January 1997

LIDS-P-2378

Research Supported By:

National Science Foundation Graduate
Research Fellowship
ONR under Grant N00014-91-J-1004
AFOSR under Grant F49620-95-1-0083
Boston Univ. under Grant GC123919NGN
NIH under Grant NINDS 1 R01 NS34189

MULTISCALE STATISTICAL METHODS FOR THE SEGMENTATION OF SIGNALS AND IMAGES

M. K. Sshneider, P. W. Fieguth, W. Clem Karl, A. S. Willsky

Multiscale Statistical Methods for the Segmentation of Signals and Images

M. K. Schneider, P. W. Fieguth, W. Clem Karl, A. S. Willsky

EDICS IP 1.5

M. K. Schneider and A. S. Willsky are affiliated with the Laboratory for Information and Decision Systems; Massachusetts Institute of Technology; 77 Mass Ave.; Cambridge, MA 02139 USA. P. W. Fieguth is affiliated with the Department of Systems Design Engineering; University of Waterloo; Waterloo, Ontario, Canada N2L-3G1. W. Clem Karl is affiliated with the Department of Electrical, Computer, and Systems Engineering; Boston University; 44 Cummington St; Boston, MA 02215 USA.

This material is based upon work supported by a National Science Foundation Graduate Research Fellowship, by ONR under Grant N00014-91-J-1004, by AFOSR under Grant F49620-95-1-0083, by Boston Univ. under Grant GC123919NGN, and by NIH under Grant NINDS 1 R01 NS34189.

Abstract

This paper addresses the problem of segmenting a signal or an image into homogeneous regions across whose boundaries there are abrupt changes in value. Motivated by large problems arising in certain scientific applications, such as medical imaging and remote sensing, two objectives for a segmentation algorithm are laid out: it should be computationally efficient and capable of generating statistics for the errors in the estimates of the homogeneous regions and boundary locations. The starting point for the development of a suitable algorithm is a variational approach to segmentation [1]. This paper then develops a precise statistical interpretation of a one-dimensional version of this variational approach to segmentation. The one-dimensional segmentation algorithm that arises as a result of this analysis is computationally efficient and capable of generating error statistics. A straightforward extension of this algorithm to two dimensions would incorporate recursive procedures for computing estimates of inhomogeneous Gaussian Markov random fields. Such procedures require an unacceptably large number of operations. To meet the objective of developing a computationally efficient algorithm, the use of recently developed multiscale statistical methods is investigated. This results in the development of a segmentation algorithm which is not only computationally efficient but also capable of generating error statistics, as desired.

I. Introduction

Many applications require segmenting an image, often corrupted by noise, into homogeneous regions bounded by curves across which the image intensity changes abruptly. In some cases, the principal interest is in obtaining an estimate of the boundaries. In others, the primary goal is to obtain estimates within each homogeneous region without undesirable smoothing across edges. For many scientific applications, such as remote sensing, one is interested in obtaining not only the estimates but also statistics for the errors in the estimates, which allow one to quantitatively evaluate the quality of the estimates. Due to the volume of data involved, automating this process is important. Although, there have been many approaches to the problems of automatic edge detection and segmentation (*e.g.* [2] - [5]), few, if any, address the issue of error statistics.

Of interest here are methods derived from models that lend themselves to statistical interpretations based upon which one can define error statistics. The starting point for the work in this paper is a variational formulation of the segmentation problem resulting from work by Mumford, Shah, Ambrosio, and Tortorelli [1], [4], [5], [6], [7]. One can prove that minimizers of the functionals for segmentation posed in these papers exist and have a number of mathematical properties that are intuitively appropriate for a segmentation. However, this deterministic variational approach to segmentation does not lend itself to a discussion of error statistics. A natural setting for this discussion is that of Bayesian estimation. One of the contributions of this paper is to provide and thoroughly characterize a Bayesian statistical interpretation of a variational approach to segmentation [1]. In particular, this paper demonstrates how such a statistical interpretation allows one to compute error statistics as well as estimates and contains a careful evaluation of the nature and quality of information provided by these statistics.

While the Bayesian interpretation can be equally well applied to both one and two dimensional signals, there is a significant difference in computational complexity in solving the resulting problems in these two different settings. As a consequence, a thorough analysis of the problem and, in particular, the error statistics is presented first in a one-dimensional (1-D) setting. The 1-D results are quite good and demonstrate that the algorithm can be successfully applied to detecting abrupt changes in 1-D signals. The major challenge in extending this approach to the two-dimensional (2-D) case is the development of computationally feasible solutions that yield both estimates and error statistics. The 2-D Bayesian estimation problems that arise from the type of variational problem of interest here make use of a particular class of priors, Markov random field priors. Unfortunately, the use of such priors typically leads to estimation problems that require a large number of computations to generate exact estimates and a prohibitively large computational load to calculate error statistics. More precisely, exactly solving the estimation problem, including calculation of error statistics, requires a per-pixel computational load that grows with image size. In contrast, our objective here is to develop an algorithm with constant per-pixel complexity that also produces useful error statistics.

There are two possible approaches to achieving such an objective, namely *approximating the solution* (*i.e.* replacing the solution to the estimation problem with one which is easier to compute) or *approximating the problem* (*i.e.* replace the estimation problem with one which has similar characteristics but which results in an exact estimation algorithm with the desired complexity). An approach of the former type is described in [8]. In this paper, an approach of the latter type is developed by altering the prior model appearing in the problem formulation. In particular, this paper examines the usefulness of multiscale prior models for image segmentation. Multiscale models, which were introduced and studied in [9] admit algorithms with constant per-pixel complexity for the calculation of both estimates and error variances, and they have also been shown to be useful in defining alternative approaches to other problems in computer vision which are often posed in a variational context [9], [10], [11], [12]. All of these previous methods, however, dealt with problems that resulted in linear estimation problems and algorithms. In contrast, image segmentation is fundamentally a nonlinear problem, and thus, this paper represents the first work on using multiscale stochastic models to solve a nonlinear problem in image processing and computer vision. As the results of this paper show, the algorithm that results not only has modest computational loads but also yields good performance.

The next section provides background on the variational methods in segmentation that form our point of departure. Sections III and IV then focus on the 1-D case. A precise statistical interpretation of the variational formulation is presented, leading to an algorithm in which both estimates and error statistics are calculated. The results of numerical experiments characterize the nature and quality of the information provided by these error statistics. Sections V to VI are devoted to the 2-D problem, including the development of a multiscale approach to the problem. To demonstrate the utility of this approach, numerical results are presented for the segmentation of an MRI brain image and satellite imagery of the Gulf Stream.

II. Variational Methods in Image Segmentation

Mumford and Shah [4], [5] have proposed formulating the image segmentation problem as the minimization of the following functional:

$$E(f,B) = \int\int_{\Omega} r^{-1}(g-f)^2 dxdy + \lambda \int\int_{\Omega - B} |\nabla f|^2 dxdy + \beta |B| \tag{1}$$

where Ω is the image domain, $g : \Omega \to \mathbf{R}$ is the image data, $f : \Omega \to \mathbf{R}$ is a piecewise smooth approximation to g, B is the union of segment boundaries, and $|B|$ is the length of B. The first term places a penalty on deviations of f from the data g; the second term ensures that f is smooth except at edge locations; and the third term penalizes spurious edges. The constants r, λ and β control the degree of interaction between the terms and ultimately determine the edginess of the final segmentation. The functional (1) has many nice mathematical and psychovisual properties [3]. The disadvantage of using this functional for segmentation is that computing minimizers is difficult because of the discrete nature of the edge term, $\beta|B|$.

Ambrosio and Tortorelli [6], [7] attempt to solve some of the computational difficulties associated with (1) by constructing a family of functionals whose minima converge to a minimum of (1). One such family lies at the core of a segmentation algorithm developed by Shah [1] and extended by Pien and Gauch [13] among others. The computational difficulties associated with the edge term are circumvented by introducing a continuous-valued edge function instead. A member of this family of functionals, parameterized by ρ, is of the form

$$E(f,s) = \int\int_\Omega \left\{ r^{-1}(g-f)^2 + \lambda|\nabla f|^2(1-s)^2 + \frac{\beta}{2}(\rho|\nabla s|^2 + \frac{s^2}{\rho}) \right\} dxdy, \tag{2}$$

where $s : \Omega \to [0,1]$ is an edge function, indicating the presence of an edge where it takes values close to one. The first and second terms constrain the approximating surface f to match the data as best as possible and also to be smooth in those places where s is close to zero, indicating the lack of an edge. The third term places constraints on the amount of edginess in the image. As shown in [7], the minima of (2) converge to a minimum of (1) as $\rho \to 0$.

The general approach Shah and Pien use to minimize (2) is coordinate descent: one alternates between fixing s and minimizing

$$E_s = \int\int_\Omega (r^{-1}(g-f)^2 + \lambda|\nabla f|^2(1-s)^2)dxdy \tag{3}$$

over possible f, and fixing f and minimizing

$$E_f = \int\int_\Omega (\lambda|\nabla f|^2(1-s)^2 + \frac{\beta}{2}(\rho|\nabla s|^2 + \frac{s^2}{\rho}))dxdy. \tag{4}$$

over possible s. Based on empirical evidence, Shah [1], and Pien and Gauch [13] have noted that this coordinate descent scheme converges to a reasonable solution and that the results are not significantly affected by the initial condition or whether one starts by estimating f or s.

III. Statistical Interpretation of Segmentation in One Dimension

Consider a prototypical quadratic minimization problem: minimize

$$E(f) = r^{-1}||g-f||^2 + \lambda||Lf||^2, \tag{5}$$

where f and g are vectors consisting of a lexicographic ordering of pixels in an image and L is a matrix chosen to ensure that the minimizer of (5) is smooth (*e.g.* L could take first differences of nearest neighbors as an approximation of a derivative). The function $\hat{f}$ that minimizes (5) is also the Bayes least-squares estimate of a process f based on the measurement

$$g = f + \sqrt{r}v \tag{6}$$

and prior probabilistic model for f given by

$$\sqrt{\lambda}Lf = w, \tag{7}$$

where v and w are independent Gaussian random vectors with identity covariance. Thus, one can view the problem at hand from the perspective of optimization or of statistical estimation.

The main advantage of this interpretation is that it casts the problem into a probabilistic framework in which it is natural to examine the accuracy of the resulting estimates. This is especially relevant in scientific applications such as remote sensing, as it provides a quantitative basis for assessing the statistical significance of the resulting estimate (*e.g.* for assessing if features in a reconstruction are meaningful or statistically insignificant artifacts). In addition, this statistical formulation brings into focus the role played by the regularization term as a prior model, opening up the possibility of using alternate models that offer certain advantages.

To proceed with the statistical interpretation of (2), consider first a discretized version of (2) in one dimension. The 1-D version of (2) is given by

$$E(f,s) = \int\int_{\Omega} (r^{-1}(g(x) - f(x))^2 + \lambda|\frac{df(x)}{dx}|^2(1 - s(x))^2 + \frac{\beta}{2}(\rho|\frac{ds(x)}{dx}|^2 + \frac{s(x)^2}{\rho}))dx. \tag{8}$$

One possibility for discretizing this functional is to replace the functions $f(x)$, $g(x)$, and $s(x)$ with regularly spaced collections of samples f_i, g_i, and s_i; the integrals with sums over i; and the derivatives with first differences. The result is the discrete functional

$$E(f,s) = r^{-1}\sum_{i=1}^{n}(f_i - g_i)^2 + \lambda\sum_{i=1}^{n-1}(1 - s_i)^2(f_{i+1} - f_i)^2 + \frac{\beta}{2}(\rho\sum_{i=1}^{n-2}(s_{i+1} - s_i)^2 + \frac{1}{\rho}\sum_{i=1}^{n-1}s_i^2), \tag{9}$$

where n denotes the number of data points g_i. As was done in [1], [13], one can use coordinate descent to minimize (9), thereby decomposing this complex problem into two simpler ones.

The problem of fixing s and finding the f that minimizes (9) is equivalent to finding the f that minimizes the discrete functional

$$E_s(f) = r^{-1}\sum_{i=1}^{n}(f_i - g_i)^2 + \sum_{i=1}^{n-1}\lambda(1 - s_i)^2(f_{i+1} - f_i)^2. \tag{10}$$

A slightly more compact form can be written by collecting the samples f_i, g_i, and s_i into vectors f and $g \in \mathbf{R}^n$ and $s \in \mathbf{R}^{n-1}$. Specifically, define the $(n-1) \times n$ matrix

$$L_n = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix} \tag{11}$$

and the diagonal matrix $S = \text{diag}(1 - s_1, \ldots, 1 - s_{n-1})$. Then, (10) simplifies to

$$E_s(f) = ||f - g||^2_{r^{-1}I} + \lambda ||L_n f||^2_{S^T S}, \tag{12}$$

where $||x||^2_W = x^T W x$. Finding the minimum of E_s for fixed invertible S is then equivalent to finding the least-squares estimate of f assuming the following measurement and prior model:

$$\begin{aligned} g &= f + \sqrt{r} v^f & (13) \\ L_n f &= \frac{1}{\sqrt{\lambda}} S^{-1} w^f & (14) \end{aligned}$$

where v^f and w^f are independent and Gaussian, with covariance I. Notice that for i such that $s_i \approx 1$, the multiplier of w^f_i, $1/(1 - s_i)$, is very large. Thus, at these locations, the variance of $f_{i+1} - f_i$ in the prior model is high, and a least-squares estimator will allow big jumps to occur in the estimate of f. This is exactly what one wants the estimator to do at edge locations.

The problem of fixing f and finding s that minimizes (9) is equivalent to minimizing

$$E_f(s) = \lambda \sum_{i=1}^{n-1} (f_{i+1} - f_i)^2 (1 - s_i)^2 + \frac{\beta\rho}{2} \sum_{i=1}^{n-2} (s_{i+1} - s_i)^2 + \sum_{i=1}^{n-1} s_i^2. \tag{15}$$

Defining $a_i = \lambda(f_{i+1} - f_i)^2$, $b = \beta/2\rho$, $c = \beta\rho/2$, and $\gamma_i = a_i/(a_i + b)$, one finds that, after completing squares, minimizing (15) is equivalent to minimizing

$$E_f(s) = \sum_{i=1}^{n-1} (a_i + b)(\gamma_i - s_i)^2 + c \sum_{i=1}^{n-2} (s_{i+1} - s_i)^2. \tag{16}$$

By defining the diagonal matrix $A = \text{diag}(\sqrt{\lambda(L_n f)_1 + b}, \ldots, \sqrt{\lambda(L_n f)_{(n-1)} + b})$, and the vector $\gamma = \begin{pmatrix} \frac{\lambda(L_n f)_1^2}{\lambda(L_n f)_1^2 + b} & \cdots & \frac{\lambda(L_n f)^2_{(n-1)}}{\lambda(L_n f)^2_{(n-1)} + b} \end{pmatrix}^T$, where $(Lf)_k$ corresponds to the kth row of Lf, one can rewrite (16) as

$$E_f(s) = ||\gamma - s||^2_{A^T A} + c||L_{n-1} s||^2. \tag{17}$$

In the original functional (2), s is constrained to lie within $[0, 1]$. If one removes this constraint, the problem of finding the s that minimizes (17) is equivalent to the problem of estimating s given the following measurement and prior model:

$$\begin{aligned} \gamma &= s + A^{-1} v^s & (18) \\ L_{n-1} s &= \frac{1}{\sqrt{c}} w^s & (19) \end{aligned}$$

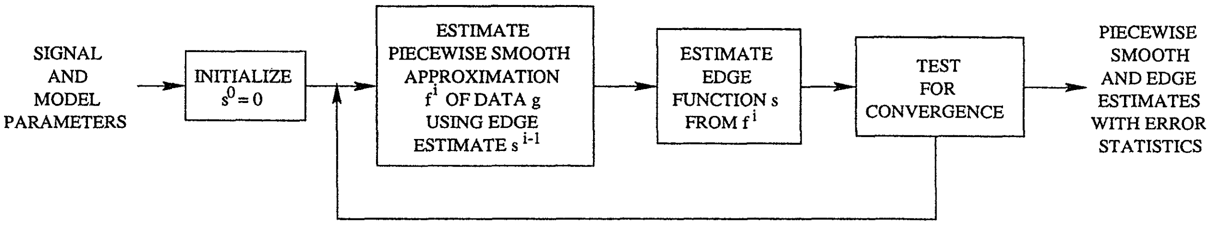

SIGNAL AND MODEL PARAMETERS
INITIALIZE $s^0 = 0$
ESTIMATE PIECEWISE SMOOTH APPROXIMATION f^i OF DATA g USING EDGE ESTIMATE s^{i-1}
ESTIMATE EDGE FUNCTION s FROM f^i
TEST FOR CONVERGENCE
PIECEWISE SMOOTH AND EDGE ESTIMATES WITH ERROR STATISTICS

Fig. 1. The one dimensional segmentation algorithm.

where v^s and w^s are independent and Gaussian with covariance I. Notice that γ plays the role of an observation of the edge function, and its components take on values near one where the difference between consecutive samples of f is large and near zero where the difference is small. Observe also that γ_i lies within $[0, 1)$; thus, the first term in (17) provides an increased penalty for functions s that do not stay within $[0, 1]$. This is desirable because a solution to the unconstrained minimization of (17) that lies within $[0, 1]$ is an optimal solution of the constrained problem. As it turns out, this is often the case, as discussed in Section IV.

As an aside, we note that one of the benefits of formulating a minimization problem in terms of statistics is that it yields a natural interpretation of the parameters, which, in turn, can be used to form a loose set of guidelines for picking parameter values suitable for a particular segmentation application. For example, adjusting λ corresponds to matching the expected variability in f to that which is anticipated in an application. The reader is referred to [14] for a discussion of this and similar interpretations for the other parameters.

IV. Numerical Results

Based on the estimation problem formulations (13), (14) and (18), (19), one can compute estimates $\hat{f}$ and $\hat{s}$ using any of many methods which require a constant number of operations per data point including direct methods for solving the associated normal equations and Kalman filter smoothing. For the simulation results that follow, computations were made by a multiscale recursive estimation algorithm [15], [12] (see also Section VI). An implementation detail concerning this and other algorithms is that they require the specification of prior variances P_0 on the first samples of the piecewise smooth function f and edge function s. However, the precise interpretation of the variational formulation as an estimation problem corresponds to viewing the initial value as unknown, which is equivalent to an infinite prior variance. While it is possible to accommodate this into the estimation formulation with no effect on algorithmic complexity,

Parameter	*Description*	*Value for the Results in Figures 2 and 5*	*Value for the Results in Figure 4*
λ	λ adjusts the smoothness in f away from edges. See (18) and (19).	1	25000
b	b affects the edginess of the edge estimate. See (18) and (19)	10	25
c	c adjusts the allowed variability in s. See (19).	100	1
r	r is the assumed noise variance in the data	1	0.2^2
P_0	P_0 is the prior covariance for initial process values.	100	100
ϵ	Estimates of s are clipped to lie within $[0, 1-\epsilon]$.	1.0×10^{-4}	1.0×10^{-4}
Δ	The algorithms stops after the percent change of the functional (9) falls below Δ.	1%	0.01%

TABLE I

DESCRIPTION OF PARAMETERS IN 1-D SEGMENTATION ALGORITHM

it is common to use an alternate approach in which one closely approximates the solution to the original problem by setting the prior covariance P_0 to a relatively large number.

The multiscale estimation algorithm is used to calculate the estimates and error variances in the 1-D segmentation algorithm diagrammed in Figure 1. Since estimating f requires that $\frac{1}{(1-s)^2}$ be well-behaved, we must also enforce a constraint on the range of s. A simple solution that proves adequate is to clip each estimate of the edge function so that for some small ϵ, $s \in [0, 1-\epsilon]$. The iterative algorithm is now completely specified except for how to start and when to stop. For all of the examples in this section, the algorithm starts by estimating f^1 using an initial edge estimate $s^0 = 0$, and the algorithm stops when the percent change of the functional (9) falls below some threshold parameter Δ. A list of all parameters input to the segmentation algorithm appears in Table I. To illustrate the operation of the algorithm, some typical examples follow. These, in turn, are followed by some Monte Carlo experiments designed to assess quantitatively the performance of the algorithm.

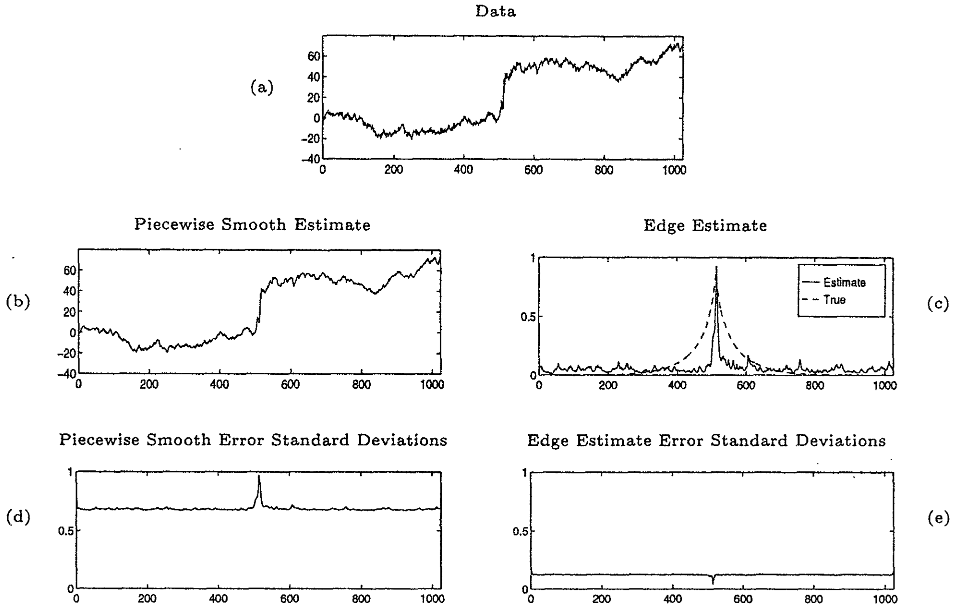


Fig. 2. The results for segmenting the data pictured in (a), a noisy observation of a process whose statistics are dictated by (13), (14) for the true edge function pictured in part (c). The measurement noise is white and Gaussian with unit variance. All parameters are set as in Table I.

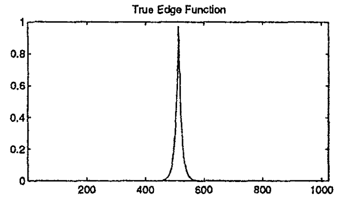


Fig. 3. The edge function used to generate the realizations for the numerical example of Figure 2 and the Monte Carlo runs of Figure 5.

A. Typical Examples

Figure 2 illustrates a segmentation for a synthetic example, using the parameters in Table I. The data g in Figure 2a consists of a signal f to which unit intensity white Gaussian measurement noise has been added. The signal f is a realization of a Gaussian process described by (14) starting with initial condition $f_0 = 0$ and with edge function s given by the exponential in Figure 3. Now, recall that where the edge function is approximately one, the variance of the increment in the model of f increases. This is clearly evident in Figure 2a in which the particular realization of f displays a clear jump in its value in the vicinity of the edge function's peak.

The data in Figure 2a are then fed into the iterative algorithm of Figure 1. The results

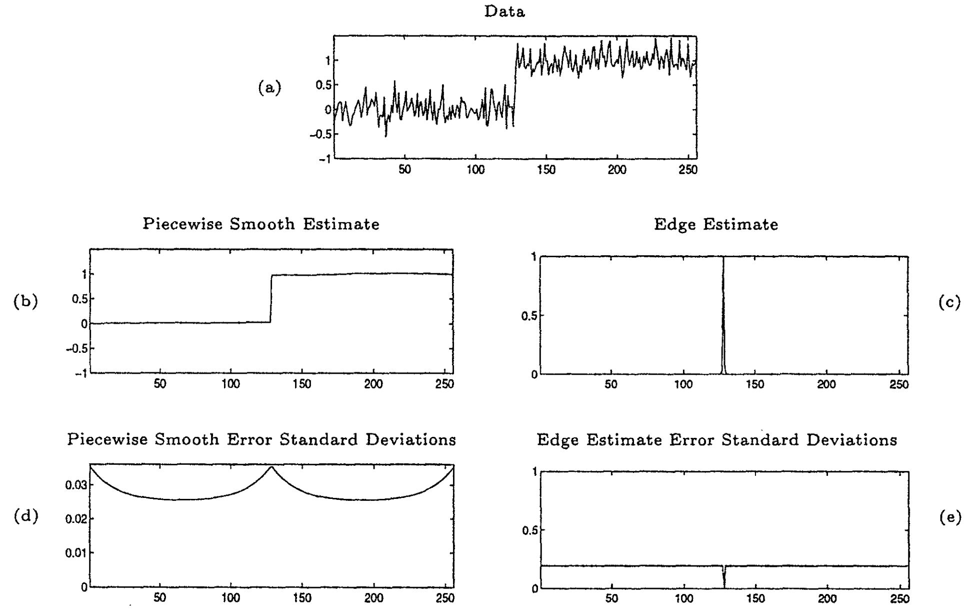


Fig. 4. The results for segmenting a noisy observation of a prototypical unit step edge as plotted in part (a). The measurement noise is white and Gaussian with standard deviation 0.2. All parameters are set as in Table I.

displayed in the remaining parts of Figure 2 are after three iterations of the algorithm, at which point, the values of the functional (9) were changing by less than $\Delta = 1\%$. No clipping was necessary during the course of the run, and thus, the results are true to the discrete form of the variational formulation (9). The final estimates yield a good segmentation. The piecewise smooth estimate is a smoother version of the data, but the edge has not been smoothed away, and the edge estimate has a strong peak at the location of the edge. In addition, the estimation error variance for f in Figure 2d displays the characteristic one would expect: away from the expected edge, considerable lowpass filtering is effected, reducing the noise variance. However, in the vicinity of the edge, one expects greater variability and, in essence, the estimator performs less noise filtering, resulting in a larger error variance. Note also that the variance in the estimate of the edge process is almost constant, with a slight drop in the vicinity of the edge, *i.e.*, where f changes abruptly, reflecting greater confidence that an edge is present in this vicinity.

The results for the preceding example are good, but not completely convincing by themselves since f is matched to the algorithm by its construction. Consider a prototypical signal not matched to the model, namely a step edge. Figure 4 displays results for a noisy observation of a unit step. The estimates are shown after 12 iterations. Once again, no clipping was necessary in

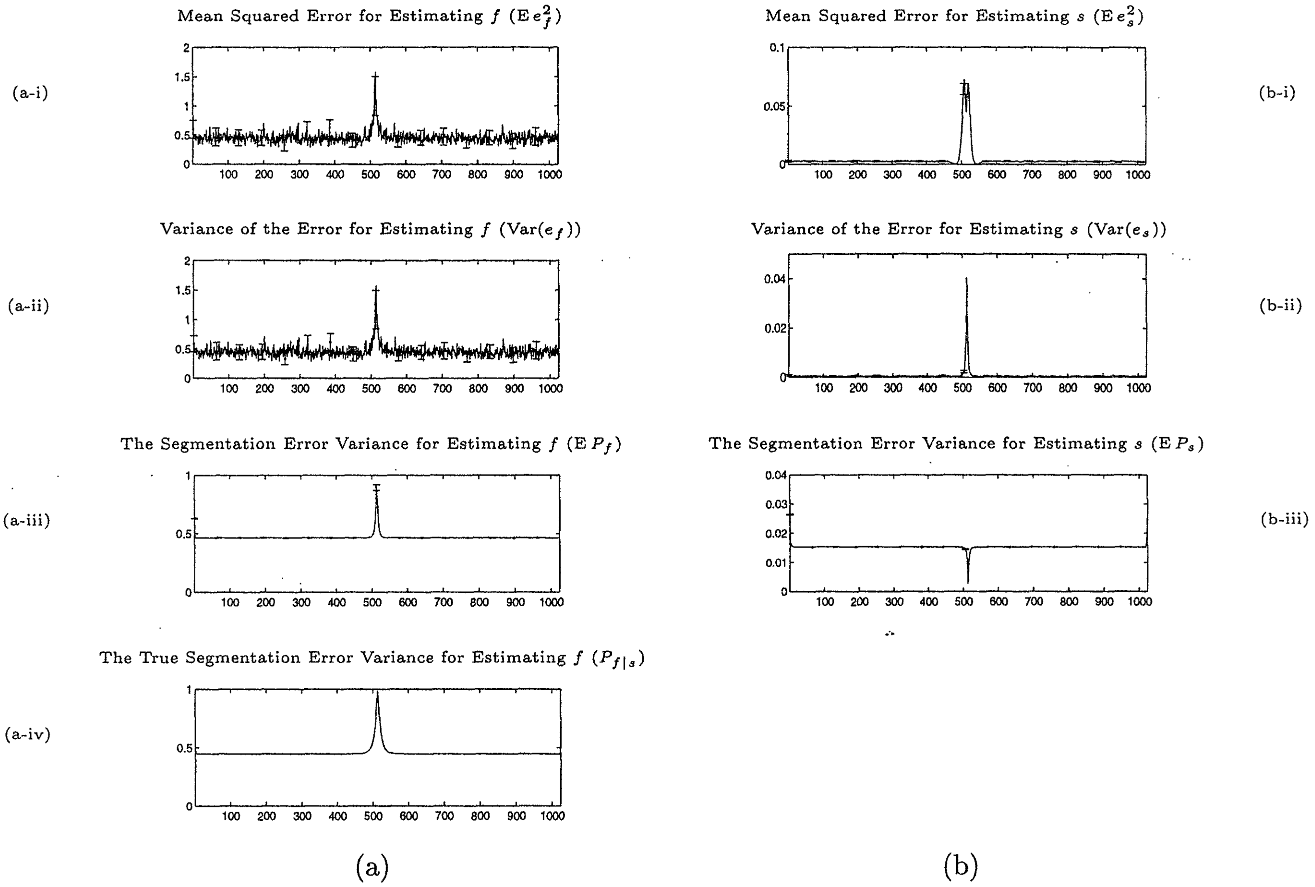


Fig. 5. A comparison of various error statistics compiled using Monte Carlo techniques for segmenting synthetic data. The data are realizations of a process whose statistics are given by (13), (14) for the exponential edge function plotted in Figure 3. Part (a) of this figure displays statistics concerning the piecewise smooth estimate errors $e_f = (\hat{f} - f)$ and the piecewise smooth estimate error standard deviations generated by the algorithm, P_f. Part (a-iv) displays the optimal error standard deviations for estimating f given that the true edge function in Figure 3 is known. Part (b) of this figure displays the statistics concerning the edge estimate errors $e_s = (\hat{s} - s)$.

the iterative process. The results demonstrate that the algorithm works as desired. It removes almost all of the noise away from the edge, while preserving the discontinuity accurately. As in the case of the first example, the error statistics reflect the fact that near the edges one expects less noise reduction in estimating f and has higher confidence in the estimate of the edge process because of the abrupt change in value of f.

B. Monte Carlo Experiments

In this section, a more careful look is taken at the error statistics provided by the segmentation algorithm in order to assess their accuracy and utility. Since the full iterative algorithm is nonlinear, the exact error variances in estimating f and s are not easily computed, and the statistics calculated by our algorithm represent approximations that result from the linear estimation problems for each of the two separate coordinate descent steps for f and s. Figure 5 presents Monte Carlo results comparing the error statistics computed by the segmentation algorithm with the actual error variances. Each experiment in this simulation corresponds to (a) generating a realization f of the process described by (13), (14) for the fixed edge function s appearing in Figure 3 and with the initial point f_0 set to 0; (b) adding white Gaussian measurement noise with unit intensity; and (c) applying the segmentation algorithm using the parameters in Table I to obtain the estimates $\hat{f}$ of the realization f and $\hat{s}$ of the edge function s as well as P_f and P_s, the error variances for these estimates that the algorithm generates.

The quantities of interest for each run are $e_f = (\hat{f} - f)$, $e_s = (\hat{s} - s)$, and the error statistics P_f and P_s computed by the algorithm. From 100 independent runs, the following quantities are estimated: $\mathrm{E}\, e_f^2$, $\mathrm{Var}(e_f)$, $\mathrm{E}\, P_f$, $\mathrm{E}\, e_s^2$, $\mathrm{Var}(e_s)$, and $\mathrm{E}\, P_s$, and these are plotted in Figure 5 along with Monte Carlo error bars set at 2 standard deviations. Comparing Figure 5a-i for $\mathrm{E}\, e_f^2$ and Figure 5a-ii for $\mathrm{Var}(e_f)$, one sees that these are quite close in value, indicating that the estimate produced by our algorithm is essentially unbiased. Comparing these two figures with the plot of $\mathrm{E}\, P_f$, one observes that the error variance computed by our algorithm has essentially the same shape, reflecting that it accurately captures the nature of the errors in estimating f. Figure 5a-iv shows a plot of the error variance for an estimator that is given perfect knowledge of the edge process. Comparing this to Figure 5a-iii, one notices that the segmentation algorithm performs nearly as well as if s were known perfectly and did not have to be estimated.

The error statistics for the edge function are depicted in Figure 5b. $\mathrm{E}\, e_s^2$ and $\mathrm{Var}(e_s)$ being small relative to one indicate that the estimate of the edge function is quite accurate and that the error does not vary very much from sample path to sample path. In addition, the shapes of these plots have several interesting features related to the behavior of the estimator in the vicinity of the edge. Note first that, as can be seen in Figure 2, the algorithm tends to estimate edge functions that are more narrow than the actual edge function. This is actually preferable for segmentation, for which the peak locations in the edge estimates are more important than the estimates' shapes. Because of this bias toward tighter edge localization, $\hat{s}$ is a slightly biased estimate of s in Figure 3, as evidenced by the broader peak of $\mathrm{E}\, e_s^2$ as compared to $\mathrm{Var}(e_s)$.

A second interesting point is that $\mathrm{E}\, e_s^2$ increases slightly in the vicinity of the edge, while the variance computed by the estimation algorithm, $\mathrm{E}\, P_s$, decreases. The reason for this is not difficult to understand. Specifically, the estimator believes that it is has more information about s when the gradient of f is large, and thus, in the vicinity of an edge, it indicates a reduction in error variance for estimating s. However, if the estimate of the *location* of the edge is at all in error, then the difference $e_s = (\hat{s} - s)$ will exhibit very localized but large errors, both positive and negative (just as one would see in the difference of two discrete-time impulses whose locations are slightly different). Thus, rather than providing as accurate an estimate of the size of the estimation error variance in this vicinity, this dip in the error variance should be viewed as a measure of confidence in the presence of an edge in the vicinity.

V. Two-Dimensional Problem

The 2-D problem poses some difficulties that are not present in the 1-D case. This section presents a 2-D statistical interpretation of Shah's segmentation formulation and, then, a discussion of some of the inherent computational difficulties in the 2-D problem. One possible approach for overcoming these difficulties is then developed in Section VI.

The first step in setting up the 2-D problem is to establish a discrete form of (2). To do this, let g_{ij}, f_{ij}, and s_{ij} be samples of g, f, and s respectively on an $n \times n$ rectangular grid, and to approximate the gradient in (2) by a first difference scheme involving nearest neighbors on the grid. The coordinate descent subproblems in 2-D can be written in the same form as the 1-D subproblems if one assembles the samples in a lexicographic ordering into column vectors g, f, and s, and introduces a 2-D difference operator $\mathcal{L}$, weighting matrices $\mathcal{S}$ (a function of s) and A (a function of f), and observation vector γ (a function of f), analogous to those in 1-D. Details are discussed in Appendix A. The problem of finding the optimal piecewise smooth estimate $\hat{f}$ to the data g for a given fixed s is that of finding f that minimizes

$$E_s(f) = ||f - g||^2_{r^{-1}I} + \lambda||\mathcal{L}f||^2_{\mathcal{S}^T\mathcal{S}}, \tag{20}$$

which is equivalent, when $\mathcal{S}$ is invertible, to finding the least-squares estimate of f given the measurement and model equations

$$\begin{aligned} g &= f + \sqrt{r} v^f && (21) \\ \mathcal{L} f &= \mathcal{S}^{-1} w^f && (22) \end{aligned}$$

where v^f and w^f are Gaussian random vectors with identity covariance. Likewise, the problem of finding the optimal edge estimate $\hat{s}$ for f fixed consists of finding the s that minimizes

$$E_f(s) = ||\gamma - s||^2_{A^T A} + c||\mathcal{L} s||^2, \quad (23)$$

or, equivalently, to finding the least-squares estimate of s based on the model

$$\begin{aligned} A\gamma &= As + v^s && (24) \\ \mathcal{L} s &= w^s && (25) \end{aligned}$$

where v^s and w^s are Gaussian random vectors with identity covariance.

While the 2-D problem has the same apparent structure as in 1-D, computing estimates and associated error variances in 2-D is not an easy task. For example, finding the minimizer $\hat{f}$ of (20) and the associated error variances involves solving

$$(r^{-1} I + \mathcal{L}^T \mathcal{S}^T \mathcal{S} \mathcal{L}) \hat{f} = g \quad (26)$$

and finding the diagonal elements of $(r^{-1} I + \mathcal{L}^T \mathcal{S}^T \mathcal{S} \mathcal{L})^{-1}$. Finding the minimizer $\hat{s}$ of (23) and associated error variances involves analogous computation. As discussed in [9], these calculations correspond to solving estimation problems with prior models that are 2-D Markov Random Fields or, equivalently, to solving discretized elliptic partial differential equations and computing the diagonal elements of the inverses of elliptic operators. There exist no known algorithms which can compute the necessary quantities for this general problem with fewer than $O(n^3)$ operations.

Since the objective is to generate estimates and error variances with constant computational complexity per pixel, (*i.e.* with $O(n^2)$ operations), one is confronted with the need to develop approximations. In [8], one such approach is described. It involves an approximation to the solutions of (26) and the analogous equation for s based on so-called marching methods. This paper focuses on a different approach which involves changing the problem. In particular, a modification is made in the prior model, as specified by the operator $\mathcal{L}$ in (26). The new prior model is chosen from a class which allows the use of a multiscale recursive estimation

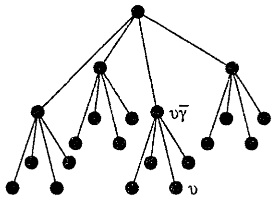


Fig. 6. In the notation of this paper, if ν is the index of some node, $\nu\bar{\gamma}$ denotes the parent of that node.

algorithm [16] to compute exact estimates and error variances. This approach is developed in detail in Section VI. Since it is based on a change in the prior model, the precise tie to Shah's formulation of the segmentation algorithm is lost; however, as will be seen, this method yields good segmentations and meaningful error statistics.

VI. A Multiscale Method for Segmentation

A. Multiscale Modeling Framework

The multiscale framework used in this paper, which was introduced in [16] and further developed in [9], [10], [11], [17], models an image as the finest scale of a stochastic process indexed by nodes on a quad-tree (see Figure 6). An abstract index ν is used to denote a node on the tree, and the notation $\nu\bar{\gamma}$ is used to refer to the parent of node ν. Processes on such a tree of interest for segmentation are specified in terms of a root-to-leaf recursion of the form

$$x_\nu = A_\nu x_{\nu\bar{\gamma}} + B_\nu w_\nu \tag{27}$$

where the w_ν and the state x_{root} at the root node are a collection of independent zero-mean Gaussian random variables, the w's with identity covariance and x_{root} with prior covariance P_{root}. The A and B matrices are deterministic quantities which define the statistics of the process on the tree. Observations g_ν of the state variables have the form

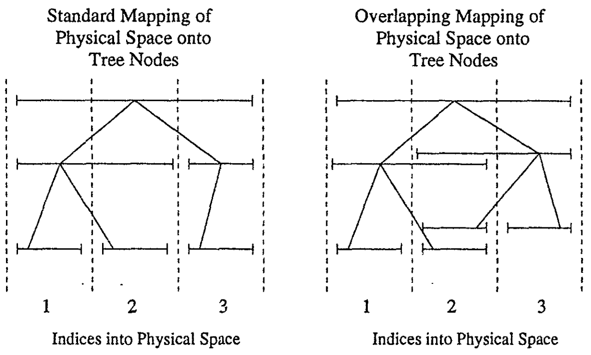
$$g_\nu = C_\nu x_\nu + v_\nu \tag{28}$$

where the v_ν are independent and Gaussian, and the matrices C_ν are deterministic. The least-squares estimates of process values at all nodes on the tree given all observations and the associated error variances can be calculated with an efficient recursive algorithm [15], [16] which requires only $O(n^2)$ operations when there are n^2 finest scale nodes. The algorithm consists of a fine-to-coarse recursion in which data in successively larger subtrees are fused up to the root node of the tree, and a subsequent coarse-to-fine recursion which produces both the optimal estimates and their error covariances. By modeling an image as the finest scale of a multiscale process, one can, thus, compute optimal estimates and error variances for an $n \times n$ image with significantly less than the $O(n^3)$ operations required by the most efficient methods when the image is modeled as a Markov Random Field.

The approach taken here to constructing multiscale models for each of the two coordinate descent steps of the multiscale segmentation algorithm build on several earlier efforts. In [9], it was demonstrated that multiscale models could be used as alternatives to so-called smoothness priors, such as penalties on the norm of the gradient of the field to be reconstructed. The work in [9], however, suffered from an artifact that may or may not be a significant concern depending upon the application, but certainly is for the segmentation problem. In particular, the multiscale models used in [9] led to reconstructions which possessed blocky artifacts, which look very much like edges. A modification of this modeling methodology developed in [18], however, removes these artifacts and, thus, provides a framework appropriate for application to the segmentation problem. The description of this methodology provided here is limited to a brief overview of the key concepts and constructs; the reader is referred to [18] for details.

The source of the blocky artifacts can be traced to the fact that some points on the finest scale of the tree in Figure 6, while close to each other in physical space, are separated by a larger distance on the tree, resulting in an apparent loss of correlation between what should be highly correlated neighboring pixels. From Figure 6, one observes that this happens only for certain finest scale nodes and not at others. So, if one associates each finest scale node with a distinct pixel in an image, there is considerable irregularity in the correlation among neighboring pixels as one moves across the image. The idea in [18] is to introduce redundancy in the mapping between finest scale values of a multiscale process and the corresponding image so that multiple fine scale nodes correspond to each individual pixel. In so doing, the effective distance between neighboring pixels (*e.g.*, as measured by the average of the tree distances between nodes corresponding to one pixel and nodes corresponding to the other) can be made much more uniform, lessening blocky artifacts dramatically. The resulting models are referred to as *overlapping models*, to reflect the fact that nodes on the tree have overlapping domains of influence in the image domain.

Consider the example diagrammed in Figure 7. Here there are three data points in physical 1-D space. Figure 7a diagrams a standard mapping between tree nodes and data points. Each finest scale node is associated with one data point. Figure 7b depicts an overlapping tree model



(a) (b)

Fig. 7. A standard and overlapping mapping of physical space onto tree nodes

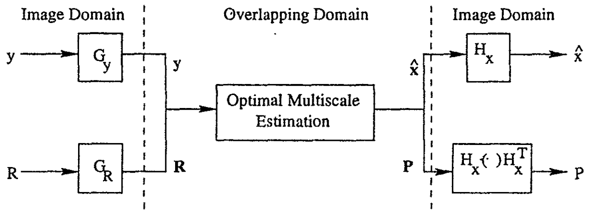


Fig. 8. The computational structure of the overlapping multiscale estimation framework.

structure. Note that, while physical point 2 is equally close to nodes 1 and 3, the tree distance in Figure 7a between the tree nodes associated with points 1 and 2 is shorter than that between the nodes associated with 2 and 3. In contrast, each node in Figure 7b has an overlapping interval of influence (as indicated by the intervals associated with each node in the figure), and, as a result, there are two finest scale nodes associated with the point 2, one of which is closer to the node corresponding to 1 and one of which is closer to the node associated with 3, resulting in the same average distance between node 2 and each of nodes 1 and 3.

The details and use of models of this type for the computation of estimates and error variances is depicted in Figure 8. In the center of this diagram is the optimal multiscale estimation algorithm. The operations on either side of the multiscale estimator connect this overlapped-domain operation to the real image domain. First, the measurements g_ν, defined in the image domain, must be *lifted* to measurements in the overlapped domain. Since there are multiple nodes in the overlapped domain that may correspond to a single point in the image domain, one must specify a method for distributing the measurements over these redundant nodes. The method used is simply to replicate the measurement value of any image domain pixel at each of the finest scale tree nodes that correspond to this pixel. The multiscale estimator, then, treats these as independent measurements. This is obviously not true and appears to suggest that the estimator assumes that it has more information available than is actually the case. However, this can be balanced by, in effect, telling the multiscale estimator that each of the replicated measurements is noisier than it actually is. Specifically, the measurement variance is increased proportionally to the number of redundant nodes on the tree onto which the measurement value is mapped. These two operations, replicating the image domain measurements at each tree node corresponding to that pixel and increasing the corresponding measurement noise variances, are represented in Figure 8 by the operators G_g and G_R respectively.

Finally, once the estimate $\hat{\mathbf{x}}$ and its error covariance $\mathbf{P}$ have been computed in the overlapped domain, one must map back into the image domain. This mapping produces an estimate at each pixel which is a weighted average of the estimates at each of the nodes on the tree corresponding to that pixel. This projection is captured by the operator H_x in Figure 8. Since $\hat{x} = H_x \hat{\mathbf{x}}$, it follows that the corresponding error covariance for $\hat{x}$ is $H_x \mathbf{P} H_x^T$. As described in [18], the operators G_x and H_x are specified completely in terms of the overlapped structure of the tree, namely how many pixels of overlap are captured at each scale of the tree. For example in Figure 7b, there is one pixel of overlap at the first scale down from the root node and no overlap at the finest scale. In general, the overlap structure of a model is specified by a vector $\mathcal{O} = \begin{pmatrix} o_1 & \ldots & o_m \end{pmatrix}$, where $o_1, \ldots, o_m$ are the number of pixels of overlap at scales $1, \ldots m$, where m is the finest scale [10], [17]. Once $\mathcal{O}$ is specified, the operator G_g is completely determined, as all this operator does is identify which redundant tree nodes correspond to which image domain pixels. Similarly, the operator G_R is also specified, since all this operator does is to amplify the measurement noise on each replicated measurement by a factor equal to the number of tree nodes corresponding to each pixel. There is still, however, some flexibility in the choice of the operator H_x, since any weighted average of the nodes corresponding to an individual image pixel can, in principle be used. One natural choice [18] which is used in this paper is to choose the weights in a way that tapers the influence of nodes that are at the ends of the intervals of overlap. For example, in Figure 7b, the natural weights on the two nodes corresponding to data point 2 are each 1/2 because of the symmetry of the overlap. More details along with a more intricate example are given in Appendix B. The complete details of the overlapping framework are presented in [18], together with a proof that the overall system depicted in Figure 8 does indeed produce the optimal estimates and error statistics in the image domain and with a demonstration that the resulting system maintains the $O(n^2)$ complexity that is desired.

B. Multiscale Models for Segmentation

The approach taken to incorporate the multiscale framework into a segmentation algorithm is to devise and use algorithms as in Figure 8 to replace each of the two main computational tasks in the coordinate descent segmentation scheme, namely, the estimation of s with f held fixed and the estimation of f with s held fixed. As seen in (20) and (23), smoothness is imposed in each of these steps by the first differencing scheme embodied in the operator $\mathcal{L}$. What follows is a discussion of the use of multiscale models as alternatives to those specified by the operator $\mathcal{L}$.

Consider first, the model used for the estimation of s. As discussed in [9] - [12], [19], the smoothness penalty associated with the gradient operator $\mathcal{L}$ corresponds to a fractal penalty, roughly equivalent to a $1/f$-like prior spectrum for the random field being modeled. Such a spectrum has a natural scaling law, namely the variances of increments at finer and finer scales decrease geometrically. In [9] - [12], it was demonstrated that a very simple multiscale model having this same scaling property leads to very similar estimates to those produced using the original smoothness penalty. It is precisely a model of this type that is used for the lifted version s of the edge process. Specifically,

$$\mathbf{s}_\nu = \mathbf{s}_{\nu\bar{\gamma}} + d_\nu B^s w_\nu^s \tag{29}$$

where B^s is a constant, and the w_ν^s are independent unit variance Gaussian random variables. The d_ν terms are constants which decrease geometrically with scale. In particular, as described in [10], [17], the decrease in the variance of the noise from one scale to the next finer scale is proportional to the ratio of the lengths of the intervals associated with nodes at each of the two scales, which, in turn, is related to the amount of overlap used in the models. More precisely, if ω_m denotes the length of the interval associated with each node at scale m, then

$$d_\nu = d_{\nu\bar{\gamma}}\sqrt{\frac{\omega_{m(\nu)}}{\omega_{m(\nu\bar{\gamma})}}} \qquad d_{\text{sub-root}} = 1 - \frac{\omega_{\text{root}}}{\omega_{\text{sub-root}}} \tag{30}$$

$$\omega_m = \frac{\omega_{m-1} + o_m}{2} \qquad \omega_{\text{root}} = N \tag{31}$$

where $m(\nu)$ denotes the scale of node ν, the indices *sub-root* and *root* denote the node at the root and any node one level below the root respectively, and N is the linear dimension of the image. The measurements and measurement error variances used in conjunction with the model for s in (29) are given by the elements of γ and A in (40) and (41).

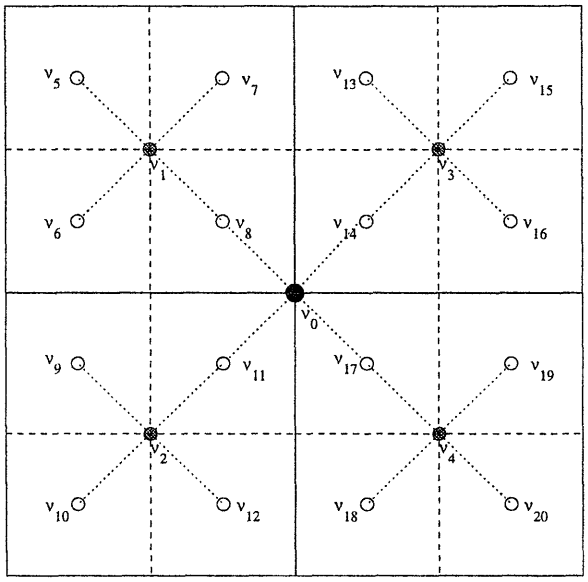


Fig. 9. How discontinuities are incorporated into the $1/f$-like multiscale model.

The multiscale model for the lifted piecewise smooth process $\mathbf{f}$ is, to some extent, similar to the one for the edge process. However, significant modification to this model is needed in order to capture the presence of discontinuities, as indicated by the edge estimates. In particular, in the 1-D case, as captured in (14), the increments of f have a variance which is inversely proportionally to the corresponding value of $(1-s)^2$. Thus, near an edge, *i.e.* where s is approximately one in value, the variance of the increment of f is large. In a similar manner, one needs to capture the idea that increments of f, as one moves to finer scales, should have variances that reflect the presence of edges, *i.e.*, that are again inversely proportional to $(1-s)^2$. This is done as follows. Note that each node on the tree can be thought of as representing the center of a subregion of the image domain. For example, the left node at the middle scale in Figure 7b can be thought of as corresponding to the pair of data points {1,2} and thus is centered at 1.5. A more complicated 2-D example is depicted in Figure 9. The dots in this figure correspond to the center points of the regions associated with different nodes on the tree. The dots are shaded according to the scale of the corresponding node on the tree; the darker the dot, the coarser the scale. Thus, for example, the node ν_0 represents the entire large square region, while the node ν_3 at the next finest scale represents the upper-right quadrant of this large square. Now, if there is an edge located between ν_0 and ν_3, *i.e.* if the values of s at image domain pixels between these nodes indicate the presence of an edge, the variance of the scale-to-scale increment of $\mathbf{f}$ between these

Parameter	*Description*	*Value for Results in Figure 10*	*Value for Results in Figure 12*
b	b affects the edginess of the edge estimate. See(41) and (40).	10	10
λ	λ affects the determination of what is an edge and what isn't. See (41) and (40).	50	500
B^s	B^s adjusts the multiscale smoothness penalty placed on s. See (29).	0.87	1/50
B^f	B^f adjusts the multiscale smoothness penalty placed on f. See (32).	0.75	1/50
r	r is the assumed noise variance in image data.	1	1
P_{root}	P_{root} is the multiscale model prior covariance for the process value at the root node.	1×10^6	1×10^6
$\mathcal{O}$	Each component of $\mathcal{O}$ specifies the amount of overlap at a particular scale.	(50 31 18 11 $\cdots$ 7 4 2 1 0 0)	(16 10 7 4 $\cdots$ 2 2 0 0)
ϵ	Estimates of s are clipped to lie within $[0, 1-\epsilon]$.	0.01	0.01
I	I is the number of iterations of estimating f and s.	2	2

TABLE II

DESCRIPTION AND VALUES OF PARAMETERS IN THE MULTISCALE METHOD.

two nodes should increase. More precisely, the model for $\mathbf{f}$ is specified by the recursion

$$\mathbf{f}_\nu = \mathbf{f}_{\nu\bar{\gamma}} + \eta_\nu d_\nu B^f w_\nu^f \qquad (32)$$

where B^f is a constant, w_ν^f are independent unit variance Gaussian random variables, and η_ν is the sum of the edge estimates $1/(1-\hat{s}_{ij})^2$ which fall on the line connecting ν and $\nu\bar{\gamma}$. In this manner, additional uncertainty is put into the recursion for $\mathbf{f}$ at the appropriate locations.

C. Numerical Results

This section presents numerical results on two test images, an MRI brain scan and AVHRR imagery of the Gulf Stream. A listing of the algorithm's parameters, their function, and their values used in the following results is given in Table II.

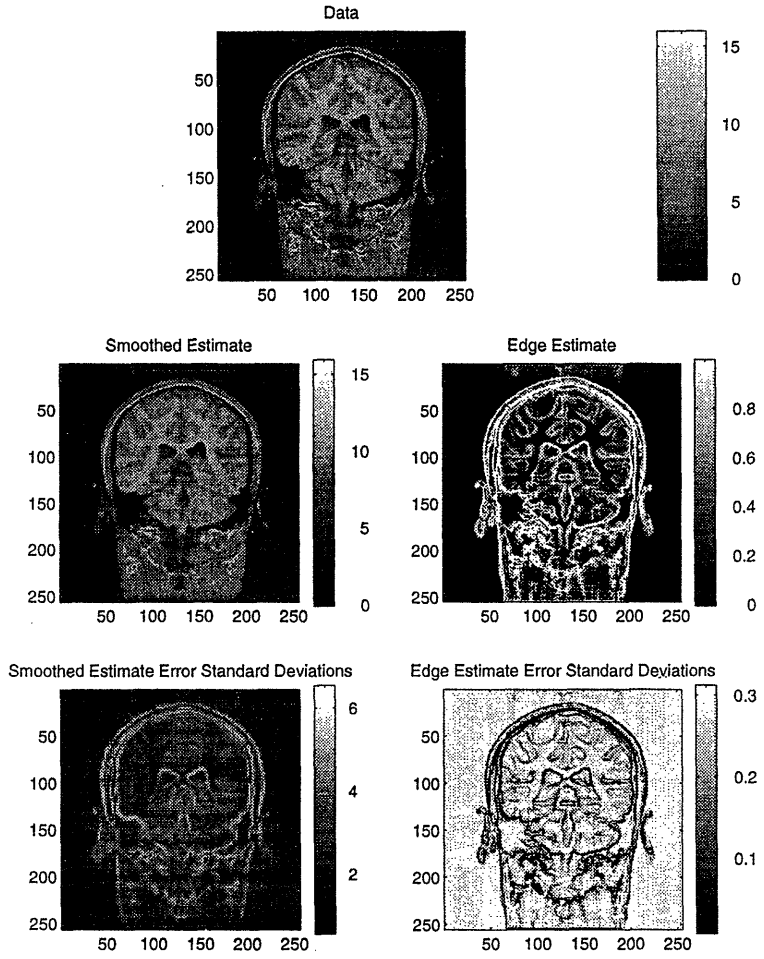


Fig. 10. MRI segmentation computed using the multiscale method.

Figure 10 displays a multiscale segmentation of an MRI brain scan. The goals in segmenting such imagery typically include demarcating the boundaries of the ventricles, the two hollow regions in the middle of the brain, and the boundaries between gray and white matter in the brain. The edge estimate displayed in Figure 10 does a good job at this. In addition to the estimates, the multiscale algorithm computes the error standard deviations. Notice that the error standard deviations for the smoothed estimate increase near edges and, for the edge estimate, decrease near edges, as in the 1-D results. Thus, one expects that that the error standard deviations generated by the multiscale segmentation algorithm are of similar significance to those generated in 1-D. Another consequence of the above-mentioned properties of the error standard deviations is that they can be used not only to estimate one's confidence in segmenting the image but also to improve one's estimate of the boundary locations since the error standard deviations mark

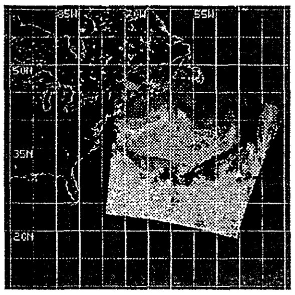


Fig. 11. AVHRR data of the North Atlantic on June 2, 1995 at 5:45:01 GMT.

the edges in the image as well or better than the edge estimate.

The multiscale segmentation algorithm has also been tested on Advanced Very High Resolution Radar (AVHRR) imagery of the North Atlantic Ocean.[1] The gray scale AVHRR imagery portrays water temperature. Lighter tones correspond to warmer water, and darker tones to cooler water. A sample image with a coastline overlay is displayed in Figure 11. Data exists only for the lightly colored square-like region. One observes a thin northeast running warmer body of water off the coast of the Carolinas. This is the Gulf Stream. The position of the Gulf Stream is important to the shipping industry and oceanographers, among others. The features that are important for a segmentation algorithm to pick out are the north and south walls of the Gulf Stream and the occasional eddies that break off to the north and south of the Gulf Stream called warm and cold core rings respectively. Performing the segmentation is difficult because there are effective measurement dropouts due to the presence of clouds. The effect of clouds is to depress the temperature from its actual value. In particular, one can observe bands of black running through the image in Figure 11. These correspond to measured temperatures of zero degrees Celsius or lower. Such measurements are considered so poor that they are simply designated points where there exists no measurement.

Figure 12 shows the results for using the multiscale segmentation algorithm to segment AVHRR imagery. Notice that the edge estimate $\hat{s}$ highlights well the north and south walls of the Gulf Stream and some of the boundaries of the eddies, as desired. In the image of the smoothed estimate of the temperature, the boundaries of the Gulf Stream and the eddies have been preserved, and the algorithm has interpolated at locations where no data point existed. The multiscale method not only computes these estimates, it also computes the associated error standard devia-

[1] The AVHRR image in this paper was obtained from an online database maintained by The University of Rhode Island Graduate School of Oceanography.

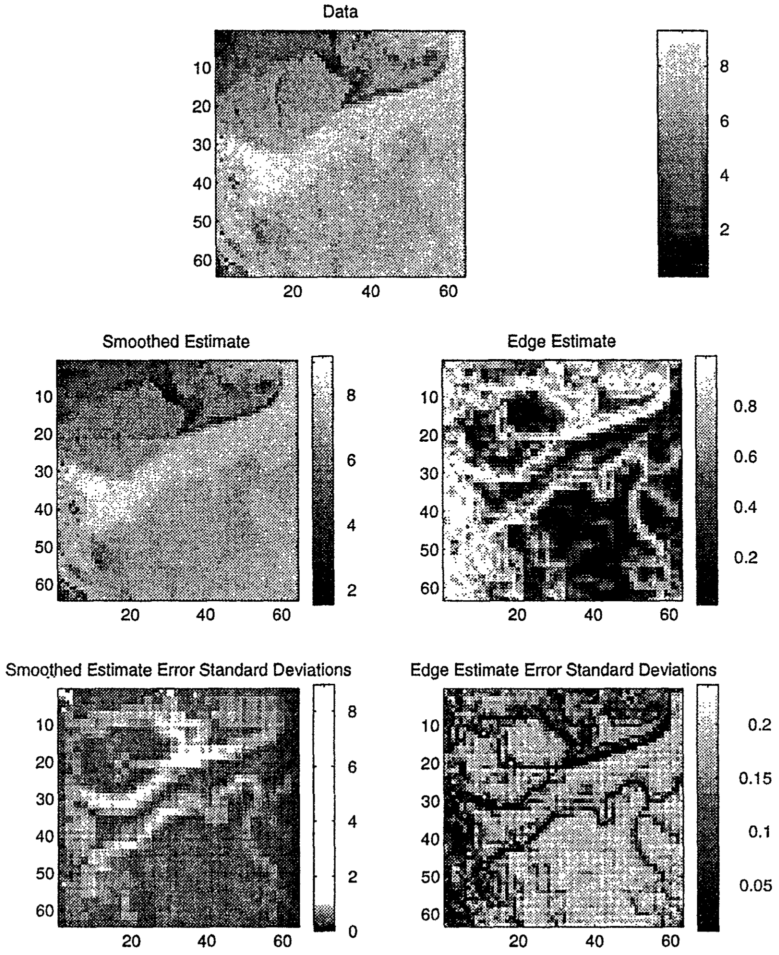


Fig. 12. AVHRR segmentation computed using the multiscale method.

tions, which are also displayed in Figure 12. The piecewise smooth error standard deviations are very large at certain points. These locations correspond to regions where no data point existed and interpolation was necessary. Away from the data dropouts, the piecewise smooth estimate error standard deviations are larger near edges, and the edge estimate error standard deviations are smaller near edges, just as in the MRI image. Thus, as in the case of the MRI image, the error standard deviations in the edge estimates delineate the boundaries between regions as clearly as the edge estimates themselves, providing high-confidence localization of prominent boundaries.

VII. Conclusion

Motivated by segmentation problems arising in remote sensing, medical imaging, and other scientific imaging areas, this paper presents a signal and image segmentation algorithm which is both computationally efficient and capable of generating error statistics. The starting point for this work is a variational approach to segmentation [1]. This approach involves using coordinate descent to minimize a non-linear functional. Each of the coordinate descent sub-problems is a convex quadratic minimization problem amenable to a precise statistical interpretation. In a one dimensional setting, this papers explores the resulting statistical approach to segmentation in great depth. Monte Carlo experiments indicate that the estimates and error statistics are accurate and meaningful quantities.

The situation in two dimensions is complicated by the fact that the straightforward extension of the work in one dimension will yield an algorithm whose computational complexity per pixel grows significantly with image size. In order to address these computational concerns, the use of multiscale methods for computing piecewise smooth and edge estimates and error variances is investigated. These methods involve changing the prior model appearing in the statistical interpretation of the variational problem to one which is appropriate in the context of segmentation but also one for which the resulting estimation problems are relatively easy to solve. This leads to the development of a multiscale segmentation algorithm which yields good results when applied to both an MRI brain scan and AVHRR imagery of the Gulf Stream.

Appendices

I. Details in the Derivation of the Two-Dimensional Estimation Problem

This appendix defines many of the quantities referred to in Section V. First, the discretization of (2) which results from using samples g_{ij}, f_{ij}, and s_{ij} of g, f, and s on a $n \times n$ rectangular grid and approximating the gradient with a first difference scheme is the functional

$$\begin{aligned} E(f,s) = {} & \sum_{i=1}^{n}\sum_{j=1}^{n}(g_{ij} - f_{ij})^2 + \lambda \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}(1 - s_{ij})^2((f_{i(j+1)} - f_{ij})^2 + (f_{(i+1)j} - f_{ij})^2) + \\ & \lambda \sum_{i=1}^{n-1}(1 - s_{in})^2(f_{(i+1)n} - f_{in})^2 + \lambda \sum_{j=1}^{n-1}(1 - s_{nj})^2(f_{n(j+1)} - f_{nj})^2 + \\ & \frac{\beta}{2}(\rho \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}((s_{i(j+1)} - s_{ij})^2 + (s_{(i+1)j} - s_{ij})^2) + \rho \sum_{i=1}^{n-1}(s_{(i+1)n} - s_{in})^2 + \\ & \rho \sum_{j=1}^{n-1}(s_{n(j=1)} - s_{nj})^2 + \frac{1}{\rho}\sum_{i=1}^{n}\sum_{j=1}^{n}(s_{ij}^2 - \delta_{in}\delta_{nj}s_{ij}^2)). \end{aligned} \quad (33)$$

As noted in Section V, one can use coordinate descent to minimize (33), and each of the subproblems will have the simple structure that was noted in one dimension. The subproblems are written out in Section V using quantities defined here.

If one assembles the samples in a lexicographic ordering into column vectors g, f, and s, one can then define a $2n(n-1) \times n^2$ matrix $\mathcal{L}$ analogous to the operator in (2). First, define the row difference operator $\mathcal{L}_r$, composed of n blocks of L_n, in (11):

$$\mathcal{L}_r = \begin{pmatrix} L_n & & & \\ & L_n & & \\ & & \ddots & \\ & & & L_n \end{pmatrix}. \tag{34}$$

Next, one can compute the first differences of all columns of the image by operating with $\mathcal{L}_c$:

$$\mathcal{L}_c = \begin{pmatrix} I_{n\times n} & -I_{n\times n} & & & & \\ & I_{n\times n} & -I_{n\times n} & & & \\ & & I_{n\times n} & -I_{n\times n} & & \\ & & & \ddots & \ddots & \\ & & & & I_{n\times n} & -I_{n\times n} \end{pmatrix}. \tag{35}$$

Finally, all first differences can be taken by operating with $\mathcal{L}$:

$$\mathcal{L} = \begin{pmatrix} \mathcal{L}_r \\ \mathcal{L}_c \end{pmatrix}. \tag{36}$$

Each of the optimization subproblems in the coordinate descent scheme involves a weighting matrix. For a fixed s, the weighting matrix $\mathcal{S}$ is defined by

$$S_r = \mathrm{diag}(1 - s_{11}, 1 - s_{21}, \ldots, 1 - s_{(n-1)1}, 1 - s_{12}, \ldots, 1 - s_{(n-1)n}) \tag{37}$$

$$S_c = \mathrm{diag}(1 - s_{11}, 1 - s_{21}, \ldots, 1 - s_{n1}, 1 - s_{12}, \ldots, 1 - s_{n(n-1)}) \tag{38}$$

$$\mathcal{S} = \begin{pmatrix} S_r & \\ & S_c \end{pmatrix}, \tag{39}$$

and for a fixed f, the weighting matrix A is defined by

$$A = \mathrm{diag}\Bigl(\sqrt{\lambda((\mathcal{L}_r f)_1^2 + (\mathcal{L}_c f)_1^2) + b}, \ldots, \sqrt{\lambda((\mathcal{L}_r f)_{n-1}^2 + (\mathcal{L}_c f)_{n-1}^2) + b}, \sqrt{\lambda(\mathcal{L}_c f)_n^2 + b},$$
$$\sqrt{\lambda((\mathcal{L}_r f)_n^2 + (\mathcal{L}_c f)_{n+1}^2) + b}, \ldots, \sqrt{\lambda((\mathcal{L}_r f)_{(n-1)^2}^2 + (\mathcal{L}_c f)_{n(n-1)}^2) + b},$$
$$\sqrt{\lambda(\mathcal{L}_r f)_{(n-1)^2+1}^2 + b}, \ldots \sqrt{\lambda(\mathcal{L}_r f)_{n(n-1)}^2 + b}, 0\Bigr). \tag{40}$$

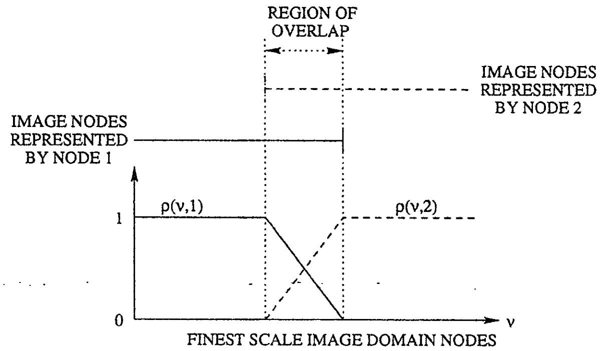

REGION OF OVERLAP
IMAGE NODES REPRESENTED BY NODE 2
IMAGE NODES REPRESENTED BY NODE 1
ρ(v,1)
ρ(v,2)
1
0
v
FINEST SCALE IMAGE DOMAIN NODES

Fig. 13. Diagram illustrating how $\rho(\nu,p)$ is determined at two overlapping sibling nodes.

Finally, the data vector γ when estimating s is given by

$$\gamma = \Bigg(\frac{\lambda((\mathcal{L}_r f)_1^2 + (\mathcal{L}_c f)_1^2)}{\lambda((\mathcal{L}_r f)_1^2 + (\mathcal{L}_c f)_1^2) + b}, \ldots, \frac{\lambda((\mathcal{L}_r f)_{n-1}^2 + (\mathcal{L}_c f)_{n-1}^2)}{\lambda((\mathcal{L}_r f)_{n-1}^2 + (\mathcal{L}_c f)_{n-1}^2) + b}, \frac{\lambda(\mathcal{L}_c f)_n^2}{\lambda(\mathcal{L}_c f)_n^2 + b},$$
$$\frac{\lambda((\mathcal{L}_r f)_n^2 + (\mathcal{L}_c f)_{n+1}^2)}{\lambda((\mathcal{L}_r f)_n^2 + (\mathcal{L}_c f)_{n+1}^2) + b}, \ldots, \frac{\lambda((\mathcal{L}_r f)_{(n-1)^2}^2 + (\mathcal{L}_c f)_{n(n-1)}^2)}{\lambda((\mathcal{L}_r f)_{(n-1)^2}^2 + (\mathcal{L}_c f)_{n(n-1)}^2) + b},$$
$$\frac{\lambda(\mathcal{L}_r f)_{(n-1)^2+1}^2}{\lambda(\mathcal{L}_r f)_{(n-1)^2+1}^2 + b}, \cdots \frac{\lambda(\mathcal{L}_r f)_{n(n-1)}^2}{\lambda(\mathcal{L}_r f)_{n(n-1)}^2 + b}, 0 \Bigg)^T . \qquad (41)$$

Given these definitions, each of the subproblems in the coordinate descent minimization of (33) involves the minimization of the quadratic functionals presented in (20) and (23).

II. Construction of H_x in the Overlapping Framework

The operator H_x is used in the overlapping framework to combine different estimates of a single process value. In particular, suppose one indexes lifted process values by $\bar{\nu}$ and image domain process values by ν, then if finest scale lifted process values $\mathbf{x}_{\bar{\nu}_1}, \mathbf{x}_{\bar{\nu}_2}, \ldots \mathbf{x}_{\bar{\nu}_q}$ all correspond to the same image domain process value x_ν, one uses H_x to write

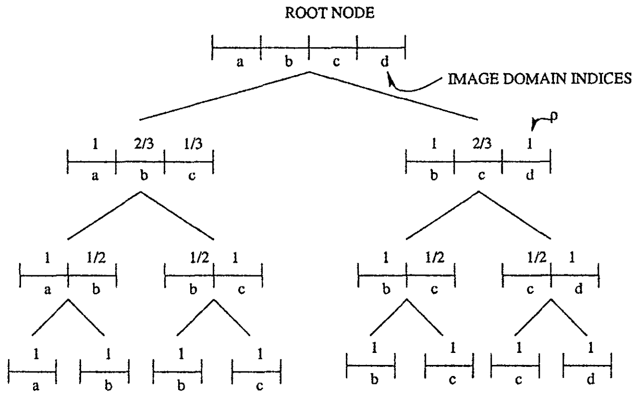
$$\hat{x}_\nu = \sum_{i=1}^{q} H_x(\bar{\nu}_i, \nu)\hat{\mathbf{x}}_{\bar{\nu}_i}. \qquad (42)$$

The only restriction on the values of $H_x(\bar{\nu}_i, \nu)$ is that they satisfy

$$\sum_{i=1}^{q} H_x(\bar{\nu}_i, \nu) = 1 \qquad (43)$$

for all finest scale image domain nodes ν. Details of the construction of the specific H_x operator employed in this paper are given here.

Consider the 1-D situation in Figure 13. One has two nodes at some scale in the tree which represent overlapping regions of pixels in the image domain. For each node p in the tree and



$$H_x = \begin{pmatrix} 1 & & & & & & & \\ & \frac{1}{3} & \frac{1}{3} & & \frac{1}{3} & & & \\ & & & \frac{1}{3} & & \frac{1}{3} & \frac{1}{3} & \\ & & & & & & & 1 \end{pmatrix}$$

Fig. 14. A 1-D example of how ρ and H_x are generated for a four pixel image sequence indexed by a, b, c, d.

finest scale image domain node ν, let $\rho(\nu, p)$ denote a weighting, relative to the parent of node p, of the degree to which the process value at node p represents the finest scale image domain process value at node ν. For this paper, $\rho(\nu, p)$ is chosen piecewise linear and such that

$$\sum_{z,\text{ siblings of } p} \rho(\nu, z) = 1 \tag{44}$$

for all ν and p. Figure 13 contains a graph of such a ρ. Once ρ is determined,

$$H_x(\bar{\nu}, nu) = \prod_{p,\text{ ancestors of } \bar{\nu}} \rho(\nu, p). \tag{45}$$

The example in Figure 14 illustrates these computations for a four point 1-D sequence.

References

[1] J. Shah, "Segmentation by nonlinear diffusion, II," in *Proc. IEEE Computer Vision and Pattern Recognition Conference*. IEEE, 1992.

[2] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[3] J. Morel and S. Solimini, *Variational Methods in Image Segmentation*, Birkhäuser, Boston, 1995.

[4] D. Mumford and J. Shah, "Boundary detection by minimizing functionals, I," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 1985.

[5] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577–684, 1989.

[6] L. Ambrosio and V.M. Tortorelli, "Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence," *Comm. Pure and Appl. Math*, vol. 43, no. 8, December 1990.

[7] L. Ambrosio and V.M. Tortorelli, "On the approximation of free discontinuity problems," *Bollettino Della Unione Matematica Italiana*, vol. 6-B, pp. 105–123, 1992.

[8] J. Kaufhold, M. K. Schneider, W. C. Karl, and A. S. Willsky, "A recursive approach to the segmentation of mri imagery," To appear in *International Journal of Pattern Recognition and Artificial Intelligence: Special Issue on Processing, Analysis and Understanding of Magnetic Resonance Images of the Human Brain.*

[9] M. Luettgen, W.C. Karl, and A.S. Willsky, "Efficient multiscale regularization with applications to the computation of optical flow," *IEEE Transactions on Image Processing*, vol. 3, no. 1, pp. 41–64, January 1994.

[10] P. Fieguth, *Application of Multiscale Estimation to Large Scale Multidimensional Imaging and Remote Sensing Problems*, Ph.D. thesis, MIT, June 1995.

[11] P. Fieguth, W. Karl, A. Willsky, and C. Wunsch, "Multiresolution optimal interpolation and statistical analysis of TOPEX/POSEIDON satellite altimetry," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 2, pp. 280–292, 1995.

[12] M. Luettgen, *Image Processing with Multiscale Stochastic Models*, Ph.D. thesis, MIT, May 1993.

[13] H. Pien and J. Gauch, "Variational segmentation of multi-channel MRI images," in *Proc. IEEE International Conference on Image Processing.* IEEE, November 1994.

[14] M. K. Schneider, "Multiscale methods for the segmentation of images," M.S. thesis, MIT, May 1996.

[15] K. Chou, *A Stochastic Modeling Approach to Multiscale Signal Processing*, Ph.D. thesis, MIT, May 1991.

[16] K. Chou, A. Willsky, and A. Beneveniste, "Multiscale recursive estimation, data fusion, and regularization," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 464–478, 1994.

[17] P. W. Fieguth, A. S. Willsky, and W. C. Karl, "Efficient multiresolution counterparts to variational methods for surface reconstruction," Submitted to *Computer Vision and Image Understanding.*

[18] P. Fieguth, W. Irving, and A. Willsky, "Multiresolution model development for overlapping trees via canonical correlation analysis," in *Proc. IEEE International Conference on Image Processing.* IEEE, 1995.

[19] R. Szeliski, *Bayesian modeling of uncertainty in low-level vision*, Kluwer Academic Publishers, Boston, 1989.