a scheme (namely the time for the image to propagate through the entire chain), a new image can be dealt with in only the time it takes to compute a single step.

6 Generalizing the Piecewise-Constant Case

The piecewise-constant case described above was an important special case in that it allowed us to see several aspects of the approach advocated here, but with a minimum of complications. This special case must be generalized significantly,

however, before it can be applied to a wide variety of real images.

We shall describe three significant generalizations in this section. The first is a generalization of the underlying image to include piecewise-smooth images. The second is a generalization of the noise model to include white noise whose variance is both unknown and spatially varying. The third is a generalization of the sensor model to include a point-spread function.

These generalizations correspond to either the modification of a component of the descriptive language or the addition of a new component. These, in turn, correspond to either the modification of a term or the addition of new terms to the basic cost function. For the sake of simplicity, each component is examined in isolation in a separate section, and the most general case combining all of these components is present in the final section. The results presented following this are all based on the most general case.

6.1 The Piecewise-Smooth Case

First, we generalize the underlying image to include an approximation to piecewise-smooth functions. As in the piecewise-constant case, the underlying image is represented by a grid of square elements, but now each element has a smooth function within its spatial domain. This function is defined in terms of Taylor coefficients, and these coefficients are represented by the vector $\mathbf{u}_i = \{u_{i,k}, k = 1, \ldots, k_{\max}\}$ associated with each element. Thus, the underlying image is of the form

$$u(x,y) = \sum_{k=1}^{k_{\max}} u_{i,k} \frac{(x - x_i)^{\alpha_k}(y - y_i)^{\beta_k}}{\alpha_k!\,\beta_k!} \qquad \forall (x,y) \in \mathcal{R}_i, \quad i \in I$$

where $\alpha_0 = 0$, $\alpha_{k+1} \geqslant \alpha_k$. The underlying image is completely represented by the vector of vectors $\mathbf{u} = \{\mathbf{u}_i, i \in I\}$, and $z_i = u(x_i,y_i) + r_i = u_{i,0} + r_i$. Contrary to the piecewise-constant case, however, removing the noise does not completely specify $\mathbf{u}$ as a function of $\mathbf{z}$; only the subset $u_{i,0} = z_i \; \forall i \in I$ is specified. Thus, even in the noiseless case, we must turn to the simplicity criterion in order to determine the underlying image.

When $k_{\max}$ is infinite, $u(x,y)$ is a general piecewise-smooth function (in fact, it is piecewise-analytic), with the constraint that discontinuities in the function and its derivatives occur only at the boundary between adjacent elements. When $k_{\max}$ is finite, $u(x,y)$ is merely piecewise-polynomial. This, perforce, is the case we examine here.

Although the square grid elements preclude this representation from being precisely invariant to rotation and translation, it is important to choose the exponents (α_k,β_k) in such a way that a rotation or translation of the coordinate system does not require the use of a different set of exponents. To ensure this, we combine the coefficients into groups for which the sum of the exponents is a constant

$$\mathcal{P}_p = \{k : \alpha_k + \beta_k = p\}, \qquad p = 0, \ldots, p_{\max}$$

and consider only underlying images of the form

$$u(x,y) = \sum_{p=0}^{p_{\max}} \sum_{k \in \mathcal{P}_p} u_{i,k} \frac{(x - x_i)^{\alpha_k}(y - y_i)^{\beta_k}}{\alpha_k!\,\beta_k!} \qquad \forall (x,y) \in \mathcal{R}_i, \quad i \in I$$

We call $p_{\max}$ the *order* of $u(x,y)$ within each element, and say that $u(x,y)$ is piecewise order-$p_{\max}$.

There are now two components required to describe $u(x,y)$: the first to describe the nonzero coefficients within each region; the second to describe the boundaries of the regions.

The first component for each region is proportional to the number of nonzero Taylor coefficients in that region. To compute this number at a single element, we must count all coefficients up to the highest-order coefficient that is nonzero. Moreover, we must group them as we did above to rule out accidental alignments with the coordinate system of the grid. Thus, the number of nonzero coefficients in a region, as computed at a single element, is

$$\sum_{p>0} n_p \left[1 - \prod_{k \geqslant \mathcal{P}_p} \delta(u_{i,k})\right]$$

where n_p is the number of elements in $\mathcal{P}_p$, and the notation $k \geqslant \mathcal{P}_p$ is shorthand for $\{k : k \in \mathcal{P}_p, \forall p' \geqslant p\}$.

Note that the order-zero coefficient is not counted, to ensure that this measure is unaffected by the addition of a constant to the real image.

Thus, a local approximation to the number of bits required to encode the coefficients in a region is

$$d \sum_i \sum_{p>0} n_p \left[1 - \prod_{k \geq \mathcal{P}_p} \delta(u_{i,k}) \right] \tag{11}$$

where d is the number of bits required to encode a nonzero coefficient, divided by the average region size.

The second term in the description length is proportional to the number of discontinuities in the function and its derivatives (up to order $p_{\max}$) between adjacent elements. To compute this term, note that the k^{th} derivative of the polynomial within the i^{th} element, evaluated at the boundary between itself and an adjacent neighbor, is

$$\begin{aligned} D_{i,j,k}(\mathbf{u}) &\equiv \frac{\partial u(x,y)}{\partial x^{\alpha_k} \partial y^{\beta_k}} \Bigg|_{(x,y) = \left(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2} \right)} \\ &= \sum_l d_{i,j,k,l} u_{i,l} \end{aligned}$$

where

$$\begin{aligned} d_{i,j,k,l} = &\left\{ \frac{\alpha_l(\alpha_l - 1) \ldots (\alpha_l - \alpha_k + 1)}{\alpha_l!} \left(\frac{x_j - x_i}{2} \right)^{\alpha_l - \alpha_k} \right\} \times \\ &\left\{ \frac{\beta_l(\beta_l - 1) \ldots (\beta_l - \beta_k + 1)}{\beta_l!} \left(\frac{y_j - y_i}{2} \right)^{\beta_l - \beta_k} \right\} \end{aligned}$$

are fixed constants. Since a discontinuity occurs whenever there is a nonzero difference between $D_{i,j,k}(\mathbf{u})$ and $D_{j,i,k}(\mathbf{u})$, an approximation to the second term in the description length is

$$\frac{b}{2} \sum_i \sum_{j \in N_i} \sum_{p>0} n_p \left[1 - \prod_{k < \mathcal{P}_p} \delta(D_{i,j,k}(\mathbf{u}) - D_{j,i,k}(\mathbf{u})) \right] \tag{12}$$

Again we have grouped the coefficients to ensure rotational symmetry. Additionally, we define the product over $k \leqslant \mathcal{P}_p$ so that discontinuities of a lower order impose discontinuities at a higher order.

Combining equations (11) and (12) with the encoding length of the residuals, we arrive at the following approximation for the total encoding length

$$\begin{aligned} L(\mathbf{u}) = &a \sum_i \left(\frac{z_i - u_{i,0}}{\sigma} \right)^2 \\ &+ \frac{b}{2} \sum_i \sum_{j \in N_i} \sum_{p>0} n_p \\ &\times \left[1 - \prod_{k < \mathcal{P}_p} \delta(D_{i,j,k}(\mathbf{u}) - D_{j,i,k}\mathbf{u}) \right] \\ &+ d \sum_i \sum_{p>0} n_p \left[1 - \prod_{k > \mathcal{P}_p} \delta(u_{i,k}) \right] \end{aligned}$$

where all additive constants have been removed.

As before, the embedding for the continuation method is defined by replacing Kronecker deltas with exponentials to obtain

$$\begin{aligned} L(\mathbf{u},s) = &a \sum_i \left(\frac{z_i - u_{i,0}}{\sigma} \right)^2 \\ &+ \frac{b}{2} \sum_i \sum_{j \in N_i} \sum_{p>0} n_p (1 - e_{i,j,p}(\mathbf{u},s)) \\ &+ d \sum_i \sum_{p>0} n_p (1 - e_{i,p}(\mathbf{u},s)) \end{aligned} \tag{14}$$

where

$$\begin{aligned} e_{i,j,p}(\mathbf{u},s) = \exp \Bigg[&- \frac{1}{n_p} \sum_{k < \mathcal{P}_p} \\ &\times \frac{D_{i,j,k}(\mathbf{u}) - D_{i,j,k}(\mathbf{u}))^2}{(s\sigma)^2} \Bigg] \end{aligned}$$

$$e_{i,p}(\mathbf{u},s) = \exp \left[- \frac{1}{n_p} \sum_{k > \mathcal{P}_p} \frac{u_{i,k}^2}{(fs\sigma)^2} \right]$$

Note that products of Kronecker deltas have been replaced by the geometric mean of the exponentials, and that equations (13) and (14) reduce to equations (6) and (7) for piecewise-order-0 surfaces.

Again, any standard descent algorithm may be used for the continuation method, but the following variant of a Gauss-Seidel iterate has proved to be quite efficient for this problem.

First, note that the elements of the gradient of L are

$$\begin{aligned}\frac{\partial L(\mathbf{u},s)}{\partial u_{i,k}} &= \frac{2a}{\sigma^2}(u_{i,0} - z_i)\delta(k) \\ &+ \frac{2b}{(s\sigma)^2}\sum_{j\in N_i}\sum_{p>0} e_{i,j,p}(\mathbf{u},s) \\ &\times \sum_{k'<r_p} [D_{i,j,k'}(\mathbf{u}) - \\ &\qquad D_{j,i,k'}(\mathbf{u})]d_{i,j,k',k} \\ &+ \frac{2d}{(fs\sigma)^2}\sum_{p>0} e_{i,p}(\mathbf{u},s) \\ &\times \sum_{k'\geq r_p} u_{i,k'}\delta(k - k') \qquad (15)\end{aligned}$$

which is of the form

$$\begin{aligned}\frac{\partial L(\mathbf{u},s)}{\partial u_{i,k}} &= \sum_{k'=0}^{k\max} a_{i,i,k,k'}(\mathbf{u},s)u_{i,k'} \\ &+ \left[\left(b_{i,k} + \sum_{i\neq i}\sum_{k'=0}^{k\max} a_{i,j,k,k'}(\mathbf{u},s)u_{j,k'}\right)\right]\end{aligned}$$

or

$$\frac{\partial L(\mathbf{u},s)}{\partial \mathbf{u}_i} = \mathbf{A}_{i,i}(\mathbf{u},s)\mathbf{u}_i + \mathbf{c}_i(\mathbf{u},s)$$

The variant on Gauss-Seidel is to solve for all of the elements of $\mathbf{u}_i$ at time $t+1$ directly by fixing $\mathbf{A}_{i,i}^t \equiv \mathbf{A}_{i,i}(\mathbf{u}^t, s^{t+1})$, $\mathbf{c}_i^t \equiv \mathbf{c}_i(\mathbf{u}^t, s^{t+1})$, and solving the system of equations

$$\mathbf{A}_{i,i}^t\mathbf{u}_i^{t+1} + \mathbf{c}_i^t = \mathbf{0}$$

in parallel for each i, namely

$$\mathbf{u}_i^{t+1} = -(\mathbf{A}_{i,i}^t)^{-1}\mathbf{c}_i^t$$

One advantage of using an explicit representation of the Taylor coefficients can now be seen: the minimization procedure requires information only from the immediate neighbors of an element, rather than information from elements within a given radius, as required for implicit finite-element representations such as in [26]. This is especially advantageous for massively parallel architectures in which the communication cost between nonadjacent units is high. A second advantage is that the value and derivatives of the underlying image can be evaluated at any given point by using only linear combinations of the coefficients, which is computationally inexpensive.

6.2 Images with Known Spatially Varying Noise

The discussion so far has assumed that the variance of the noise is both constant and known a priori. We can deal with known variance that is different from point to point simply by changing σ to σ_i in the first summation of the encoding length functions (equations (6) and (13)).

For example, equation (6) becomes

$$\begin{aligned}L(\mathbf{u}) = a\sum_{i\in I}\left(\frac{u_i - z_i}{\sigma_i}\right)^2 \\ + b\sum_{i\in I}\sum_{j\in N_i}(1 - \delta(u_i - u_j)) \qquad (17)\end{aligned}$$

and the exponentials in the embedding become

$$e_{i,j}(\mathbf{u},s) = \exp\left\{-\frac{(u_i - u_j)^2}{[s(\sigma_i + \sigma_j)/2]^2}\right\} \qquad (18)$$

Note that the average of σ_i and σ_j is used to get symmetric interaction strengths. Analogous changes can be made for the piecewise-smooth case.

6.3 Images with Unknown Spatially Varying Noise

The more interesting case, of course, is when the σ_is are unknown. Then, according to the minimal encoding-length criterion, we must find those values of the σ_is that minimize the overall encoding length, *including the cost of encoding the* σ_is *themselves in some descriptive language.*

One possible model for spatially varying noise is for the variance to be piecewise-constant, with the variance boundaries constrained to coincide with the intensity boundaries. The motivation for this model is that, for real images, the residuals are due not only to sensor noise (which is roughly spatially uniform) but also to small-scale texturing of the objects. Hence, we should expect the variance to differ from region to region. Piece-

wise-smooth-variance models are also possible, but will not be examined here.

For the piecewise-constant-variance model, the cost of encoding a region boundary will now be slightly higher (since the cost of the boundary subsumes the cost of encoding the parameters within the region, which must now include the cost of encoding the variance), and we need to include a term that ensures that $\sigma_i = \sigma_j$ for all adjacent elements not crossing a region boundary. Again, using the piecewise-constant-intensity model for simplicity, and reinserting the term involving $\log \sigma$ that was removed for convenience when σ was fixed (see the definition of a and c, equation (5)), the encoding-length function becomes

$$\begin{aligned} L(\mathbf{u},\sigma) = & \frac{1}{\log 2} \sum_i \left(\frac{u_i - z_i}{\sigma_i} \right)^2 \\ & + \frac{b}{2} \sum_i \sum_{j \in N_i} (1 - \delta(u_i - u_j)) \\ & + \frac{1}{\log 2} \sum_i \log \sigma_i \\ & + \frac{g}{2} \sum_i \sum_{j \in N_i} \delta(u_i - u_j) \\ & \times (1 - \delta(\sigma_i - \sigma_j)) \end{aligned} \tag{19}$$

where b is now slightly larger than before and $g \gg b$.

We could find the global minimum of this function by defining an embedding as before, but the derivative with respect to σ cannot be effectively linearized, making the descent algorithm computationally expensive. Instead, we use the following line of reasoning. Observe that at a local minimum $(\mathbf{u}^*, \sigma^*)$ of equation (19), and for a sufficiently large g, $\sigma_i^* = \sigma_j^*$ whenever $u_i^* = u_j^*$. That is, σ^* is constant within the contiguous regions that $\mathbf{u}^*$ is constant, as we demanded, and the last summation is identically zero. To make this explicit, let R_r denote the set of indexes of the elements within the r^{th} region, let u_r and σ_r denote the constant values of $\mathbf{u}$ and σ within this region, and let $R = \{R_r\}$. Thus, equation (19) can be rewritten at a local minimum as

$$\begin{aligned} & L(R) \\ & = \frac{1}{\log 2} \sum_r \sum_{i \in R_r} \left[\frac{1}{2} \left(\frac{z_i - u_r}{\sigma_r} \right)^2 + \log \sigma_r \right] \\ & + \sum_r (\text{boundary length of } R_r) \end{aligned} \tag{20}$$

To compute the minimal values $(\mathbf{u}^*, \sigma^*)$ for a given set R, note that since the boundaries are fixed by definition, equation (20) is minimal when each term of the first summation is itself minimal. Thus, since each term is now differentiable, we can determine $(\mathbf{u}^*, \sigma^*)$ by differentiating and setting to zero, which yields the unique solution

$$u_r^* = \frac{1}{n_r} \sum_{i \in R_r} z_i$$

$$(\sigma_r^*)^2 = \frac{1}{n_r} \sum_{i \in R_r} (u_r^* - z_i)^2$$

where n_r is the number of elements in R_r. In other words, as we might have expected, the minimal-length encoding occurs when the intensity estimate within each region equals the region average, and the variance estimate equals the region variance; the unknown being what the regions are.

Thus, if we knew the region variance, we could minimize equation (19) by substituting $\sigma_i = \sigma_r^*$ for $i \in R_r$ and defining an embedding exactly as before. Of course, this is not directly possible without already knowing the solution to the problem, but we can come close by noting that the region variance is approximately equal to the average of local estimates of the variance within the region:[3]

$$(\sigma_r^*)^2 = \frac{1}{n_r} \sum_{i \in R_r} (u_r^* - z_i)^2 \approx \frac{1}{n_r} \sum_{i \in R_r} \hat{\sigma}_i^2$$

where

$$\begin{aligned} \hat{\sigma}_i^2 &= \frac{\sum_{j \in N_i \cap R_r} (u_r^* - z_j)^2}{\sum_{j \in N_i \cap R_r} 1} \\ &= \frac{\sum_{j \in N_i} \delta(u_i^* - u_j^*)(u_r^* - z_j)^2}{\sum_{j \in N_i} \delta(u_i^* - u_j^*)} \end{aligned}$$

[3]The inequality occurs only because of boundary conditions. Thus, the approximation is best for large regions, where the effects of boundary conditions are minimal.

Thus, by defining an embedding in which the log σ_i term is replaced by one that converges to the average of σ_i within each region, we should come close to the global minimum.

To achieve this, the embedding is defined recursively by starting with local estimates of the variance based directly on the data, and improving these estimates by basing them on the local minimum $(\mathbf{u}^{*t-1},\sigma^{*t-1})$ of the previous iteration. (The superscript $*t-1$ indicates the last time instant at which the descent algorithm converged to a local minimum.) Let

$$u_i^{*0} \equiv z_i$$

$$\hat{\sigma}_i^0 \equiv \max\left\{q, \sqrt{\frac{\sum_{j\in N_i}(u_i^{*0}-z_j)^2}{\sum_{j\in N_i}1}}\right\}$$

and for $t > 0$, let

$$e_{ij}^{*t-1} = \exp\left\{-\frac{(u_i^{*t-1}-u_j^{*t-1})^2}{[s^t(\sigma_i^{*t-1}+\sigma_j^{*t-1})/2]^2}\right\}$$

$$\hat{\sigma}_i^{*t-1} \equiv \max\left\{q, \sqrt{\frac{\sum_{j\in N_i} e_{i,j}^{*t-1}(u_i^{*t-1}-z_j)^2}{\sum_{j\in N_i} e_{i,j}^{*t-1}}}\right\}$$

Note that the local estimate of the variance is constrained to be greater than q to satisfy equation 3, thereby avoiding quantization problems. The embedding is

$$L(\mathbf{u},\sigma,\mathbf{u}^{*t-1},\sigma^{*t-1},s^t) = a\sum_i\left(\frac{u_i-z_i}{\sigma_i^{*t-1}}\right)^2$$

$$+\frac{b}{2}\sum_{i\in I}\sum_{j\in N_i}\left[1-\exp\left\{-\frac{(u_i-u_j)^2}{[s^t(\sigma_i^{*t-1}+\sigma_j^{*t-1})/2]^2}\right\}\right]$$

$$+a\sum_i\left(\frac{\sigma_i-\hat{\sigma}_i^{t-1}}{\sigma_i^{*t-1}}\right)^2+\frac{g}{2}\sum_{i\in I}\sum_{j\in N_i}e_{i,j}^{*t-1}$$

$$\times\left[1-\exp\left\{-\frac{(\sigma_i-\sigma_j)^2}{[s^t(\sigma_i^{*t-1}+\sigma_j^{*t-1})/2]^2}\right\}\right]$$

The descent algorithm alternately finds a local minimum in $\mathbf{u}$, then in σ, and then decreases s as before.

6.4 Including the Point-Spread Function

Finally, we can include our model for the point-spread function of the image sensor, namely convolution with some known kernel. Since we model the underlying image by using square grid elements with a fixed shape, we can model the point-spread function as a discrete convolution. Thus, if $K_{i,j}, j \in S_i$ are the elements of the convolution kernel, with S_i being the spatial support of the kernel for the i^{th} element, then equation (1) can be rewritten as

$$z_i = \sum_{j\in S_i} K_{i,j}u_j + r_i$$

or

$$r_i = z_i - \sum_{j\in S_i} K_{i,j}u_j \tag{21}$$

This definition of the residuals can then be directly substituted into the cost functions and embeddings defined above.

6.5 The General Case

In the most general case examined here, which is the one used for all of the examples in the following section, all of the modifications and additions are included. Thus, the approximation to the cost of encoding the underlying image and residuals becomes (with additive constants removed)

$$L(\mathbf{u},\boldsymbol{\sigma}) = a\sum_{i\in I}\left(\frac{z_i-\sum_{j\in S_i}K_{i,j}u_{j,0}}{\sigma_i}\right)^2$$

$$+\frac{b}{2}\sum_i\sum_{j\in N_i}\sum_{p>0}n_p\left[1-\prod_{k<\not>_p}\delta(D_{i,j,k}(\mathbf{u})-D_{j,i,k}(\mathbf{u}))\right]$$

$$+d\sum_i\sum_{p>0}n_p\left[1-\prod_{k>\not>_p}\delta(u_{i,k})\right]$$

$$+\frac{1}{\log 2}\sum_i\log\sigma_i+\frac{g}{2}\sum_i\sum_{j\in N_i}\delta(D_{i,j,0}(\mathbf{u})-D_{j,i,0}(\mathbf{u}))(1-\delta(\sigma_i-\sigma_j))$$

and the embedding becomes

$$L(\mathbf{u},\boldsymbol{\sigma},\mathbf{u}^{*t-1},\boldsymbol{\sigma}^{*t-1},s^t)$$

$$= a\sum_i\left(\frac{z_i-\sum_{j\in S_i}K_{i,j}u_{j,0}}{\sigma_i^{*t-1}}\right)^2+\frac{b}{2}\sum_i\sum_{j\in N_i}\sum_{p>0}n_p$$

$$\times\left[1-\exp\left\{-\frac{1}{n_p}\times\sum_{k<\not>_p}\frac{(D_{i,j,k}(\mathbf{u})-D_{j,i,k}(\mathbf{u}))^2}{[s^t(\sigma_i^{*t-1}+\sigma_j^{*t-1})/2]^2}\right\}\right]$$

$$+a\sum_i\left(\frac{\sigma_i-\hat{\sigma}_i^{*t-1}}{\sigma_i^{*t-1}}\right)^2+\frac{g}{2}\sum_{i\in I}\sum_{j\in N_i}e_{i,j,0}^{*t-1}$$

$$\times\left[1-\exp\left\{-\frac{(\sigma_i-\sigma_j)^2}{[s^t(\sigma_i^{*t-1}+\sigma_j^{*t-1})/2]^2}\right\}\right]$$

$$+d\sum_i\sum_{p>0}n_p\times\left\{1-\exp\left[-\frac{1}{n_p}\sum_{k>\not>_p}\frac{u_{i,k}^2}{(fs^t\sigma_i^{*t-1})^2}\right)\right\}$$

where $(\mathbf{u}^{*t-1},\sigma^{*t-1})$ is the local minimum found at $t-1$,

$$e_{i,j,p}^{*t-1}=\exp\left\{-\frac{1}{n_p}\times\sum_{k<\not>_p}\frac{(D_{i,j,k}(\mathbf{u}^{*t-1})-D_{j,i,k}(\mathbf{u}^{*t-1}))^2}{[s^t(\sigma_i^{*t-1}+\sigma_j^{*t-1})/2]^2}\right\}$$

$$\hat{\sigma}_i^{*t-1}=\max\left\{q,\left[\sum_{j\in N_i}e_{i,j,0}^{*t-1}\times(z_j-\sum_{p>0}\sum_{k\not>_p}d_{i,j,k}u_{i,k}^{*t-1})^2\div\sum_{j\in N_i}e_{i,j,0}^{*t-1}\right]^{\frac{1}{2}}\right\}$$

$$d_{i,j,k}=\frac{(x_j-x_i)^{\alpha_k}(y_j-y_i)^{\beta_k}}{\alpha_k!\,\beta_k!}$$

(i.e., the local estimate of the variance is computed by extending the i^{th} element out to the center point of its neighbors), and the recursion is grounded at $t=0$ by defining

$$u_{i,0}^{*0}\equiv z_i$$

$$u_{i,k}^{*0}\equiv 0,\qquad k>0$$

$$\hat{\sigma}_i^{*0}\equiv\max\left\{q,\sqrt{\frac{\sum_{j\in N_i}(z_j-z_i)^2}{\sum_{j\in N_i}1}}\right\}$$

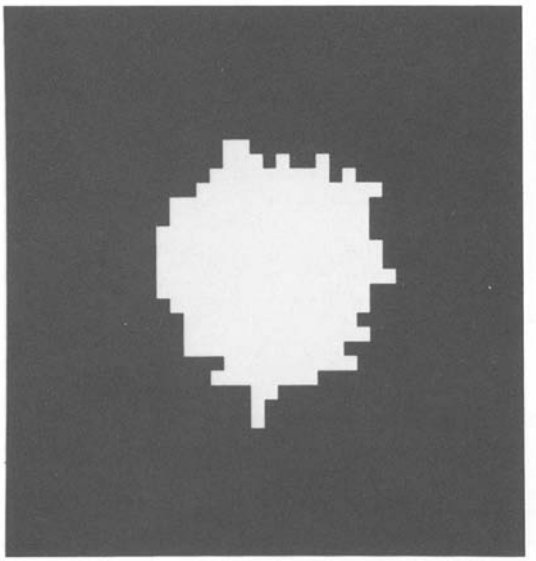
The descent algorithm alternately finds a local minimum in $\mathbf{u}$, then in σ, and then decreases s as before.

7 Results

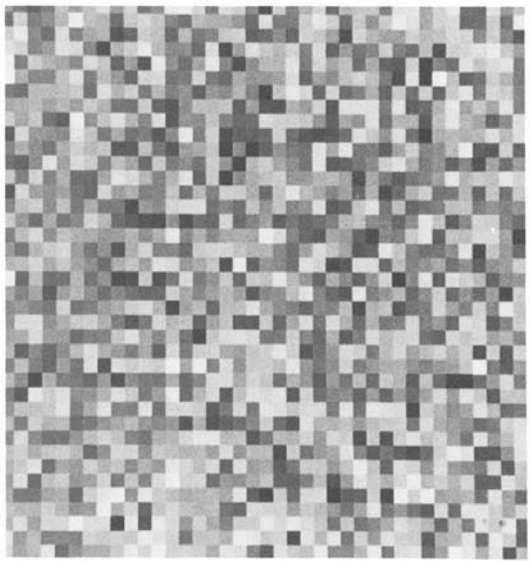
All of the results in this section were obtained by using the most general form of the encoding-length function, in which the underlying image is piecewise smooth, the variance of the noise is unknown and piecewise constant, and the sensor model includes a point-spread function. A key point about these examples is that they were all obtained by using *precisely the same parameters*, with the following exceptions. First, a Gaussian point-spread function with $\sigma=1$ was used for all of the real image, but no point-spread function was used for any of the synthetic images (taking advantage of our a priori knowledge about how these synthetic images were created). Second, for demonstrative purposes only and as noted for each example, several values of $p_{\max}$, the order of the underlying image, were used. The conclusion that emerges from these and many other examples not presented here is that a piecewise-second-order underlying image is appropriate for a large class of real images.

The first example is a series of synthetic images with decreasing signal-to-noise ratio. Each image is the sum of the 39×39 piecewise-constant image of figure 3a and the white-noise image of figure 3b multiplied by a constant. The piecewise-constant image has unit contrast, and the white-noise image is the output of a pseudo-random white-noise generator with zero mean and unit variance. Thus, the inverse of the multiplier just mentioned is simply the local signal-to-noise ratio. We use the same white-noise image in each case to make the comparisons as similar as possible.

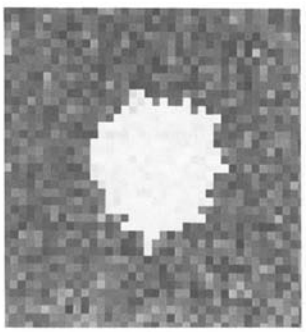
The leftmost images of figures 3c–f are the input images, with signal-to-noise ratios of 8.0, 2.0, 1.0, and 1/2, respectively. For this example, $p_{\max}=0$ and the procedure was stopped at a very low stability value of 1/16. The result of the procedure is illustrated by the image of $\mathbf{u}^t$ with overlaid discontinuities (in the center of each figure), as well as by the graph of the data and elements of the middle row (on the right). All of the

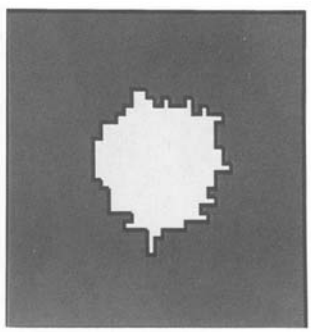


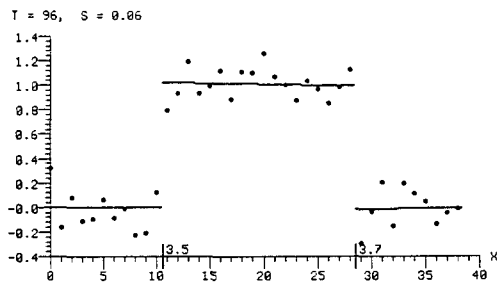
3a



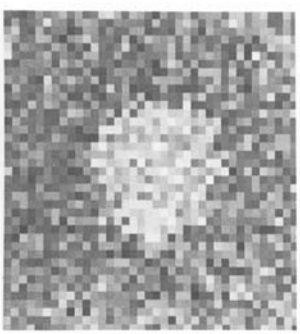
3b

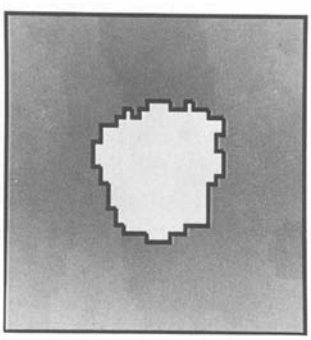


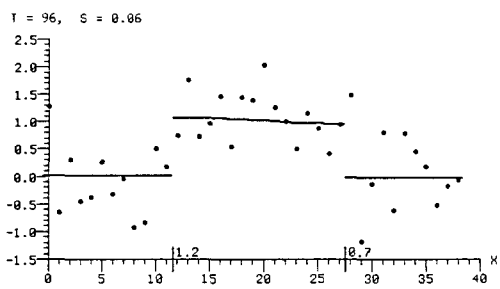




3c



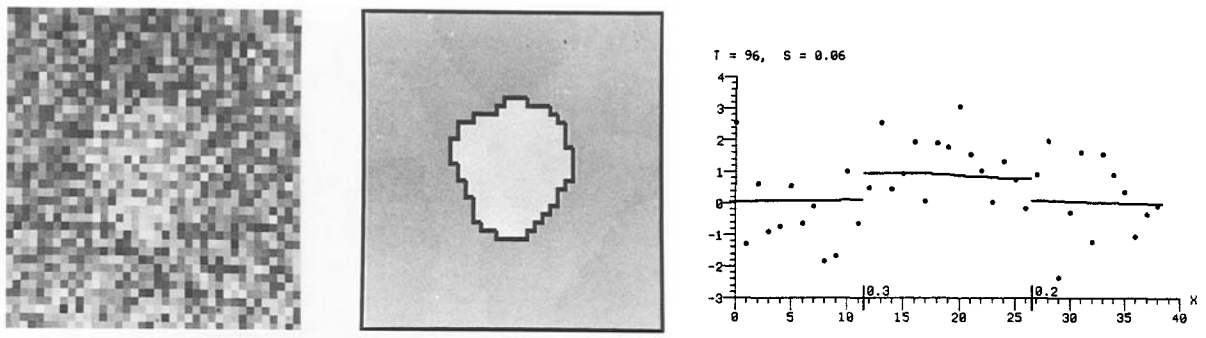




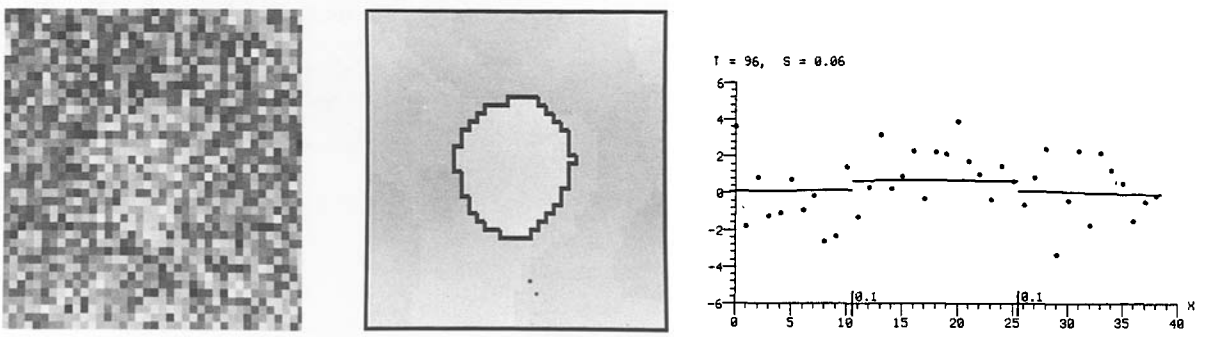
3d

Fig. 3. A series of synthetic images with decreasing signal-to-noise ratio. Figures 3(c)–(f) show the input synthetic images, the resulting underlying images and discontinuities, and a graph of the center row of the procedure's output. (a) The underlying piecewise-constant image, with unit contrast. (b) White noise with $\sigma = 1.0$. (c) The sum of (a) and 1/8 times (b). (d) The sum of (a) and 1/2 times (b). (e) The sum of (a) and (b). (f) The sum of (a) and 2 times (b).

(continued)



3e



3f

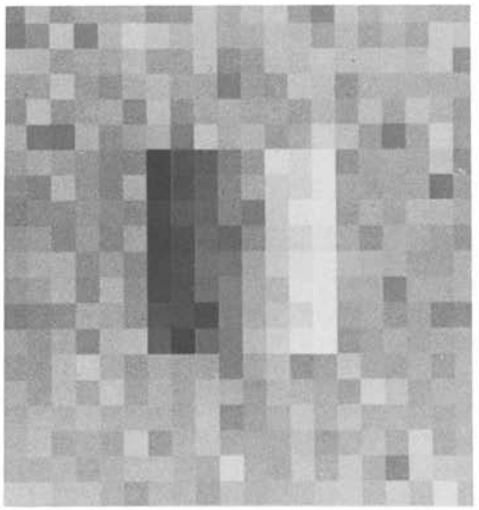
Fig. 3 (continued)

results were obtained with fewer than 100 iterations. For the sake of completeness, the procedure was also applied to the noise image alone. No discontinuities were found.

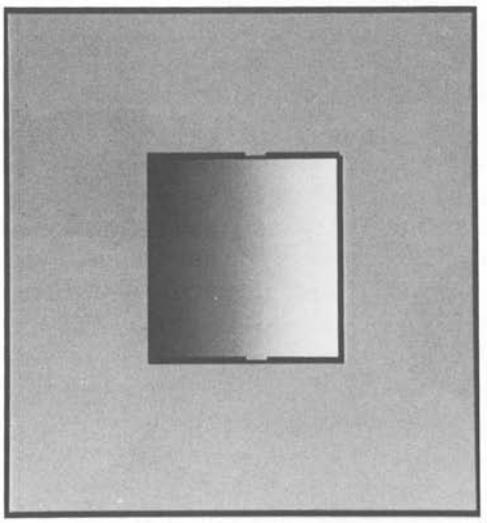
The above series of examples illustrates the behavior of the procedure as a function of the signal-to-noise ratio. In particular, when the signal-to-noise ratio is sufficiently high, every detail of the discontinuities in the underlying image is preserved. As the image is degraded, the precise details are sometimes lost, but the procedure continues to find two distinct regions in the underlying image. (Eventually, of course, the two regions are lost entirely. The precise point at which this occurs depends on the total size and shape of the regions, and, to a lesser extent, on the vagaries of the noise.)

The second example illustrates the power of global optimization compared with purely local, noniterative, operations. Figure 4a is the 20×20 input image, which is the sum of a piecewise-first-order image and zero-mean white noise with unit variance. The outer region of the underlying image has intensity 0.0, the center ramp has a slope of 1.0, and the contrast at either end of the ramp with the outer region is 4.0. Of course, the contrast of the center of the ramp with the background is 0.

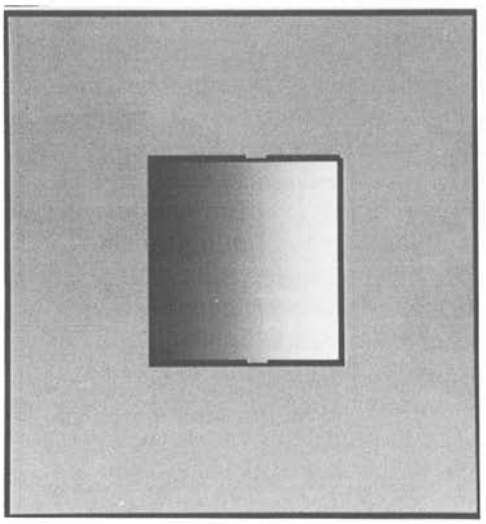
Figures 4b and 4c illustrate the result of the procedure for $p_{\max} = 1$ and 2, respectively, stopping at $s' = 1/4$. First, note that the entire ramp is separated from the background, even in the center where the local signal-to-noise ratio is 0 (the thinner line separating the ramp from the background near the center indicates that the discontinuity is only of order 1, i.e., a discontinuity in the first derivative of the underlying image). This is in contradistinction to the output of the Canny edge detector [5]. For a small spatial scale (figure 4d), the Canny operator leaves a gap (not to mention the introduction of spurious discontinuities due to the assumption that edges are locally piece-



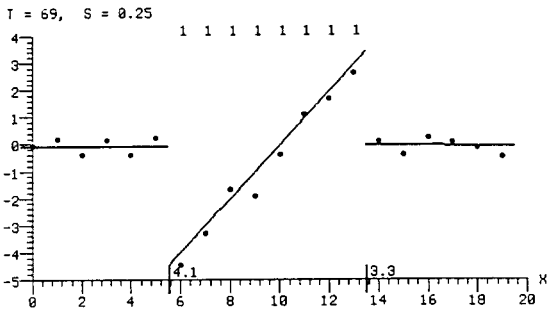
4a



4b



4c



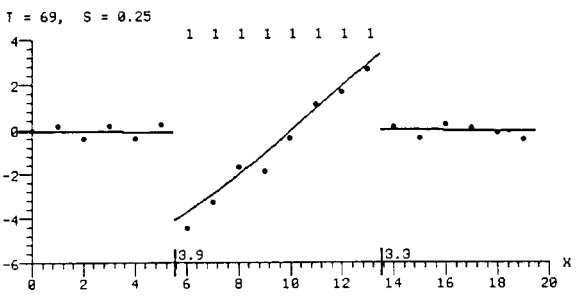
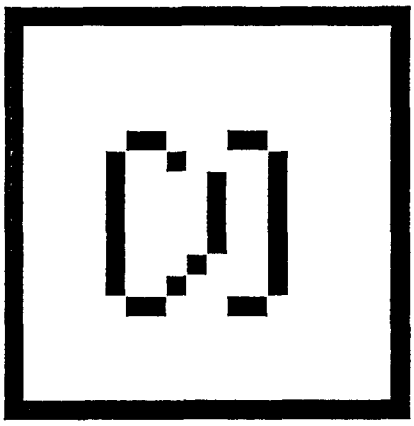
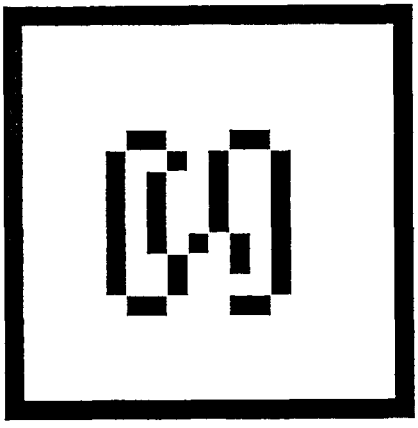


Fig. 4.

(continued)



4d



4e

Fig. 4. An illustration of the power of global optimization. (a) The input synthetic image. (b) The result of the procedure for $p_{max} = 1$. (c) The result of the procedure for $p_{max} = 2$. (d) The output of the Canny operator, mask size = 4. (e) The output of the Canny operator, mask size = 8.

wise-constant), whereas a larger spatial scale (figure 4e) simply makes the artifacts worse. (The operator was unable to find the correct outline for any parameter settings.) Second, note that the elements of the ramp have been determined to be order 1 (as indicated by the number immediately above each element, no number means that the element is order 0), whereas the elements of the outer region have been determined to be order 0. Thus, the procedure has not only located the discontinuities correctly, but has also determined the correct order for each region.

To achieve the above results, the procedure was stopped at fairly low stability values. This required a value of r (the ratio s^{t-1}/s^t) equal to 0.95. Typically, the lower the final stability value, the closer r must be to 1.0. This is because smaller values of s^t require higher accuracies in the calculation of $\mathbf{u}^t$ (since the interaction-strengths, $e^t_{i,j}$, are a function of the difference in adjacent element values relative to s^t), and cumulative errors are a function of how close r is to 1.0.

Stopping the procedure at low stability, as we did above, was reasonable only because it was known that the descriptive language corresponded fairly closely to the manner in which the images were generated. For real images, on the other hand, the elements of the descriptive languages we have just defined are only extreme simplifications of the processes that form an image. There is no direct notion of the three-dimensional nature of objects, their interaction with illuminants, or even any notion of texture. Thus, if we wish to obtain a description whose discontinuities are invariant to the precise nature of the simplifications and approximations, we must stop the procedure at higher stability values. Doing this means that we necessarily lose discontinuities with low signal-to-noise ratios, but this is rarely important. A different strategy might be to stop at a much lower stability value, thereby producing completely closed regions, but then take into account the stability of the individual discontinuities in the further processing of the image. In the examples below, we stopped at $s^t = 1/4$.

Figure 5 illustrates an application of the procedure to an aerial image of a house, with $p_{max} = 1$. Figures 5b and 5c show the resulting underlying image and discontinuities. Figure 5d is an image of the stability measure for these discontinuities, with the darkest lines indicating the most stable discontinuities. Two interesting points emerge from this example. First, the four bushes in the upper-left corner are almost completely delineated, even though the contrast along that part of their boundaries is virtually nil. This is an example of the "zero contrast" situation similar to the previous synthetic ramp image. Second, the majority of discontinuities that form closed regions have high stability measures. This is a fairly strong indication that the piecewise-first-order

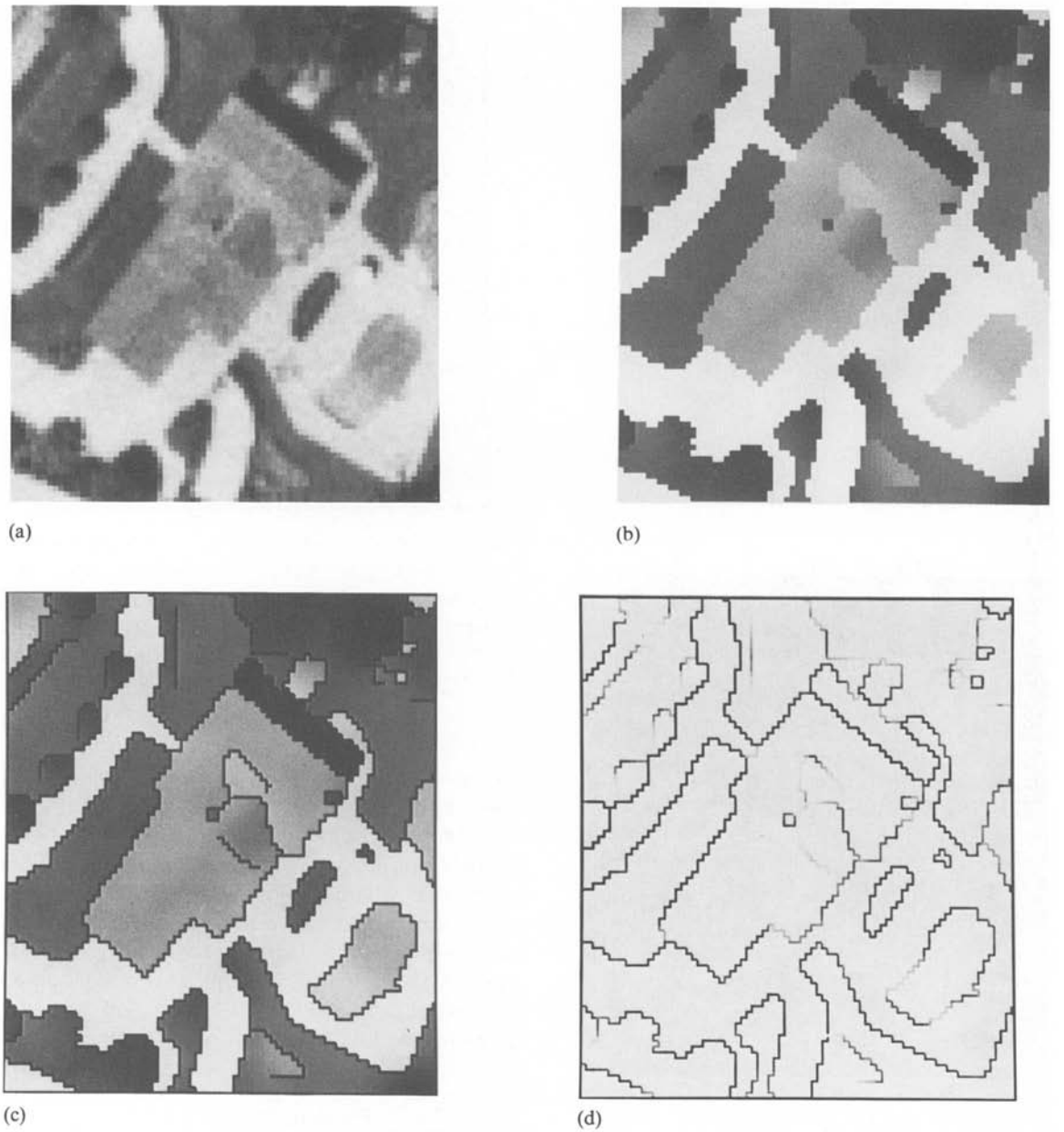
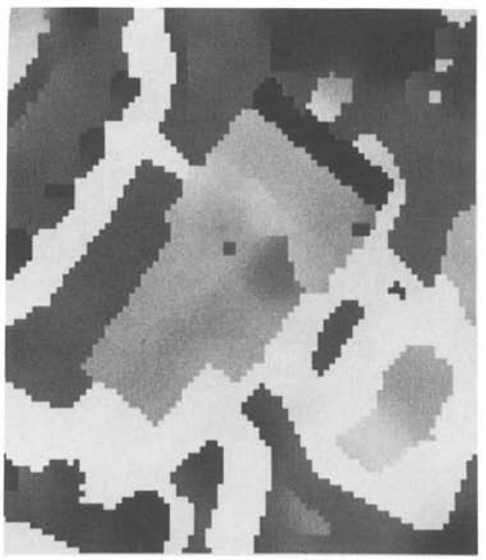
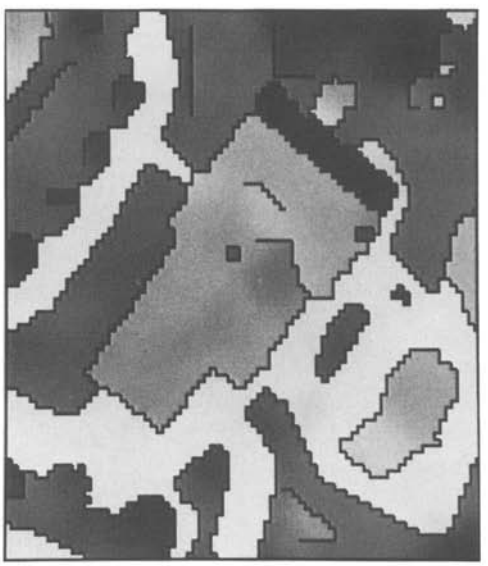


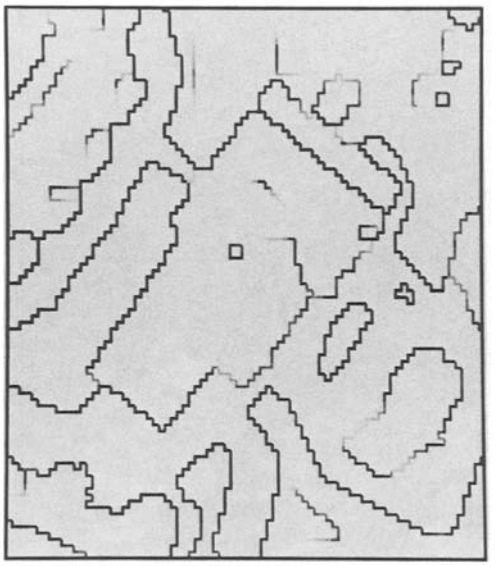
Fig. 5. An application of the procedure to an aerial image of a house, with $p_{max} = 1$. (a) The input image. (b) The resulting underlying image. (c) The underlying image with overlaid discontinuities. (d) The stability measure of the discontinuities; the darkest discontinuities are the most stable.



6a



6b



6c

Fig. 6. Same as the prior figure, but with $p_{max} = 2$. (a) The resulting underlying image. (b) The underlying image with overlaid discontinuities. (c) The stability measure of the discontinuities.

(or higher-order) model is appropriate for this image. To verify this conclusion, observe that the discontinuities obtained using $p_{max} = 2$ (figure 6) are virtually identical; the only exceptions being the few very low stability discontinuities.

Figure 7 illustrates an application of the same model with $p_{max} = 1$ (using precisely the same parameters) to the image of a face. In this example, about half the discontinuities have a fairly low stability measure. This indicates that the language is probably not appropriate for this image. This is especially evident in the cheek and chin areas where a higher-order model is clearly more appropriate. Even so, the discontinuities with high stability measures appear to be good candidates for region boundaries. Figure 8 shows the results for $p_{max} = 2$, in which the artifacts due to using too low an order are entirely absent.

8 Relation to Previous Work

Much work has been done recently on the problem of reconstructing piecewise-smooth surfaces in one or more dimensions, given corrupted samples of the surface [1, 4, 10, 12, 17, 19, 20, 25, 27]. There are several especially difficult aspects to the problem. The first is to determine automatically the appropriate degree of smoothness of the surface as a function of the given data. The second is to determine automatically both the position and

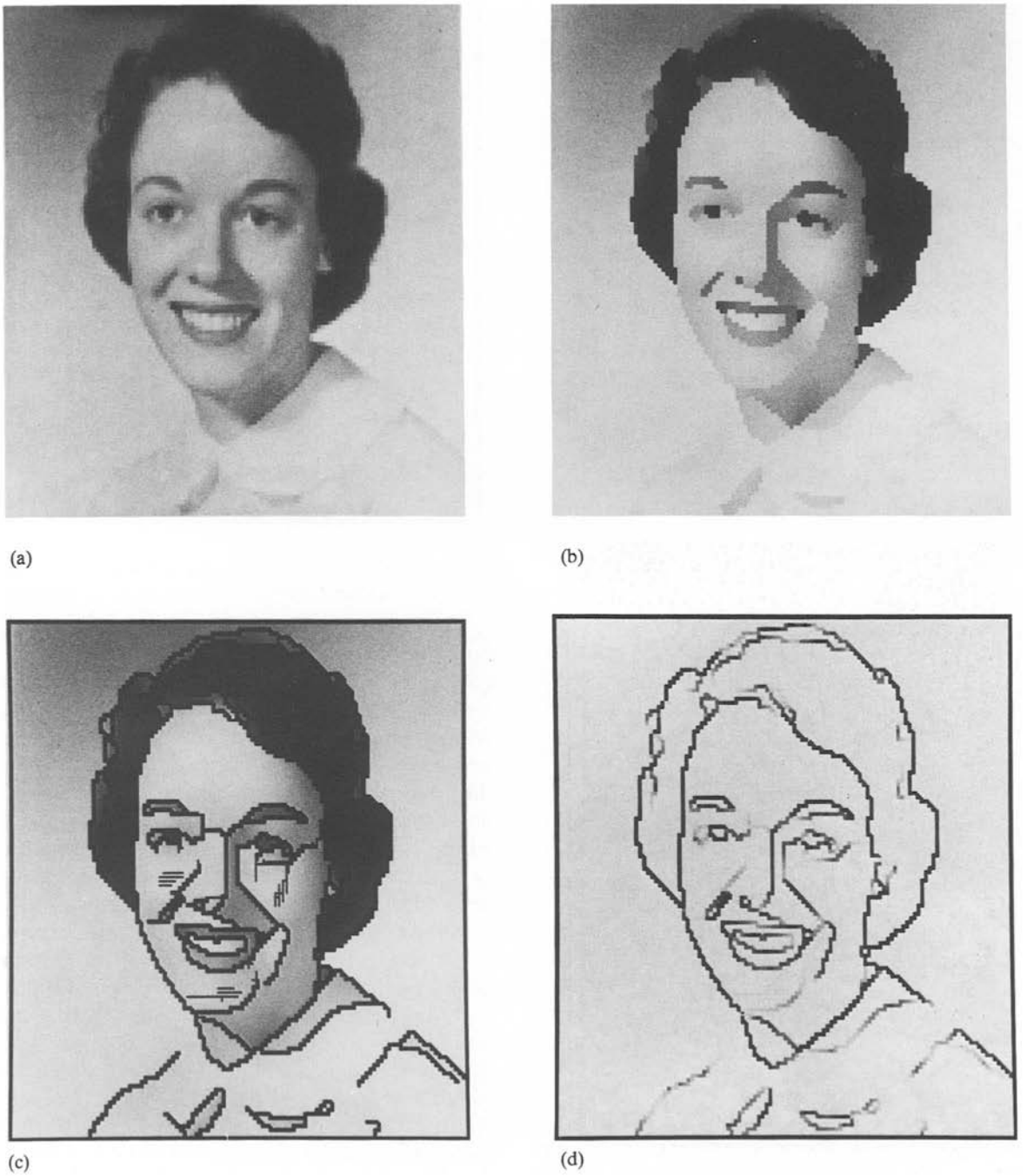
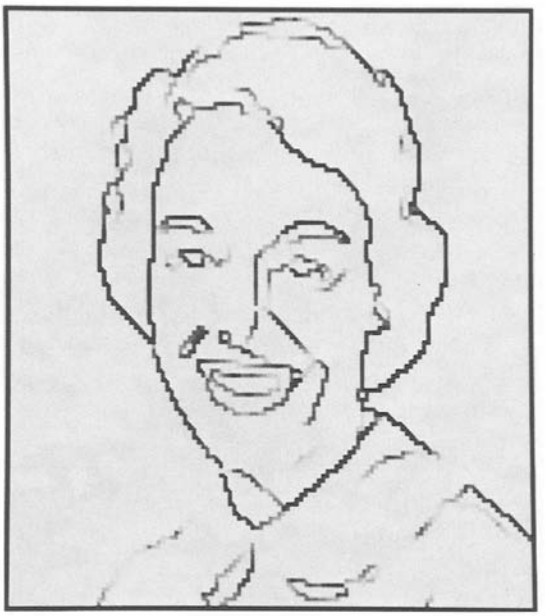


Fig. 7. An application of the procedure to the image of a face, with $p_{max} = 1$. (a) The input image. (b) The resulting underlying image. (c) The underlying image with overlaid discontinuities. (d) The stability measure of the discontinuities.

(a)

(b)



(c)

Fig. 8. Same as the prior figure, but with $p_{max} = 2$. (a) The resulting underlying image and discontinuities. (b) The underlying image with overlaid discontinuities. (c) The stability measure of the discontinuities.

order of the discontinuities. The third is to ascertain when such a description is appropriate for the data. We have resolved these difficulties by (1) posing the problem as an optimization problem in which the objective function is based on the information-theoretic notion of minimum-length descriptions, and (2) defining an algorithm that balances simplicity of description against stability of description by finding the most stable aspects of the description first.

Perhaps the closest in spirit to the work presented here are the excellent book and papers of Blake and Zisserman [4], Marroquin et al. [19], and Mumford and Shah [20]. In these works, the problem is posed as an optimization problem that resembles ours, in kind, but one in which the data are weighted uniformly, in essence independently of the data. In other words, the algorithms do not adapt to the diverse conditions that obtain in different parts of an image, thereby resulting in the heuristic setting of various parameters. Furthermore, no method of determining the appropriate amount of smoothing (i.e., the order of the smoothing term in their cost functionals) is mentioned. The advantage of these simplifications,

however, is that the authors of [4, 20] were able to prove that their algorithms found the global minimum for restricted classes of one-dimensional signals, something that has not yet been possible for our method. No proofs were given for two- dimensional signals. Finally, it is difficult to see how these approaches can be extended to subsequent stages of the vision problem without explicitly bringing to bear some of the notions of descriptive languages, simplicity, and stability (notions that are effectively implicit in the foregoing authors' work).

The works of Besl and Jain [1], Grimson and Pavlidis [10], Langridge [12], Lee and Pavlidis [17], and Terzopoulos [27] view the problem as one of smoothing (or regularization) with embedded discontinuities, but for which the discontinuities are first "detected" in some way from the data, with perhaps some attempt at improving the results iteratively. The heart of the problem—formally including the cost of introducing discontinuities as part of the optimization process—is missing (although Terzopoulos [27] devotes some attention to the problem).

Finally, Saint-Marc and Medioni [25] present a simple adaptive smoothing technique that bears a certain resemblance to our special case of a piecewise-constant underlying image with known variance. While it lacks the true adaptation to spatially varying noise, and depends on a heuristic parameter, it may nevertheless be possible to derive a formal relationship between their approach and ours.

9 Summary

We have presented a new approach to the image-partitioning problem: construct a complete and stable description of an image in terms of a descriptive language that is simplest in the sense of being shortest. We have presented criteria on which to base formal definitions of completeness, stability, and simplicity, and have embodied these criteria within the theory of minimum-length descriptions. This formalism is very general and is likely to be applicable to other stages of the scene-analysis process.

For the specific image-partitioning problem, we described real images as the corruption of ideal (piecewise-polynomial) images by blurring and the addition of spatially varying white noise. We defined a language for describing both the ideal image and the corruptions, and presented an algorithm for finding the simplest description of an image, in terms of this language, for a given measure of stability. This measure has proved *crucial* because we are interested in descriptions that are not only as simple as possible, but that are also as invariant as possible to the severe approximations embodied in any low-level descriptive language. The algorithm not only determines the position of discontinuities in the ideal image, but also determines both the order of the discontinuity and the order of the polynomial within the regions; all of this is done without the need to adjust any parameters. Furthermore, the algorithm is local, parallel, and iterative, making it ideally suited to massively parallel computer architectures.

Applications of this formalism to real images indicate that, even though the descriptive language we have defined is extremely simple (with no models of three-dimensional shape, lighting, or texture, for example), the simplest and most stable descriptions in this language yields excellent image partitions.

Acknowledgements

I wish to thank Edwin Pednault for introducing me, in 1986, to minimal-length encoding and its application to one-dimensional splines; Demetri Terzopoulos for the many discussions we have had concerning optimization theory; Ken Laws for discussions concerning statistics; and Martin Fischler for discussions on the importance of criteria other than simplicity.

References

1. P.J. Besl and R.C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. PAMI* 10(2):167–192, 1988.

2. T.O. Binford, "Inferring surfaces from images," *Artificial Intelligence,* 17(1-3):205-244, 1981.
3. A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction," *IEEE Trans. PAMI* 11(1):2-12.
4. A. Blake and A. Zisserman, *Visual Reconstruction,* MIT Press: Cambridge, MA, 1987.
5. J.F. Canny, "A computational approach to edge detection," *IEEE Trans. PAMI* 8(6):679-698, 1986.
6. G. Dahlquist and A. Björck, *Numerical Methods,* N. Anderson (trans.), Prentice-Hall: Englewood Cliffs, NJ, 1974.
7. P. Fua and A.J. Hanson, "Generic feature extraction using probability-based objective functions," submitted to *Machine Vision and Applications,* 1988.
8. S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. PAMI* 6(6): 721-741, 1984.
9. M.P. Georgeoff and C.S. Wallace, "A general selection criterion for inductive inference," *SRI Tech. Note 372,* SRI International, Menlo Park, CA, 1985.
10. W.E.L. Grimson and T. Pavlidis, "Discontinuity detection for visual surface reconstruction," *Computer Vision, Graphics, and Image Processing* 30:316-330, 1985.
11. R.M. Haralick, "Digital step edges from zero crossings of second directional derivatives," *IEEE Trans. PAMI* 6(1): 58-68, 1984.
12. D.J. Langridge, "Detection of discontinuities in the first derivatives of surfaces," *Computer Vision, Graphics, and Image Processing* 27:291-308, 1984.
13. K.I. Laws, "Textured Image Segmentation," Ph. D. Thesis, Report USCIPI 940, Image Processing Institute, U. Southern California, Los Angelos, CA, 1980.
14. Y.G. Leclerc, "Capturing the local structure of image discontinuities in two dimensions," *Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition,* San Francisco, CA, pp. 34-38, June, 1985.
15. Y.G. Leclerc, "The Local Structure of Image Intensity Discontinuities," Ph. D. dissertation, McGill University, Montréal, Québec, Canada, in preparation.
16. Y.G. Leclerc and S.W. Zucker, "The local structure of image discontinuities in one dimension," *IEEE Trans. PAMI* 9(3):341-355, 1987.
17. D. Lee and T. Pavlidis, "One-dimensional regularization with discontinuities," *Proc. 1st Intern. Conf. Computer Vision,* London, pp. 572-577, 1987.
18. D.G. Luenberger, "Linear and Nonlinear Programming, (2nd ed.), Addison-Wesley: Menlo Park, CA, 1984.
19. J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Am. Stat. Assoc.* 82(397):76-89, 1987.
20. D. Mumford and J. Shah, "Boundary detection by minimizing functionals, I," *Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition,* San Francisco, CA, pp. 22-26, 1985.
21. R. Ohlander, K. Price, and D.R. Reddy, "Picture segmentation using a recursive region splitting method," *Computer Graphics and Image Processing* 8(3):313-333, 1978.
22. T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature,* vol. 317, 1985.
23. J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Annals of Statistics* 11(2):416-431, 1983.
24. J. Rissanen, "Minimum-description-length principle." In *Encyclopedia of Statistical Sciences,* vol. 5, Wiley: New York pp. 523-527, 1987.
25. P. Saint-Marc and G. Medioni, "Adaptive smoothing for feature extraction," *Proc. DARPA Image Understanding Workshop,* Cambridge, MA, pp. 1100-1113, 1988.
27. D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. PAMI* 8(4): 413-424, 1986.
28. A.W. Witkin, "Scale space filtering," *Proc. 8th Intern. Joint Conf. Artif. Intell.,* Karlsruhe, West Germany, pp. 1019-1021, 1983.
29. A.W. Witkin, D. Terzopoulos, and M. Kass, "Signal matching through scale space," *Intern. J. Computer Vision* 1(2):133-144, 1987.