A Subpixel Image Restoration Algorithm

John Gavin; Christopher Jennison

Journal of Computational and Graphical Statistics, Vol. 6, No. 2. (Jun., 1997), pp. 182-201.

Stable URL:
http://links.jstor.org/sici?sici=1061-8600%28199706%296%3A2%3C182%3AASIRA%3E2.0.CO%3B2-U

Journal of Computational and Graphical Statistics is currently published by American Statistical Association.

A Subpixel Image Restoration Algorithm

John GAVIN and Christopher JENNISON

In statistical image reconstruction, data are often recorded on a regular grid of squares, known as pixels, and the reconstructed image is defined on the same pixel grid. Thus, the reconstruction of a continuous planar image is piecewise constant on pixels, and boundaries in the image consist of horizontal and vertical edges lying between pixels. This approximation to the true boundary can result in a loss of information that may be quite noticeable for small objects, only a few pixels in size. Increasing the resolution of the sensor may not be a practical alternative. If some prior assumptions are made about the true image, however, reconstruction to a greater accuracy than that of the recording sensor's pixel grid is possible. We adopt a Bayesian approach, incorporating prior information about the true image in a stochastic model that attaches higher probability to images with shorter total edge length. In reconstructions, pixels may be of a single color or split between two colors. The model is illustrated using both real and simulated data.

Key Words: Bayesian statistical image reconstruction; Confocal microscopy; Deconvolution; Edge detection; Gibbs sampler; Markov chain Monte Carlo; Metropolis algorithm; Subpixel resolution.

1. INTRODUCTION

Image analysis is the science of extracting information from pictures. The human eye is very good at extracting qualitative information but more consistent, objective methods are needed. Also, automatic processing of images is becoming more essential as more and more data are captured via images. Statistical methods can be used to extract quantitative information automatically. For example, we might want to count the number of objects in an image, estimate object areas, measure distances between objects, describe the shape of objects, or find their boundaries.

In image analysis, data are collected by a perspective projection of objects onto a planar grid of rectangular or square pixels on a sensor array. The location and orientation of the pixel grid can have a noticeable effect on the shape of objects at the resolution limits of the recording sensor.

John Gavin is Research Student, and Christopher Jennison is Professor of Statistics, School of Mathematical Sciences, University of Bath, Bath BA2 7AY, United Kingdom; e-mail: C.Jennison@maths.bath.ac.uk.

Journal of Computational and Graphical Statistics, Volume 6, Number 2, Pages 182–201

1.1 EXAMPLE 1: BENEFITS OF CONTINUOUS SUBPIXEL RECONSTRUCTIONS

As an example, a continuous binary image is shown in Figure 1(a). In Figure 1(b), the corresponding observed data, called the record, is a 16×16 gray-scale image. The level of gray in each record element, also called a pixel, of this discretized image is recorded as a real number. This number represents the weighted average of the two colors in the original image. In our example, the foreground color (black) has a value of one and the background color (white) has a value of zero. The weights are the areas of the two colors within that pixel. There is a small amount of noise, $\sigma^2 = .01$, and no blurring in Figure 1(b). The loss of information between Figure 1(a) and Figure 1(b) is due mainly to the discrete nature of the sensor. Figure 1(c) shows a full pixel reconstruction where each pixel is either black or white. Increasing the resolution of the sensor may be undesirable or not technically possible in some applications. For example, an increase in the intensity of illumination may damage a biological sample. However, by expressing fairly

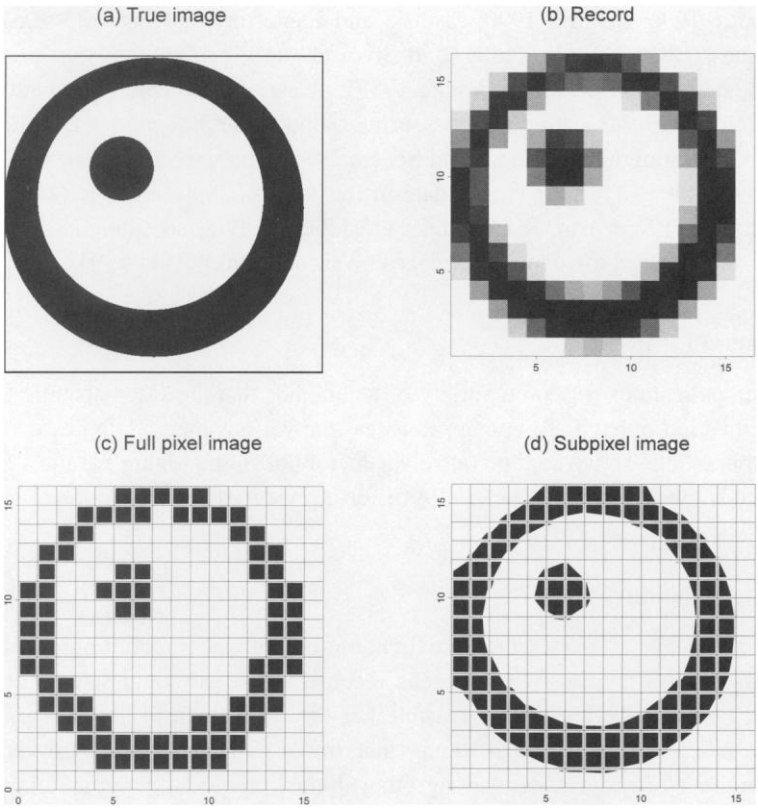

Figure 1. Benefits of Continuous Subpixel Reconstructions. A continuous binary image, the observed record shown as a 16×16 gray-scale pixel image, a full pixel reconstruction, and a subpixel reconstruction of the continuous image are shown in plots (a)–(d) respectively. The true image shows two objects, one lying completely inside the other. Both objects are circular rather than straight-lined and are near the resolution of the pixel grid. Plot (c) represents a typical full pixel reconstruction. In contrast, plot (d) shows the benefits of a subpixel reconstruction. Edges are more clearly defined and this facilitates more accurate edge-length and surface-area estimation.

mild assumptions about the underlying image in a mathematical model and combining this with the observed data, it is possible to reconstruct the image to a higher accuracy than the scanning resolution of the sensor. This process of "subpixel resolution" can be used to provide improved estimates of edge length and area of objects and to detect the presence and shape of objects that are only a few pixels in size. Figure 1(d) shows the reconstruction obtained using the algorithm we describe in this article. Apart from a few errors at the edges of the image, the object boundaries are recovered to a higher degree of accuracy than the full pixel reconstruction.

For each reconstruction in Figures 1(c) and (d), a pixel grid has been superimposed and the region of black in each pixel has been drawn to lie inside the boundary of each pixel. This is purely a graphical device to help illustrate how the image is constructed.

1.2 APPLICATIONS OF SUBPIXEL METHODS

Applications of subpixel resolution include: measuring the size and shape of microscopic fibres (Hitchcock and Glasbey 1994); classification of subpixel vegetation cover (Foody 1994; Jasinski 1990; Jasinski and Eagleson 1990); remote sensing of active volcanos (Bhattacharya, Reddy, and Srivastav 1993); high-precision measurement of machine component positions (Young 1987); measuring the width and path of a laser beam (Szirányi 1992); locating and measuring blood vessel boundaries and diameters in cine coronary angiography (Sandor and Spears 1985); subpixel alignment in lithography (Gatherer and Meng 1992); pictorial data in the form of line drawings (Koplowitz and Sundar Raj 1987; Sriraman, Koplowitz, and Mohan 1989); and subpixel deconvolution of optical and infrared astronomical images (Weir and Djorgovski 1991).

1.3 SUBPIXEL METHODS

These applications rely on a variety of techniques that are often specific to the particular application in hand. To attempt to assess the various methodologies is beyond the scope of this article, so we confine ourselves merely to summarizing the most commonly used methods. See West and Clarke (1990) for a review of several subpixel methods.

1.3.1 Interpolation

View the image as a discrete approximation to an unknown, continuous object sampled at points on a lattice. To enlarge the reconstruction we could sample at a greater frequency, but this is often neither feasible nor desirable. Instead, the true image is estimated at interpixel points by assuming that smooth changes occur between the four nearest neighbor record values. Among other things, this requires an assumption about the order of the polynomial used in the interpolation. This method is usually used in conjunction with other methods, such as filtering. For example, Lorensen and Cline (1987) proposed a three-dimensional (3D) algorithm that effectively thresholds the data, which they call "marching cubes." It efficiently draws a surface through the 3D data by interpolating between neighboring voxel values.

1.3.2 Filters

Filters enhance or emphasize certain features of images by applying transformations based on sets of neighboring pixels. The objective is to remove noise or enhance edges. These methods provide fast subpixel algorithms because the filter is usually chosen to be simple and spatially invariant, for convenience. Iteration is not usually required nor is knowledge about the shape of the edges in the image. However, subpixel accuracy cannot be achieved without using some form of interpolation. An assumption about the filter width is required. Some allowance is also needed for the effect on accuracy, if edges are in close proximity but not connected (Boult and Wolberg 1993; Huertas and Medioni 1986; Oakley and Shann 1991; Sandor and Spears 1985).

1.3.3 Transforms

If the low-dimensional parametric description of the object is known, then a model-fitting procedure can be used to find the parameter values for that object that best fits the data. The method works well even in the presence of noise. For example, Hitchcock and Glasbey (1994) parameterize small objects, peanut kernels, using Fourier descriptors for the boundary and then optimize the parameters. They also describe an image of fibers using a network of splines. Gatherer and Meng (1992) used a discrete Fourier transform for subpixel alignment in lithography.

1.3.4 Template Matching

An ideal edge is fitted to the observed record values, lying in some window, by matching statistics calculated from the proposed edge and the record values or by maximizing some joint statistic such as correlation. So the ideal edge is rotated, translated, or rescaled to best fit the observed data where, for example, the ideal edge might be a step or ramp edge. Because the shape of the object is assumed known, correlation can be used to determine the translation, rotation, and scale parameters. Correlation is used because the cross-correlation function is smooth, enabling interpolation to find the correlation peak (West and Clarke 1990). The ideal moments are calculated by integrating the ideal edge over a chosen window, subject to the sample moments being equal to the moments of the ideal edge.

This approach offers a closed-form solution, works well even when noise is present, and requires no interpolation or iteration. However, it requires assumptions about the shape of the edge (e.g., the edge forms a straight line or is circular over the edge-fitting window) and the size of the window over which the edge is estimated. Tabatabai and Mitchell (1984) discussed a method of moments algorithm and Lyvers, Mitchell, Akey, and Reeves (1989) discussed an extension to this algorithm that includes spatial information in the moment equations.

1.3.5 Stochastic Models

Jennison and Jubb (1988) and Jubb and Jennison (1991) introduced a stochastic model for the edge process. They first find the boundary around an object and then optimize the position of a sequence of linked line segments around the boundary to provide a subpixel estimate of the edge. The model favors patterns with smaller total edge length. Our model is similar to that of Jubb and Jennison (1991), but the algorithm used to implement this model is much improved.

These methods have much to offer, but no single method is universally satisfactory nor is this ever likely to be the case. The method proposed in this article provides an addition to an analyst's tool box. It has its own limitations, which must be recognized. However, it offers mathematically and computationally attractive features not found in other methods.

1.4 OUR METHOD

Our proposed method of subpixel reconstruction uses a probabilistic model for the original image that is combined with the recorded signal by Bayes's theorem to produce a posterior image distribution. Markov chain Monte Carlo (MCMC) algorithms are then used to search for the mode of the posterior distribution, which we take as our point estimate of the true but unknown image. This Bayesian formulation offers a unified approach to image analysis covering low-level analysis, such as the removal of noise and blur, to higher level work such as tomographic reconstruction, segmentation, classification and texture modeling and, at the highest level, object recognition (Besag, Green, Higdon, and Mengersen 1995).

In our model for the true image, we assume that the boundaries between regions of color are smooth and that regions of any one color are fairly large. Thus, the image has locally at most two colors.

We break down the formidable problem of finding the mode of the posterior image distribution into stages and use the solution at the end of each stage as the starting point for the next. As the algorithm progresses, it focuses on finer details of the image but it is constrained from straying too far from its starting point as it searches larger image spaces in the later stages.

In Section 2 of this article, we describe the Bayesian approach to image reconstruction and introduce our model for the true image. We define an algorithm for implementing this model in Section 3, present an application from microscopy in Section 4, and draw our conclusions in Section 5.

2. MODELS AND INFERENCE

2.1 BAYESIAN IMAGE RECONSTRUCTION

The methodology used in this article is based on the Bayesian approach to two-dimensional image reconstruction proposed by Grenander (1983), Geman and Geman (1984), and Besag (1986). A Markov random field probability model is used to introduce prior knowledge about the image. This is combined with the observed data to produce a

posterior distribution for the image X given the observed data Y. Let the prior density for the scene X be denoted by $f_X(x)$, where $x \in \Omega$; let the probability density of the observed image Y given $X = x$ be $f_{Y|X}(y, x)$; and let $f_Y(y)$ denote the marginal density of Y. Then, in a Bayesian analysis, inference about X is based on the posterior density of X given $Y = y$

$$f_{X|Y}(x, y) = f_{Y|X}(y, x) f_X(x) / f_Y(y) \propto f_{Y|X}(y, x) f_X(x), \qquad x \in \Omega. \tag{2.1}$$

A commonly used estimator of the true image is the mode over x of the posterior density $f_{X|Y}(x, y)$. This maximum a posteriori (MAP) estimator is a convenient choice for our subpixel problem. See Besag et al. (1995) for other possible estimators.

2.2 THE TRUE IMAGE

In this article, we consider images composed of two colors—c^1 and c^2. In principle, we would like to allow any division of the image between these colors in continuous space; however, our methods rely on the assumption that regions of a particular color are fairly large and the boundaries between areas of opposite color are smooth. For practical reasons, we allow at most a single straight line edge to divide any pixel between the two colors in our reconstructions and we incorporate this property into our prior model for the true, continuous scene.

2.3 THE PRIOR IMAGE DISTRIBUTION

Our prior distribution for the true scene X is on a class of binary images, composed of colors c^1 and c^2, in which boundaries are continuous and piecewise linear. An image is made up of a rectangular array of pixels that are either of a single color or divided into two regions of different colors by a single straight line edge; the edge meets with adjacent edges in neighboring pixels. Edges are not allowed to pass through a corner of a pixel, to avoid anomalies, but edges can be arbitrarily close to a corner. The color of a pixel is the same as that of all of its neighbors, with agreement on each side of an edge when one is present. We define Ω to be the set of images satisfying these conditions (see Fig. 3(d) for an example of a scene $x \in \Omega$). We index the pixels of an image by $i = 1, 2, \ldots, n$ and denote the coloring of pixel i by x_i. We shall also use the notation $x(z) \in \{c^1, c^2\}$ to denote the color of image x at the point $z \in \Re^2$.

Let $L(x)$ denote the total edge length in the reconstruction x, which is the sum of the lengths of straight line segments in those pixels that contain an edge. In our prior model we set

$$f_X(x) = k \exp\{-\beta L(x)\} \qquad x \in \Omega, \tag{2.2}$$

as the density for image x. Here β is a smoothing parameter that controls the distribution, lower values of β leading to a greater probability of images with long edges. The constant k ensures that f_X integrates to one but its value need not be known to implement our method.

Technically, the density $f_X(x)$ must be defined with respect to a measure on Ω. We define two images in the space Ω to be "similar" if one can be obtained from the

other by repositioning the points of intersection of straight line edge segments on the pixel edges that they cross. Thus, Ω can be partitioned into a finite collection of similar images. Let $\mu(x)$ be the measure obtained by ascribing equal probability to all elements of this partition and then supposing that the end-points of each edge segment are placed uniformly within the appropriate pixel edge. The density in Equation (2.2) is defined with respect to this measure μ.

2.4 Observed Data and Likelihood

The observed data record $Y = \{Y_i \in \Re : i = 1, 2, \ldots, n\}$ is recorded on a regular lattice of points $\{z_i \in \Re^2 : i = 1, 2, \ldots, n\}$. The sensor's output for each record element Y_i represents the average intensity within that pixel but it can be subject to degradation because of blurring and noise in the recording sensor. So at the ith pixel, a single record element Y_i is observed that reflects the mixture of colors present in some neighborhood of pixel i. The exact form of the blurring and noise is problem specific and in general a detailed analysis is required to specify these parameters. In the examples in this article, we assume Gaussian blurring and additive Gaussian noise, so

$$Y_i = \int x(z)g(z_i, z)dz + e_i, \tag{2.3}$$

where $x(z) \in \{c^1, c^2\}$ is the value of the true image at the point $z \in \Re^2$, $g(z_i, z)$ is a Gaussian blurring function that decreases in value as the distance between z and the point z_i increases, and e_i is independent, additive, Gaussian sensor noise. Our method does not rely specifically on these assumptions and we would expect it to work for other cases of the general degradation process described by Geman and Geman (1984, sec. II, eq. (2.1)).

In practice, we use a discrete approximation to the blurring kernel in which $g(z_i, z)$ is assumed to be piecewise constant in z across pixels. In this case, $\int x(z)g(z_i, z)dz$ depends on X only through the proportion of each color in each pixel. We denote the average value of $x(z)$ over points z in pixel j by $h_j(x) = c^1 p_j^1 + c^2 p_j^2$, where p_j^1 and p_j^2 are the proportions of pixel j covered by colors c^1 and c^2, respectively. Letting $K_{ij} = g(z_i, z_j)$, Equation (2.3) becomes $Y_i = \sum_{j:j\in\delta_i} K_{ij}h_j(x) + e_i$, where the set δ_i contains indexes j of pixels sufficiently close to i that $K_{ij} \neq 0$.

Note that when blurring is present, pixels around the border of the image are only observed indirectly through their contributions to the records of neighboring pixels. In this case, there are fewer elements in the record Y than there are pixels in the image X. We shall assume the blurring coefficients K_{ij} and noise variance σ^2 are known or can be estimated from the data. The probability density of the signal y given the true image x is then

$$f_{Y|X}(y, x) \propto \exp\{-(2\sigma^2)^{-1}||y - Kh(x)||^2\}. \tag{2.4}$$

2.5 THE POSTERIOR DISTRIBUTION

The posterior distribution is formed by substituting Equations (2.2) and (2.4) into Equation (2.1) to get

$$f_{X|Y}(x,y) \quad \propto \quad \exp\{-\beta L(x) - (2\sigma^2)^{-1}||y - Kh(x)||^2\}, \qquad x \in \Omega. \tag{2.5}$$

The posterior distribution is still a Markov random field but with larger neighborhoods than in the prior because of blurring.

The MAP estimate is defined to be the mode over Ω of the posterior distribution of X given the observed record $Y = y$. Thus, we are faced with the problem of maximizing the right-hand side of Equation (2.5) over $x \in \Omega$.

3. FINDING THE MAP SUBPIXEL ESTIMATE

3.1 THE ALGORITHM

Our ultimate aim is to find the image x in the sample space Ω that maximizes the posterior probability $f_{X|Y}(x,y)$ defined by Equation (2.5). The sample space Ω, as defined in Section 2.3, is complex and it is difficult to find a good initial estimate in Ω directly. Instead, we proceed in stages, optimizing first over simpler image spaces in order to obtain a good starting image in Ω for the final optimization. At each stage we seek the maximum of

$$\phi(x) = \exp\{-\beta L(x) - (2\sigma^2)^{-1}||y - Kh(x)||^2\} \tag{3.1}$$

over images x in specified spaces, where $L(x)$ is the total edge length in image x and $h_i(x)$ the average value of x over pixel i. Our algorithm for seeking the MAP estimate is summarized as follows.

3.1.1 Algorithm 1: Subpixel Reconstruction

Stage 1. Full pixel reconstruction. Define Ω_1 to be the set of images in which each pixel is of a single color, c^1 or c^2. From a convenient starting point, search for the image in Ω_1 that maximizes $\phi(x)$, using simulated annealing based on the Gibbs sampler (see Sec. 3.6.1).

Stage 2. Initial subpixel estimation. Define Ω_2 to be the space of images in which each pixel takes one of the 14 states shown in Figure 2. Starting from the final reconstruction from Stage 1, search for the image in Ω_2 that maximizes $\phi(x)$, again using simulated annealing based on the Gibbs sampler.

Stage 3. Conversion to an image in Ω. Apply a deterministic algorithm (see Sec. 3.4) to convert the end product of Stage 2 to an image in Ω with as little modification as possible.

Stage 4. Final subpixel estimation. Starting from the reconstruction obtained in Stage 3, search for the image in Ω that maximizes $\phi(x)$. This time a form of simulated annealing based on the Metropolis algorithm is used (see Sec. 3.6.2).

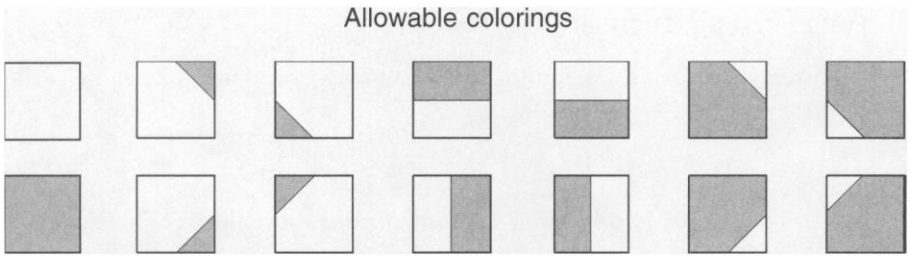

Figure 2. Stage 2 Proposals. This figure shows the 14 pixel colorings allowed in Stage 2. These 14 possibilities arise from constraining every edge segment to terminate midway along a pixel boundary.

3.2 EXAMPLE 2: ALGORITHM 1 APPLIED TO A SMALL IMAGE

Examples of the types of images produced in the four stages are shown in Figure 3 with the edge length associated with the center pixel in each 3×3 image highlighted by broken lines. Note that in Stage 1 edges lie on the pixel boundary only and in Stage 2 pixels can also be divided by lines with vertices located midway along a pixel edge. The intention in these first two stages is to produce an initial approximation to the MAP subpixel estimate in an unconstrained search over the spaces Ω_1 and Ω_2. There is no guarantee that the reconstruction at the end of Stage 2 will be in the set Ω as edges of regions of a particular color may lie along pixel boundaries rather than within pixels; even a method such as that of Geman, Geman, Graffigne, and Dong (1990) in which a penalty for "taboo" states is increased during the course of simulated annealing, does not necessarily ensure that taboo states will not arise. Thus, a deterministic algorithm is required in Stage 3 to amend any features that contravene the rules obeyed by images in Ω. After Stage 3, all edges lie within pixels and their vertices are at the midpoints of the pixel edges. From this point, the pixel-to-pixel route of each edge is fixed. Thus, the end product of Stage 3 determines a subset of Ω to be searched in Stage 4 when the vertices of each edge segment are allowed to move within their specified pixel edges. The starting temperature for simulated annealing in Stage 4 can be set to a high value in order to explore this subspace of Ω fully.

Note that it is not essential to fix the pixels in which edges lie throughout Stage 4. Jubb and Jennison (1991) described an algorithm that allows edges to be re-routed through a new sequence of pixels in their equivalent of our Stage 4. Such re-routing would certainly be desirable when simulating from the posterior distribution in Equation (2.5) over all of Ω. This problem is complicated by the dependency of the dimension of the parameter vector needed to define an image (one variable specifying the location of each vertex) on the number of edges present. Grenander and Miller (1994) and Srivastava, Miller, and Grenander (1991) have proposed jump-diffusion simulation which would allow regions of one color to appear or disappear and Green (1994, 1996) described an explicit class of MCMC methods that use reversible Metropolis–Hastings jumps between subspaces that might also be applicable in this context.

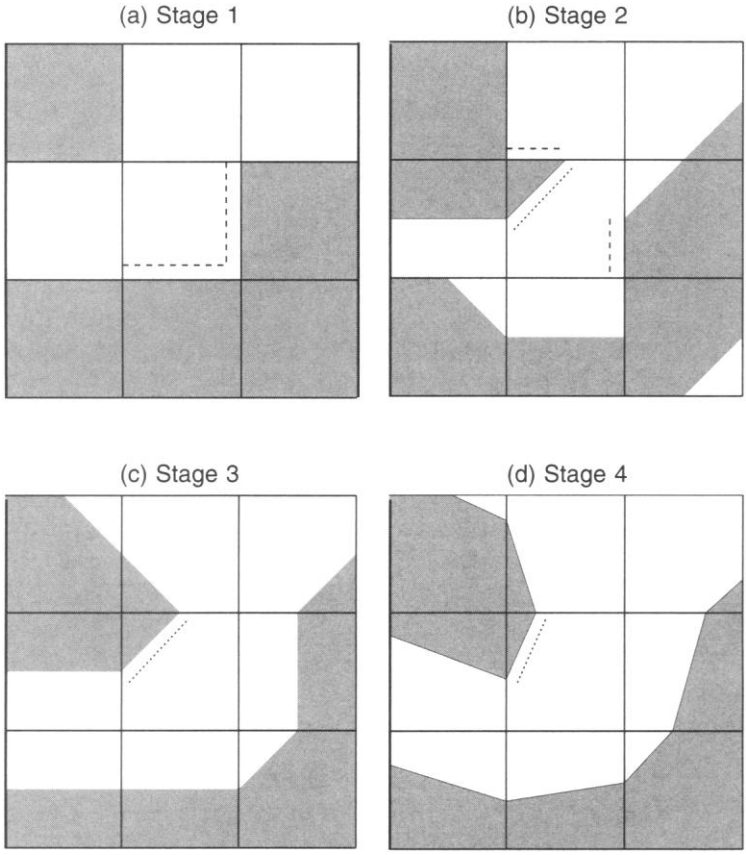

Figure 3. The Four Stages in a Reconstruction. Each 3×3 image is an example from the state space used in the specified stage of Algorithm 1. In each image the center pixel's contribution to the edge length L(x) is highlighted using dashed lines for edges lying on the pixel boundary and dotted lines for the edge inside the pixel.

3.3 EXAMPLE 3: RECONSTRUCTIONS FROM ALGORITHM 1 FOR A 16 X 16 IMAGE

As a more substantial example, Figure 4 shows steps in the reconstruction of the image depicted in Figure 1(d) (p. 183). The record $Y = \{Y_i \in \Re : i = 1, 2, \ldots, n\}$ used here is the 16×16 array of gray-level values shown in Figure 1(b). There is no blurring in this example, just independent, Gaussian noise with variance $\sigma^2 = .01$. The foreground and background colors are $c^1 = 1$ and $c^2 = 0$, respectively. The algorithm is initialized at the beginning of Stage 1 by thresholding the record Y at $1/2$. The parameter for the prior image model in Equation (2.2) is subjectively set at $\beta = 25$ and 50 sweeps of simulated annealing with a geometric cooling schedule are used in each of Stages 1, 2, and 4. These choices are discussed later in Section 4.

The initial image for Stage 1 is shown in Figure 4. The reconstructions at the end of Stages 2, 3, and 4 are also shown. In Stage 2, edges that contravene the rules defining Ω are liable to increase $L(x)$ and decrease $\phi(x)$. Thus, it is not surprising that the Stage 2 image infringes these rules in only a few isolated instances on the edges of the ring and in several contiguous pixels on the right side of the large black circle. These

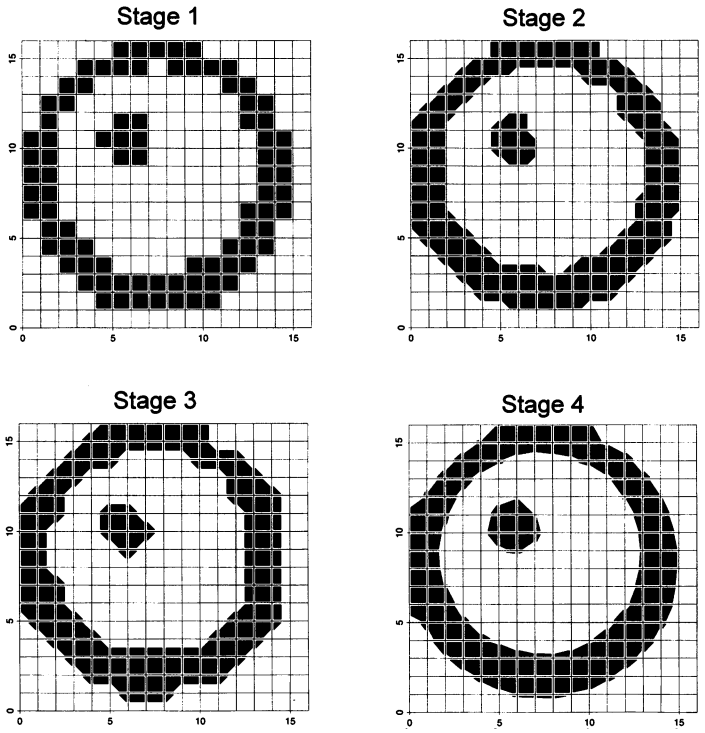

Figure 4. Stages in the Reconstruction of a 16×16 Image. These reconstructions illustrate each of the four stages in Algorithm 1. In the Stage 1 reconstruction, each pixel takes one of two colors. In Stage 2 each pixel is permitted to take one of the 14 states, shown in Figure 2, with no further restrictions. Only a few places require adjustment in Stage 3 in order to obtain an image in Ω. Note that the final reconstruction has a tendency to "cling" to the boundary as this reduces the total edge length within the image and so can increase $\phi(x)$.

pixels are rectified in Stage 3 and further processing in Stage 4 yields a reconstruction that approximates the original image well. The most noticeable errors occur where boundaries between black and white regions meet the image edge. Here distortions can arise as our prior model penalizes edges running close and almost parallel to the image edge. The model favors more abrupt termination of such edges at an earlier point.

3.4 THE CONVERSION ALGORITHM

An image in Ω_2 can be fully specified by stating the color present in each corner of each pixel. Note that of the 18 possible combinations of two colors only 14 are permitted. These are shown in Figure 2. The other possibilities are excluded because of the restriction to only one line segment crossing any one pixel, for simplicity. If an image in Ω_2 is also in the set Ω, the same color must be associated with all four pixel corners meeting at a vertex or with both pixel corners when two pixels meet at a side of the image. It is straightforward to check that this is a necessary and sufficient condition for an image in Ω_2 to be in Ω and we base our conversion algorithm on this fact.

3.4.1 Algorithm 2: Conversion from Ω_2 to Ω

Initialization. Use the final reconstruction from Stage 2 to assign a color to each of the four corners of each pixel.

Sweep 1. Visit each pixel of the image and each of the four corners within that pixel in a fixed raster scan, taking rows from the top of the image to the bottom and moving from left to right within each row. Observe the colors of the four adjacent pixel corners or two adjacent corners around the boundary, except at the four corners of the image.

- If all corners have the same color, leave them unchanged.
- If one color dominates by occupying three out of four corners, then recolor the minority corner to agree with the other three.
- If the colors are evenly split, one to one or two against two, calculate the average value of the record elements associated with all the pixels concerned and color all corners with the color that lies closest to the average value of the record elements.

Remark: Here we take the record element associated with a pixel to be the record element to whose mean the pixel makes the largest contribution.

Sweep 2. Visit pixels in a raster scan taking rows from the top of the image to the bottom and moving from left to right within each row.

- If the colors in the four corners of a pixel correspond to a pixel coloring in Figure 2, assign this coloring to the pixel.
- If not, we must have one pair of diagonally opposite corners of one color and the other pair of the other color. To rectify this, switch the value in the pixel's southeast corner and also the values in the corner of each neighboring pixel which meet at this vertex. The pixel currently being visited now has a pattern of corner values in agreement with a coloring in Figure 2 and we assign it this coloring.

In Sweep 1, we use information from the record to achieve an accurate reconstruction. We expect that this first sweep will resolve matters in most pixels and that very few pixels will be met in Sweep 2 that do not conform to a coloring in Figure 2. When such pixels do arise, the algorithm deals with them in a somewhat *ad hoc* manner, but one which is guaranteed to produce an image in Ω. Treatments that make greater use of record values are possible, but we see little benefit in creating a more complex procedure for handling a small number of border-line pixels.

3.5 EXAMPLE 4: CONVERSION TO AN IMAGE IN Ω

Figure 5 shows the conversion of the example in Figure 3 from Stage 2 to Stage 3 using Algorithm 2. The color of each corner of each pixel, from Stage 2 of Figure 3, is used to initialize Figure 5(a). During the first sweep, each corner of each pixel along with

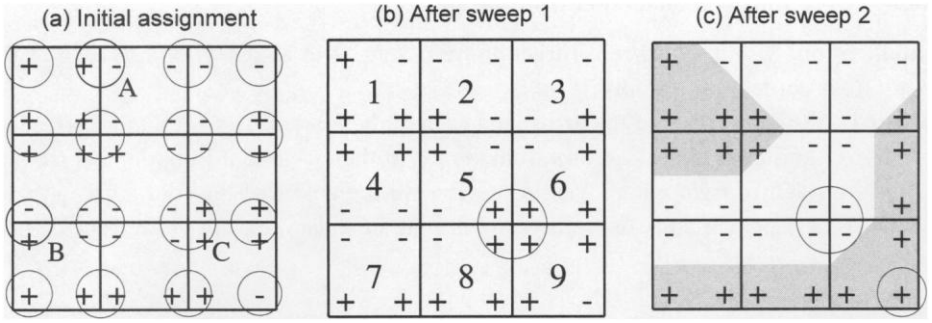

Figure 5. Conversion to an Image in Ω. This figure shows the effect of applying Algorithm 2 to the reconstruction from Stage 2 of Figure 3. In plot (a), the circles denote the grouping of corner values at pixel vertices. During Sweep 1, rules are applied to ensure that all corners meeting at a vertex have the same color. During Sweep 2, the corners within each pixel are used to decide the coloring for that pixel. To avoid an anomaly in pixel 5, the color of the southeast corner and its neighbors are flipped. This switch has a knock-on effect on pixel 9, so that the southeast corner of that pixel also changes.

its neighboring corners is processed and the circles in Figure 5(a) highlight the corners involved in each step. For the circles labeled A, B, and C, there is a tie between the two colors. This is resolved by considering the average of the records associated with those pixels.

During the second sweep of the image, the four corners within each pixel are considered and one of the 14 proposals shown in Figure 2 is inserted, based on the coloring of the four corners within that pixel. An ambiguity arises at the pixel labeled 5, in Figure 5(b), because the coloring is not one of the possibilities shown in Figure 2. The southeast corner of pixel 5 is flipped and this has a knock-on effect on pixel 9, which now has a pattern of colors similar to that of pixel 5. After switching the southeast corner, we obtain the result shown in Figure 5(c) which is the Stage 3 reconstruction of Figure 3.

3.6 SIMULATED ANNEALING

3.6.1 Simulated Annealing With the Gibbs Sampler

In the algorithm just described, an analytic approach to maximizing $\phi(x)$ within each stage is not feasible because of the high dimensionality of the spaces of images under consideration. Instead, we use the stochastic optimization algorithm of simulated annealing at each stage. Simulated annealing is based on an MCMC method for sampling from a target probability distribution. In our algorithm, we use simulated annealing based on the Gibbs sampler (Geman and Geman 1984) and the Metropolis–Hastings algorithm (Metropolis et al. 1953; Hastings 1970). The modification that enables simulated annealing to optimize rather than sample is the raising of the target distribution to higher and higher powers over the course of the algorithm and this places an increasing probability at the globally optimum state.

Simulated annealing based on the Gibbs sampler is used in Stages 1 and 2 of Algorithm 1 to seek the maximum of $\phi(x)$ over sample spaces Ω_1 and Ω_2, respectively. In

each iteration $t = 1, 2, \ldots, n$, the image is swept in a raster scan and pixels are updated in turn. Let x_{-i} denote the values of all pixels other than i in an image x. If the current image is x, just before pixel i is updated, then the Gibbs sampler would generate a new image x' with probability proportional to $\phi(x')$, for all x' in the sample space satisfying $x'_{-i} = x_{-i}$. So in Stage 1, pixel i can take either color c^1 or c^2 and in Stage 2 pixel i can take any of the 14 possible colorings shown in Figure 2. The modification in simulated annealing is that in iteration t, the probability of x' is proportional to $\{\phi(x')\}^{1/T(t)}$, where $T(t)$, referred to as temperature, decreases with t.

In our example, we used a geometric temperature schedule falling from initial temperature $a = 5$ to final temperature $b = .1$ in $N = 50$ sweeps,

$$T(t) \quad = \quad a(b/a)^{(t-1)/(N-1)}, \qquad t = 1, \ldots, N.$$

Following Geman, Geman, and Graffigne (1987), the image attaining the maximum value of $\phi(x)$ during the N sweeps was noted and the algorithm was rerun at zero temperature from this starting point, until convergence at a local maximum of $\phi(x)$. This last stage is equivalent to applying the strictly uphill search of iterated conditional modes (Besag 1986) from the starting point formed by simulated annealing.

3.6.2 Simulated Annealing With a Metropolis Update

The Stage 4 image space is continuous and it is convenient to use simulated annealing based on the Metropolis-Hastings algorithm. After Stage 3, the coloring of the image is specified up to the location of the vertices of the linked edge segments, each of which can lie within a specified pixel edge.

Suppose the vector $\theta = (\theta_1, \ldots, \theta_r)$ contains parameters, taking values in $(0, 1)$, which specify the location of the vertices. Let $x(\theta)$ denote the image defined by θ, let θ_{-i} denote the values of θ in all elements other than i and let θ' be the vector with ith element θ'_i and $\theta'_{-i} = \theta_{-i}$.

The goal is to find the value that maximizes $\phi\{x(\theta)\}$ over $\theta \in (0, 1)^r$. The Metropolis algorithm is designed to sample from the distribution of θ with density proportional to $\phi\{x(\theta)\}$.

3.6.3 Algorithm 3: Sweeping an Image in Ω

- Sweep the image, updating every vertex location once. In effect, the vector θ is swept and its elements θ_i are updated in turn, where $i = 1, 2, \ldots, r$.

  Remark: Typically the number of updates required to sweep an image is far less than the number of pixels in the image ($r \ll n$) as only a few pixels contain two colors. This reduces the amount of computation required.

- In updating θ_i a proposal value θ'_i is drawn from the uniform density on $(0, 1)$.

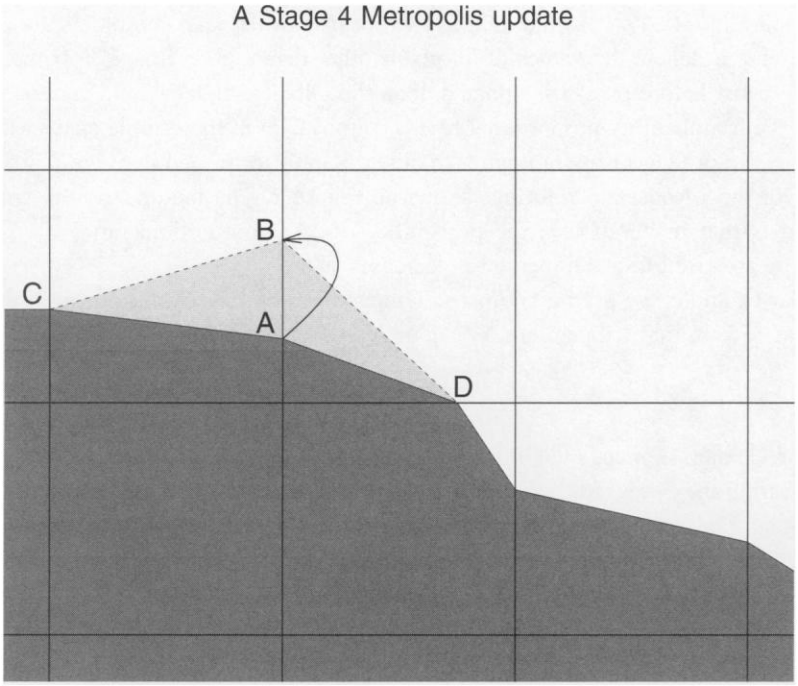

Figure 6. A Stage 4 Move. In Stage 4, the Metropolis algorithm proposes a move from the point A *to a point* B *randomly selected along the common pixel boundary between the two linked edges. The move from the point* A *to the point* B *would add an additional edge length of* |CB| + |BD| − |CA| − |AD| *and change the proportion of black in the two pixels by the areas of triangles* ABC *and* ABD. *The choice between moving from* A *to* B *or staying at* A *is then made according to Equation (3.3).*

Replace θ_i by θ_i' with probability

$$\alpha(\theta, \theta') = \min\left[1, \phi\{x(\theta')\}/\phi\{x(\theta)\}\right] \tag{3.2}$$

otherwise the value θ_i is retained.

- Increment i by one and go to the previous step to update the next element θ_{i+1}, until $i = r$.

The effect of changing a single vertex location θ_i is illustrated in Figure 6. The coloring of two pixels is *simultaneously* affected as the two edge-segments meeting at the vertex in question are repositioned. The only modification with the simulated annealing algorithm is the replacement of Equation (3.2) by

$$\alpha(\theta, \theta') \quad = \quad \min\left[1, \left\{\phi\{x(\theta')\}/\phi\{x(\theta)\}\right\}^{1/T(t)}\right]. \tag{3.3}$$

The optimization in Stage 4 is not especially onerous, because the number of vertices is typically much less than the number of pixels in the image.

4. AN APPLICATION IN MICROSCOPY

In this section we illustrate how the subpixel model can be used to restore a blurred microscopic image of the type considered by Hitchcock and Glasbey (1994). The observed image of a fungus mycelium is shown on an opaque background in Figure 7(a). It consists of a network of hyphae from a single fungal organism growing on a microscope slide which has been coated with a cellophane-coated nutrient agar (Ritz and Crawford 1990). A fundamental characteristic of this class of fungi is their mycelial growth form, which is an effective mechanism by which habitats can be explored in order to find new food bases and subsequently exploit them. The fungi need to search their habitat efficiently to maximize their chances of finding food while simultaneously minimizing the amount of energy consumed. Some evidence suggests that the fungi forage according to the spatial distribution of their food. Traditionally, results were based on qualitative data from field and laboratory experiments. Today, image analysis can be used to examine the spatial structure of the fungal hyphae in relation to their environment. Glasbey and Horgan (1994, chaps. 5 and 6) used a thinning operation to get a skeleton that is one pixel thick to estimate the total length of hyphae. They allow for effects due to lines being represented as lattice points rather than being in continuous space. Crawford, Ritz, and Young (1993) discussed further work on fungal morphology and its relationship to soil structure.

4.1 EXAMPLE 5: FUNGAL MYCELIUM RECONSTRUCTION—1

At each of the 41×51 pixels in Figure 7(a), a blurred gray-scale record element in the range $[0, 255]$ is observed. Our objective is to reconstruct, without noise and blurring, the true image that consists of two long fungus arms plus some isolated fungus objects, several of which are only a few pixels in size.

The foreground and background colors were estimated from a histogram of the data to be $c^1 = 240$ and $c^2 = 30$, respectively, but the algorithm is not very sensitive to these parameters. The record variance $\sigma^2 = 4$ was estimated from samples taken from parts of the image that contain just one color. The blurring kernel was estimated by taking cross-sections through the data. Where we have a discontinuity in the image, we match the shape of the observed features to the theoretical features of an ideal edge. If no blurring were present, a one-dimensional slice through the image would show most record element values near c^1 or c^2 and the value would change abruptly at the boundary between these two levels. The rate at which the cross-section moves from the one color level to the other provides an estimate for the variance of the bivariate Gaussian blurring kernel of .75. The blurring kernel was truncated to a 3×3 window, as kernel weights outside this window were close to zero.

For Stages 1, 2, and 4, simulated annealing was used with a starting temperature of 5, for 100 sweeps of the image on a geometric schedule, to finish at a temperature of .1, followed by a strictly uphill search to convergence. Other temperature schedules were also considered but the results were not sensitive to small changes in these parameters.

Figures 7(b) and (c) show reconstructions where the smoothing parameter values are $\beta = 50$ and $\beta = 450$, respectively. Both reconstructions have an artificial boundary

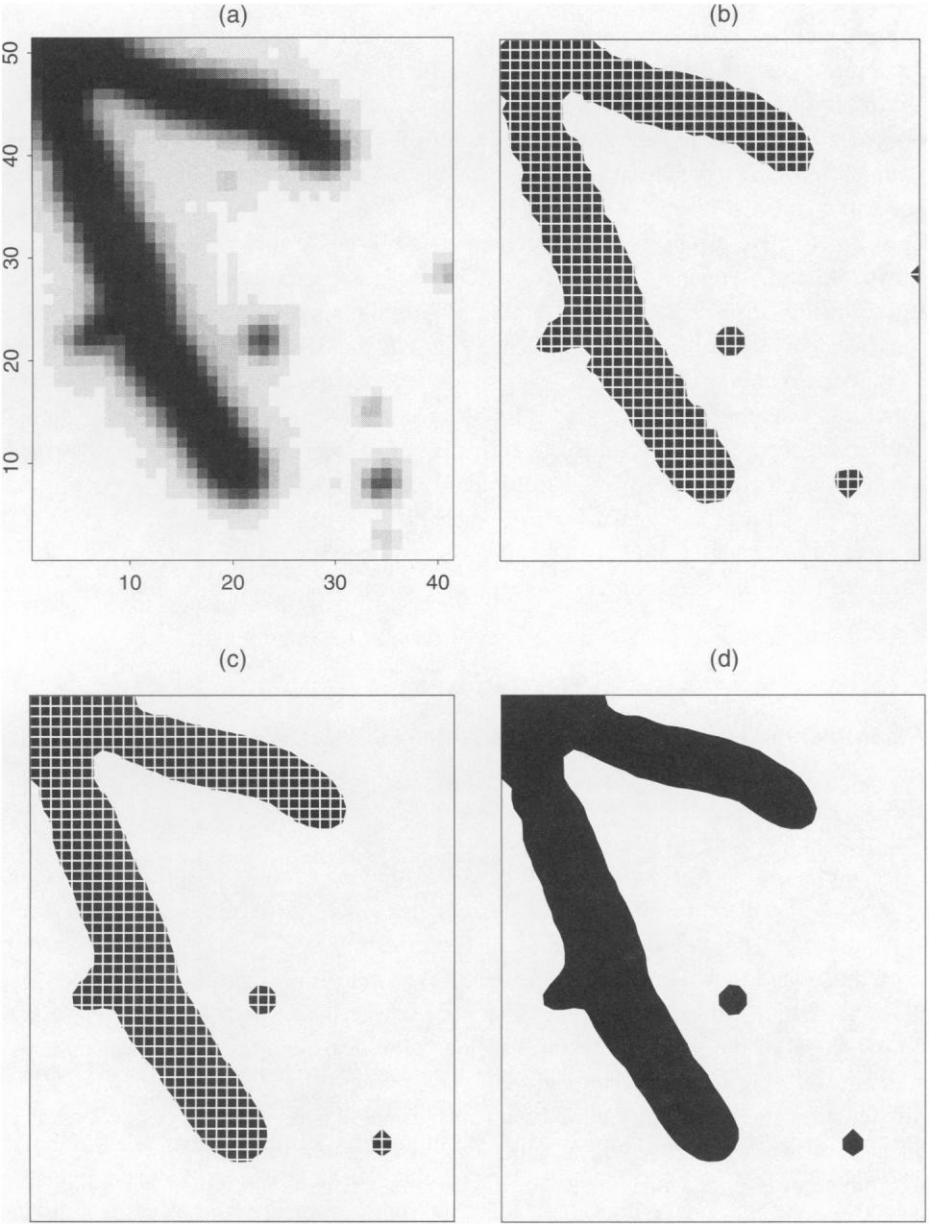

Figure 7. Fungal Mycelium Reconstruction—1. Plot (a) shows a part of a fungus mycelium growing on a microscope slide. The image is 41×51 pixels in size. Plots (b) and (c) show reconstructions of the image where the smoothing parameters are $\beta = 50$ and $\beta = 450$, respectively, with an artificial boundary imposed to highlight the individual pixels. The reconstruction for $\beta = 450$ without the artificial boundary is shown in plot (d).

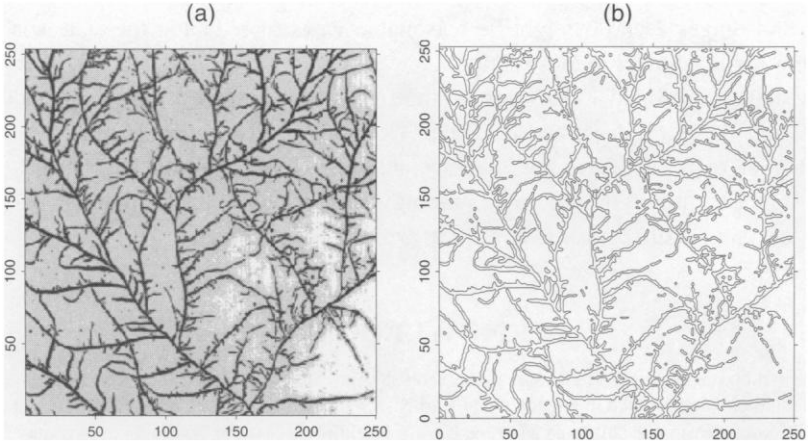

Figure 8. Fungal Mycelium Reconstruction—2. Plot (a) shows a microscopic view of a more complex fungal structure. Individual fungal arms are only 2–3 pixels wide on average. Plot (b) shows a reconstruction of the image in Plot (a).

around each pixel to illustrate the amount of subpixel detail recovered. The reconstruction for $\beta = 450$ without this artificial boundary is also shown in Figure 7(d). Generally, the boundaries become smoother and very small objects are more likely to disappear as the value for β increases, because there is a greater penalty for edge length.

4.2 EXAMPLE 6: FUNGAL MYCELIUM RECONSTRUCTION—2

We briefly illustrate a second example to show that the algorithm can be used with more complex images. Figure 8(a) shows a larger record than in Figure 7(a). Figure 8(b) shows the final reconstruction using the same parameters as in the previous example except $\beta = 300$. Just the edges of the fungal arms are shown as this gives a clearer view of the reconstruction.

5. CONCLUSIONS

A multistage algorithm is used to refine the reconstructed image in stages, down to a subpixel level. The final reconstruction segments the pixels in the image by allowing boundaries to consist of piecewise continuous straight lines across pixels. Thus, reconstructed boundaries are not constrained to lie along the boundary between pixels.

Reconstructing the image in stages, where the final estimate from one stage is used as the starting point for the next stage, breaks a large optimization problem down into a more manageable format. Furthermore, each stage in the reconstruction has a good starting point so the amount of work required at each stage is reduced.

The algorithm does not require knowledge of the number, shape, or orientation of the objects in the image. The proposed method is computationally intensive so it is most suitable for reconstructing small objects; although Example 6 shows that the algorithm can be applied to large images. The algorithm is not dependent on the particular case of Gaussian blurring nor additive Gaussian noise that we have concentrated on in this

article, and Stages 2 and 3 might be reasonable ways of reducing the state space when searching for a subpixel reconstruction.

We have developed extensions of the algorithm described here to deal with 3D images, and these are described by Gavin (1995). We also note that our methods have the potential to be extended to images containing several colors: the basic algorithm can be applied within any region containing just two colors and additional features are needed only in small regions where three or more colors meet.

ACKNOWLEDGMENTS

The authors thank David Hitchcock at the Scottish Agricultural Statistical Services for bringing to our attention the problem discussed in Section 4 and Karl Ritz at the Scottish Crop Research Institute for supplying the data. Thanks also to our colleague Merrilee Hurn for helpful discussions and to an anonymous referee for some constructive comments.

[Received February 1995. Revised March 1996.]

REFERENCES

Besag, J. E. (1986), "On the Statistical Analysis of Dirty Pictures" (with discussion), *Journal of the Royal Statistical Society,* Series B, 48, 259–302.

Besag, J. E., Green, P. J., Higdon, D., and Mengersen, K. (1995), "Bayesian Computation and Stochastic Systems," *Statistical Science,* 10, 3–66.

Bhattacharya, A., Reddy, C. S. S., and Srivastav, S. K. (1993), "Remote Sensing for Active Volcano Monitoring in Barren Island, India," *Photogrammetric Engineering and Remote Sensing,* LIX (8), 1293.

Boult, T. E., and Wolberg, G. (1993), "Local Image Reconstruction and Subpixel Restoration Algorithms," *CVGIP: Graphical Models and Image Processing,* 55, 63–77.

Crawford, J. W., Ritz, K., and Young, I. M. (1993), "Quantification of Fungal Morphology, Gaseous Transport and Microbial Dynamics in Soil: An Integrated Framework Using Fractal Geometry," *Geoderma,* 56, 157–172.

Foody, G. M. (1994), "Ordinal-Level Classification of Sub-Pixel Tropical Forest Cover," *Photogrammetric Engineering and Remote Sensing,* LX (1), 61.

Gatherer, A., and Meng, T. (1992), "Robust Subpixel Alignment in Lithography," *Journal of Vacuum Science & Technology B,* 10, 2662–2666.

Gavin, J. (1995), "Subpixel Image Analysis," unpublished PhD thesis, University of Bath.

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6, 721–741.

Gernan, D., Geman, S., and Graffigne, C. (1987), "Locating Texture and Object Boundaries," in *Pattern Recognition Theory and Application,* (vol. F30), eds. P. A. Devijver and J. Kittler, Berlin: Springer-Verlag, pp. 165–177.

Geman, D., Geman, S., Graffigne, C., and Dong, P. (1990), "Boundary Detection by Constrained Optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 12, 609–628.

Glasbey, C. A., and Horgan, G. W. (1994), *Image Analysis for the Biological Sciences,* Chichester, England: John Wiley & Sons.

Green, P. J. (1994), Discussion of "Representations of Knowledge in Complex Systems," by U. Grenander and M. Miller, *Journal of the Royal Statistical Society,* Series B, 56, 589–590.

——— (1995), "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika,* 82, 711–732.

Grenander, U. (1983), "Tutorial in Pattern Theory," unpublished technical report, Brown University, Div. of Applied Mathematics.

Grenander, U., and Miller, M. (1994), "Representations of Knowledge in Complex Systems" (with discussion), *Journal of the Royal Statistical Society,* Series B, 56, 549–603.

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika,* 57, 97–109.

Hitchcock, D., and Glasbey, C. A. (1994), "Binary Image Restoration at Subpixel Resolution From Multi-level Data," unpublished technical report, Scottish Agricultural Statistics Service, JCMB, King's Buildings, Edinburgh, EH9 3JZ, Scotland.

Huertas, A., and Medioni, G. (1986), "Detection of Intensity Changes With Subpixel Accuracy Using Laplacian–Gaussian Masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 8, 651–663.

Jasinski, M. F. (1990), "Sensitivity of the Normalized Difference Vegetation Index to Subpixel Canopy Cover, Soil Albedo and Pixel Scale," *Remote Sensing of the Environment,* 32, 169–187.

Jasinski, M. F., and Eagleson, P. S. (1990), "Estimation of Subpixel Vegetation Cover Using Red Infrared Scattergrams," *IEEE Transactions on Geoscience and Remote Sensing,* 28, 253–267.

Jennison, C., and Jubb, M. (1988), "Statistical Image Restoration and Refinement," in *Information Processing in Medical Imaging,* eds. C. N. de Graaf and M. A. Viergerer, New York: Plenum Press, pp. 255–262.

Jubb, M., and Jennison, C. (1991), "Aggregation and Refinement in Binary Image Restoration," in *Spatial Statistics and Imaging, IMS Lecture Notes,* (vol. 20), ed. A. Possolo, Hayward, CA: pp. 150–162.

Koplowitz, J., and Sundar Raj, A. P. (1987), "A Robust Filtering Algorithm for Subpixel Reconstruction of Chain Coded Line Drawings," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 9, 451–457.

Lorensen, W. E., and Cline, H. E. (1987), "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics,* 91, 163–169.

Lyvers, E. P., Mitchell, O. R. Akey, M. L., and Reeves, A. P. (1989), "Subpixel Measurements Using a Moment-Based Edge Operator," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 11, 1293–1309.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics,* 21, 1087–1092.

Oakley, J. P., and Shann, R. T. (1991), "Efficient Method for Finding the Position of Object Boundaries to Sub-Pixel Precision," *Image and Vision Computing,* 9, 262–272.

Ritz, K., and Crawford, J. W. (1990), "Quantification of the Fractal Nature of Colonies of *Trichoderma viride,*" *Mycological Research,* 94, 1138–1141.

Sandor, T., and Spears, J. R. (1985), "Statistical Considerations on the Precision of Assessing Blood Vessel Diameter in Cine Coronary Angiography," *Computers and Biomedical Research,* 18, 531–543.

Sriraman, R., Koplowitz, J., and Mohan, S. (1989), "Tree Searched Chain Coding for Subpixel Reconstruction of Planar Curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 11, 95–104.

Srivastava, A., Miller, M. I., and Grenander, U. (1991), "Jump Diffusion Processes for Object Tracking and Direction Tracking and Direction Finding," in *Proceedings of the 29th Annual Allerton Conference on Communication, Control and Computing,* pp. 563–570.

Szirányi, T. (1992), "Statistical Subpixel Measurement Technology of Light-Beam Width and Profiles," *Optics and Lasers in Engineering,* 16, 1–15.

Tabatabai, A. J., and Mitchell, O. R. (1984), "Edge Location to Subpixel Values in Digital Irnagery," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6, 188–201.

Weir, N., and Djorgovski, S. (1991), "A Subpixel Deconvolution Method for Astronomical Images," *Fundamental Theories of Physics,* 43, 275–283.

West, G. A. W., and Clarke, T. A. (1990), "A Survey and Examination of Subpixel Measurernent Techniques," *Proceedings of the Society of Photo-optical Instrumentation Engineers,* 1395, pp. 456–463.

Young, R. A. (1987), "Locating Industrial Parts With Subpixel Accuracies," *Proceedings of the Society of Photo-optical Instrumentation Engineers,* 728, pp. 2–9.