Change-point Problems
IMS Lecture Notes – Monograph Series (Volume 23, 1994)

CUBE SPLITTING IN MULTIDIMENSIONAL EDGE ESTIMATION

BY H. G. MÜLLER AND K.-S. SONG
University of California, Davis

Assume noisy measurements are available and that an edge or boundary is given which induces a partition of the domain into two subsets. The regression function on one subset is equal to a constant c_1, on the other subset to a constant c_2. Each measurement is made within a regular pixel. The problem we consider is the estimation of the edge or boundary curve (change curve), for the case that the domain is in $\Re^d$. We propose to seek boundary estimates as maximizers of a weighted squared difference statistic where we maximize over unions of cubes of aggregated pixels. Rates of almost sure convergence of this procedure are established. Its central advantage is its numerical feasibility, as the number of cubes of aggregated pixels to be investigated for inclusion in one of the partitioning sets can be kept small. A numerically efficient "cube splitting" ("CUSP") algorithm is suggested which implements this proposal: Start with an iteratively grown union of big cubes of aggregated pixels to find a first approximate edge/boundary estimate on a coarse level of approximation. Then split those cubes falling near the boundary into smaller cubes and check their allocation to one of the partitioning sets in order to obtain a more refined boundary estimate. This cube splitting (refinement) step may then be iterated until the desired level of resolution is achieved.

1. Introduction. Our main concern in this article is a numerically efficient way of estimating edges, i.e., discontinuities, of a regression function in higher dimensions. We also discuss asymptotic rates of almost sure convergence for one such estimation method. Our basic idea is to aggregate data into larger blocks ("cubes") and to assume that the true "edge" is also anchored on such larger blocks. The edge to be estimated therefore depends on the sample size n, reflecting the intuition that increasing sample size should allow for increasing degrees of resolution of the edge estimate.

The problem of edge estimation in higher dimensions recently found some interest among statisticians (see for instance the various approaches discussed

---

AMS 1991 Subject Classification: Primary 62G05; Secondary 62J02, 62H11

Key words and phrases: Break curve, change-point, discontinuity, image analysis, jump size, multivariate regression, pixels, rate of convergence, symmetric difference.

by Carlstein and Krishnamoorthy (1992), Korostelev and Tsybakov (1993), Müller and Song (1993), and Rudemo and Stryhn (1991) and also the previous work by Brodskii and Darkhovski (1986)). This problem has been around for some time in digital image processing, where edge estimation is an important tool for automatic image segmentation. Common methods there focus on binary images and the assumption of a Markov random field (see for instance Geman and Geman (1984)), and apparently methods related to statistical change-point models and their properties have not been much considered. The *edge estimation problem* considered here may also be referred to as estimation of a multidimensional change-point, boundary, change curve, or *break curve,* the latter name being particularly suited for regression models (as this problem is a natural extension of the one-dimensional break point regression model to higher dimensions).

We work in the context of the following fixed design regression problem: A sample of n measurements is available which are made at fixed locations $x_{i,n} \in [0,1]^d$ on a regular grid, i.e., $n = \prod_{\ell=1}^{d} n_\ell, i = (i_1, \ldots, i_d), 1 \le i_\ell \le n_\ell, 1 \le \ell \le d$, $\min_{1 \le \ell \le d} n_\ell \ge cn^{1/d}$ for a constant $c > 0$, and $x_{i,n} = \left(\frac{i_1 - 1/2}{n_1}, \ldots, \frac{i_d - 1/2}{n_d}\right) \in [0,1]^d$, where $d \ge 1$. Then assume

$$y_{i,n} = g(x_{i,n}) + \epsilon_{i,n} \tag{1}$$

with errors $\epsilon_{i,n}$.

If $d = 2$, we could view the data $y_{i,n}$ as noisy values of an image which are available on regular-sized "pixels". It is assumed that g (the "regression function" or "true image") has a simple structure: $g \equiv c_1$ on one side of the "edge" Γ_n, and $g \equiv c_2$ on the other side, where Γ_n divides $[0,1]^d$ into two subsets. The edge Γ_n depends on the sample size n. The jumpsize $\Delta = \Delta_n = c_2 - c_1$ may also depend on n.

In the following section, more precise definitions and assumptions are given and our main result on the rate of almost sure convergence when aggregating pixels into bigger blocks (cubes) is stated as a Theorem. Section 3 introduces an efficient cube splitting (CUSP) algorithm, which is based on the idea to first seek a coarse approximation to the edge based on big cubes of aggregated pixels which are less subject to random fluctuations. The approximation is then iteratively improved as the cubes are split into smaller and smaller subcubes. The performance of this algorithm is demonstrated for the two-dimensional case in an example discussed in Section 4. Section 5 contains the proof of the Theorem.

2. Convergence of Edge Estimate. The proposed edge estimator is a global method, i.e., the estimate is selected from a pool of candidate edges Γ_n.

As we consider only edges which divide $[0,1]^d$ into two subsets, we identify from now on the edge estimation problem with the equivalent problem of estimating a connected "plateau set" B_n, enclosed by Γ_n, which may depend on n, where $\Gamma_n = \partial B_n$. Assume for the regression function g in model (1),

$$g(x) = c_1 1_{\{x \in B_n\}} + c_2 1_{\{x \in B_n^c\}}. \tag{2}$$

Let $\mathcal{A}_n$ be a collection of "candidate plateau sets" each of which determines an edge Γ_n, and which are subsets of $\mathcal{B}([0,1]^d)$, the Borel sets in $[0,1]^d$. Consider the test statistic

$$T_n(A) = [\lambda(A)\lambda(A^c)]^{1/2} \left[(\#A^c)^{-1} \sum_{x_{i,n} \in A^c} y_{i,n} - (\#A)^{-1} \sum_{x_{i,n} \in A} y_{i,n} \right], \tag{3}$$

where λ is the Lebesgue measure in $\mathcal{B}([0,1]^d)$, and $\#A$ denotes the number of points $x_{i,n}$ with the property $x_{i,n} \in A$. Related statistics in a univariate distribution change-point problem within a maximum likelihood framework were considered by Bhattacharya and Brockwell (1976). Our estimator for the plateau set B_n is then

$$\hat{B}_n = \underset{A \in \mathcal{A}_n}{\operatorname{argmax}} |T_n(A)|, \tag{4}$$

compare Müller and Song (1992). This is the candidate plateau set which provides for the maximal weighted difference of means inside and outside the plateau set.

We need some further notation. Let $(D_{j,n})_{j=1,\dots,n}$ be a partition of $[0,1]^d$ into "pixels" $D_{j,n}$ such that $x_{j,n} \in D_{j,n}$, $x_{i,n} \notin D_{j,n}$ for $i \neq j$, and the $D_{j,n}$'s are cubes with volume $\frac{1}{n}$ and edges parallel to the coordinate axes. Let ψ_n be the "anchoring mapping" which anchors a given set on the pixels $D_{j,n}$, in the following sense: For a set $F \in \mathcal{B}([0,1]^d)$, let $\psi_n(F) = \bigcup_{F \cap D_{j,n} \neq \emptyset} D_{j,n}$. Then define "aggregated pixels" $C_{j,n}^{(m)}$, $m < n$, where $(C_{j,n}^{(m)})_{j=1,\dots,m}$, forms a partition of $[0,1]^d$ such that the $C_{j,n}^{(m)}$ are cubes with edges parallel to the axes and $C_{j,n}^{(m)} = \psi_n(C_{j,n}^{(m)})$, i.e., the $C_{j,n}^{(m)}$ are "anchored on the pixels". The number of cubes $(C_{j,n}^{(m)})$ is $m = m(n)$, so that

$$\lambda\left(C_{j,n}^{(m)}\right) = \frac{1}{m}, \quad j = 1, \dots, m, \tag{5}$$

and we note that for consistency of the plateau estimate, a minimum requirement is that $m\gamma_n^2 \log n/n \to 0$ according to (8).

We require the following assumptions (A1)–(A5), defining $\varphi_n^{(m)}$ as the "anchoring mapping" which anchors sets on cubes of aggregated pixels $C_{j,n}^{(m)}$, e.g., for sets $F \in \mathcal{B}([0,1]^d)$, let

$$\varphi_n^{(m)}(F) = \bigcup_{F \cap C_{j,n}^{(m)} \neq \emptyset} C_{j,n}^{(m)}. \tag{6}$$

(A1) The sequence of plateau sets B_n satisfies $B_n = \varphi_n^{(m)}(B_n)$ and $\eta < \lambda(B_n) < 1 - \eta$ for a constant η, $0 < \eta < 1/2$, which is independent of n.

One way a sequence B_n satisfying (A1) can be constructed from a fixed plateau set $B \in \mathcal{B}([0,1]^d)$ is to assume that $B_n = \varphi_n^{(m)}(B)$, $n \geq 1$. If B is sufficiently "regular", $\varphi_n^{(m)}(B) \downarrow B$ as $n \to \infty$. A further requirement is

(A2) All candidate plateau sets $A_n \in \mathcal{A}_n$ satisfy $A_n = \varphi_n^{(m)}(A_n)$, and with η as in (A1), $\eta < \lambda(A_n) < 1 - \eta$ and $\lambda(A_n \cap B_n) \geq \eta$, $\lambda(A_n^c \cap B_n^c) \geq \eta$.

Conditions (A1), (A2) reflect the basic idea to anchor true and candidate plateau sets on cubes of aggregated pixels. The last part of (A2) is an identifiability constraint.

(A3) The errors $\epsilon_{i,n}$ are i.i.d. with $\mathrm{E}\epsilon_{i,n} = 0$, $\mathrm{E}\epsilon_{i,n}^2 = \sigma^2 < \infty$, and $E|\epsilon_{i,n}|^s < \infty$ for some $s > 4$.

(A4) The jumpsize $\Delta = \Delta_n = c_2 - c_1$ satisfies $\Delta = \gamma_n^{-1}$ for a sequence γ_n which might be a sequence of constants (constant jump size case).

(A5) For an r with $4 < r < s, \liminf_{n\to\infty} n^{-2/r}(n/m\log n)^{1/2} > 0$; $n/(m\log n) \to \infty$.

As measure of distance between $\hat{B}_n$ and B_n we take the Lebesgue measure of the symmetric set difference, $\hat{B}_n \Delta B_n = (\hat{B}_n \cap B_n^c) \cup (\hat{B}_n^c \cap B_n)$.

THEOREM. *Under* (A1)-(A5),

$$\lambda\left(\hat{B}_n \Delta B_n\right) = O\left[\left(\frac{m\gamma_n^2 \log n}{n}\right)^{1/2}\right] \; a.s. \tag{7}$$

The proof of this result is in Section 5. We notice that as m gets larger, i.e., the size of cubes $C_{j,n}^{(m)}$ which have volume $\frac{1}{m}$ gets smaller and the degree of resolution of the estimate thus better, the rate decreases. For fixed m and fixed jump size, the rate is $[\log n/n]^{1/2}$. The estimator $\hat{B}_n$ is strongly consistent for B_n as long as

$$m\gamma_n^2 \log n/n \to 0. \tag{8}$$

For fixed jump size this requires that the edge length $b = m^{-1/d}$ of cubes $C_{j,n}^{(m)}$ satisfies

$$b/\left(\frac{\log n}{n}\right)^{1/d} \to \infty. \tag{9}$$

3. The Cube Splitting (CUSP) Algorithm. Computation of the plateau set estimate $\hat{B}_n$ (4) (and therefore of the edge estimate) requires in principle the evaluation of the statistic T_n over all candidate plateau sets in $\mathcal{A}_n$. As the number of elements grows approximately as 2^m, this becomes quickly unmanageable if $m \sim n$; even for the case $m \sim \log n$, when the problem is polynomial, the number of calculations can be extremely large for practically relevant sample sizes. We therefore propose here an iterative algorithm to find the estimate (4). The idea is to first seek an approximate solution on a reduced set of candidate plateau sets which is then refined iteratively along its boundary.

Assume that $n = (2^d)^p \ell^d k^d$, for integers $k, \ell, p \geq 0$, and let the smallest cubes $(C_{j,n}^{(m)})_{1 \leq j \leq m}$, defining the most refined level at which we want to estimate, correspond to $m = (2^d)^q \ell^d$, where $0 \leq q < p$. Here, the desired "level of resolution" m is defined in (5). Assume we start at a coarser level of resolution $m_1 = (2^d)^{q_1} \ell^d$, where $0 \leq q_1 \leq q$. Note that the finest "level of resolution" m considered corresponds to an edge estimate which is a union of cubes $(C_{j,n}^{(m)})_{1 \leq j \leq m}$ each of which consists of $(n/m) = (2^d)^{p-q} k^d$ pixels, whereas the coarser starting level m_1 corresponds to an edge estimate based on cubes $(C_{j,n}^{(m_1)})$, $1 \leq j \leq m_1$, each of them containing $(2^d)^{p-q_1} k^d$ pixels.

Consider the finite sequence $m_1 = (2^d)^{q_1} \ell^d$, $m_2 = (2^d)^{q_1+1} \ell^d, \ldots, m_r = (2^d)^q \ell^d = m$, where $r = q - q_1 + 1$, and corresponding partitions of $[0,1]^d$ into cubes $\{C_{j,n}^{(m_i)}\}_{1 \leq j \leq m_i}$, $1 \leq i \leq r$, where cubes $C_{j,n}^{(m_i)}$ have volume $\frac{1}{m_i}$ with edges parallel to the coordinate axes. Each cube $C_{j,n}^{(m_i)}$ contains exactly 2^d cubes of the partition $\{C_{j,n}^{(m_{i+1})}\}$. For a given cube $C_{j,n}^{(m_i)}$, consider the $(2d)$ cubes in $\{C_{jn}^{(m_i)}\}_{1 \leq j \leq m_i}$, which share a hyperplane with $C_{j,n}^{(m_i)}$ as the *neighbors* of $C_{j,n}^{(m_i)}$. For any set E with $E = \varphi_n^{(m_i)}(E)$, i.e., which is anchored on cubes $C_{j,n}^{(m_i)}$ according to (6), define the collections $\mathcal{A}^{(m_i)}(E)$ of all sets with one neighboring cube outside the boundary of E added to E, and $\mathcal{D}^{(m_i)}(E)$ of all sets with one cube at the boundary, neighboring the outside of E, deleted. Formally,

$\mathcal{A}^{(m_i)}(E) = \{E \cup C_{j,n}^{(m_i)} : C_{j,n}^{(m_i)} \subset E^c$ and $C_{j,n}^{(m_i)}$ is a neighbor of a cube $C_{j',n}^{(m_i)} \subset E\}$, $\mathcal{D}^{(m_i)}(E) = \{E \backslash C_{j,n}^{(m_i)} : C_{j,n}^{(m_i)} \subset E$ and $C_{j,n}^{(m_i)}$ has a neighbor $C_{j',n}^{(m_i)}$ with $C_{j',n}^{(m_i)} \subset E^c$, $\lambda(E \backslash C_{j,n}^{(m_i)}) > 0$, and each cube $C_{j'',n}^{(m_i)} \subset E \backslash C_{j,n}^{(m_i)}$ has a neighbor $C_{j''',n}^{(m_i)} \subset E \backslash C_{j,n}^{(m_i)}\}$.

The proposed cube splitting (CUSP) algorithm starts with the following initial step:

$$B_{11} = \operatorname*{argmax}_{A \in \{C_{j,n}^{(m_1)}\}_{1 \le j \le m_1}} |T_n(A)|,$$

i.e., the "best" single cube at level of resolution m_1 (first level) is selected. The aim at this first level is to find the best plateau set estimate anchored on cubes $\{C_{j,n}^{(m_1)}\}$ by iteratively adding the respectively "best" neighbors to the current plateau set estimate. "Best" means maximizing $|T_n(\cdot)|$. Accordingly, given B_{1j},

$$B_{1,j+1} = \operatorname*{argmax}_{A \in \mathcal{A}^{(m_1)}(B_{1j})} |T_n(A)|,$$

and choosing $B_1 = B_{1\ell}$ whenever $B_{1\ell} = B_{1,\ell+1}$ provides then the plateau set estimate at the first level.

At the second level, we split each cube $C_{j,n}^{(m_1)}$ into smaller cubes of the collection $\{C_{j,n}^{(m_2)}\}$ and consider alternating rounds of deleting and adding these smaller cubes along the periphery of the current estimate. Formally, noting that $B_1 = \varphi_n^{(m_2)}(B_1)$, $B_{21} = B_1$, $B_{2,j+1} = \text{argmax}_{A \in \mathcal{A}^{(m_2)}(B_{2j}) \cup \mathcal{D}^{(m_2)}(B_{2j})} |T_n(A)|$,

and $B_2 = B_{2\ell}$ if $B_{2_\ell} = B_{2,\ell+1}$. The estimate at the second level is determined whenever adding or deleting a cube $C_{j,n}^{(m_2)}$ is either not possible or will not improve the statistic T_n. This process is now iterated for levels $s+1 = 3, ..., r$:

$$B_{s+1,1} = B_s$$

$$B_{s+1,j+1} = \operatorname*{argmax}_{A \in \mathcal{A}^{(m_{s+1})}(B_{s+1,j}) \cup \mathcal{D}^{(m_{s+1})}(B_{s+1,j})} |T_n(A)|,$$

$$B_{s+1} = B_{s+1,\ell} \quad \text{if} \quad B_{s+1,\ell} = B_{s+1,\ell+1},$$

until

$$\hat{B} = B_r$$

is reached at the r-th level of the algorithm. This estimate then has the desired level of resolution m.

The rationale behind the algorithm is that gross errors in the general location of the initial (level 1) "big blocks" B_1 are unlikely as these blocks average over many pixels, keeping the random variation down. Successive refinement along the periphery then leads to $\hat{B}$. The procedure depends on a sensible choice of m_1 and $m = m_r$. The choice of m_1 has to be small enough to ensure that the initial "big block" estimate is approximately correct and is not critical. The last level of resolution on which $\hat{B}$ can be anchored is more difficult to determine. If m is too small, there may be a bias problem if one intends to estimate a "smooth" boundary. On the other hand, choice of a relatively large m may lead to improved resolution and bias but at the same time to increased variance and randomness in the location of the cubes at the periphery of $\hat{B}$.

A good choice will depend on the unknown signal-to-noise ratio of the data where the "signal" corresponds to the jump size Δ, whereas the noise is quantified by the error variance σ^2. One possibility to approach this problem would be to first choose m conservatively, i.e., rather small, to carry out the algorithm and to obtain the natural jump size estimate

$$\hat{\Delta} = \left\{ \frac{1}{\#\hat{B}^c} \sum_{x_i \in \hat{B}^c} y_i - \frac{1}{\#\hat{B}} \sum_{x_i \in \hat{B}} y_i \right\} \tag{10}$$

and at the same time a coarse estimate of σ^2, based on

$$\hat{\sigma}^2(A) = \frac{1}{\#A} \sum_{x_i \in A} y_i^2 - \left\{ \frac{1}{\#A} \sum_{x_i \in A} y_i \right\}^2 ,$$

so that with $\hat{B}^I \subset \hat{B}$, $\hat{B}^{cI} \subset \hat{B}^c$ being "interior" sets of $\hat{B}$, $\hat{B}^c$, we obtain

$$\hat{\sigma}^2 = \alpha \hat{\sigma}^2(\hat{B}^I) + (1-\alpha)\hat{\sigma}^2(\hat{B}^{cI}) \tag{11}$$

with properly determined weight α, $0 < \alpha < 1$, reflecting unequal sample sizes in $\hat{B}^I$, $\hat{B}^{cI}$.

We can then use $\hat{B}$, $\hat{\Delta}$ and $\hat{\sigma}^2$ to simulate the procedure and develop reasonable choices of m from such a Monte Carlo approach. As the CUSP algorithm is relatively fast, this approach is not prohibitively expensive.

4. A Two-Dimensional Example. The above algorithm was applied to a lattice of 60x60 regular pixels located equidistantly in $[0,1]^2$. The edge was defined by the circle $(x-0.5)^2 + (y-0.5)^2 = 0.25^2$. The measurements for pixels inside the circle were generated as $\mathcal{N}(1+|\Delta|, 1)$ pseudo random variables, those outside the circle as $\mathcal{N}(1,1)$ pseudo random variables. Thus $\sigma^2 = 1.0$, and three different jump sizes with values $|\Delta| = 1.5, 1.0$ and 0.5 were considered.

Observing that $n = 3600 = 4^2 5^2 3^2$, we choose $m_1 = 25$, $m_2 = 100$, and $m_3 = m = 400$ in the notation of Section 3 so that at the first level each aggregated cube consists of 144, at the second level of 36, and at the third level of 9 observations (pixels). At the third level, each cube is a square with edge length $\frac{1}{20}$.

The results for the three cases $|\Delta| = 1.5, 1.0$ and 0.5 utilizing each time the same pseudo-random numbers are shown in Figure 1-3. These are "intermediate" cases, being neither particularly good or bad. In the upper left corner is an "image plot" showing the pixeled data (the darker, the higher the value). The result of the first level (B_1 in the notation of Section 3) is displayed in the right upper corner, the result of the second level (B_2) in the left lower corner and the result of the third level ($B_3 = \hat{B}$) in the right lower corner. The jump size estimates (10) for the three cases are $|\hat{\Delta}| = 1.4978$, $|\hat{\Delta}| = 1.0254$ and $|\hat{\Delta}| = 0.5690$ for the true jump sizes of $|\Delta| = 1.5$, 1.0, 0.5 respectively.

Note that for $|\Delta| = 1.5$, the fit of $\hat{B}$ is very good, though not completely symmetric (for $|\Delta| = 3.0$, the algorithm in most cases achieved a perfect fit). For $|\Delta| = 1.0$, the estimate is still quite good, but it starts to show some random variation along the periphery. This random variation is considerably stronger for the case $\Delta = 0.5$. Note that in the image plot the circle is extremely hard to discern due to the low signal-to-noise ratio in this case.

Our conclusion is that the CUSP algorithm works very well, at least in the examples we have seen, and is able to track edges in unfavorable signal-to-noise situations. The algorithm is also quite fast and numerically efficient. It is therefore possible to apply it in computer-intensive methods for the construction of confidence regions and the selection of the level of resolution m, like bootstrap methods or the Monte Carlo procedures outlined in Section 3.

5. Proof of the Theorem. In the following, indices n will be omitted whenever feasible. Denote the collection of cubes $C_{j,n}^{(m)}$ by $\mathcal{C}_n$. We use the abbreviations

$$\rho(A) = [\lambda(A)\lambda(A^c)]^{1/2}, \; AB = A \cap B,$$

$$R(A) = \rho(A)\left\{(\#A^c)^{-1} \sum_{x_i \in A^c} \epsilon_i - (\#A)^{-1} \sum_{x_i \in A} \epsilon_i \right\},$$

for any sets $A, B \in \mathcal{B}([0,1]^d)$. The complement A^c of a set A is defined as $A^c = [0,1]^d \backslash A$, so that $\lambda(A^c) = 1 - \lambda(A)$. $\|\cdot\|^d$ denotes the Euclidean norm in $\Re^d$. We require the following lemmas. Define the edge length $b = b_n = m^{-1/d}$ for cubes $C_{j,n}^{(m)}$ of volume $\frac{1}{m}$ and let $\alpha_n = [\log n/(nb^d)]^{1/2}$. The assumptions for the Theorem are required to hold.

LEMMA 5.1.

$$\sup_{A \in \mathcal{A}_n} |\frac{1}{\#A} \sum_{x_i \in A} \epsilon_i| = O(\alpha_n) \qquad a.s. \tag{12}$$

PROOF. We apply (A3) and first show:

$$\sup_{C \in \mathcal{C}_n} |\frac{1}{\#C} \sum_{x_i \in C} \epsilon_i| = O(\alpha_n) \qquad a.s. \tag{13}$$

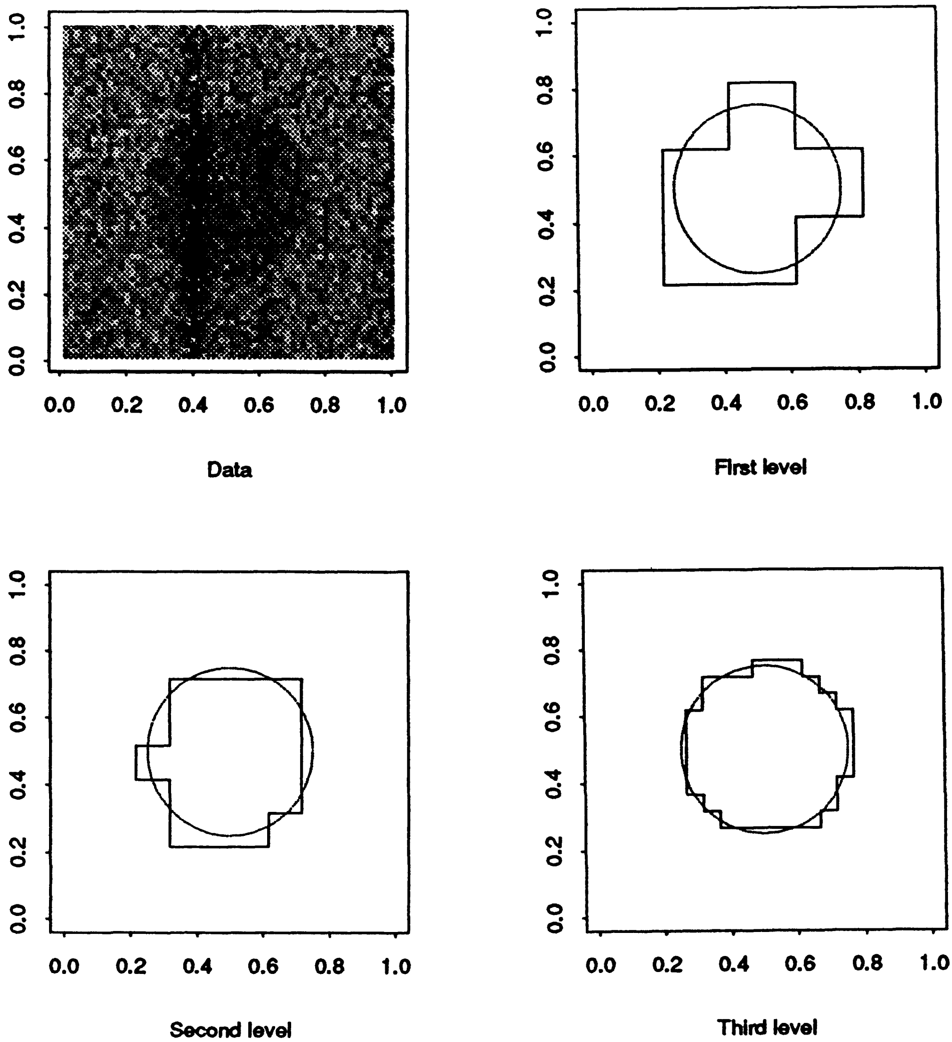

Figure 1: Cusp algorithm for $\Delta = 1.5$

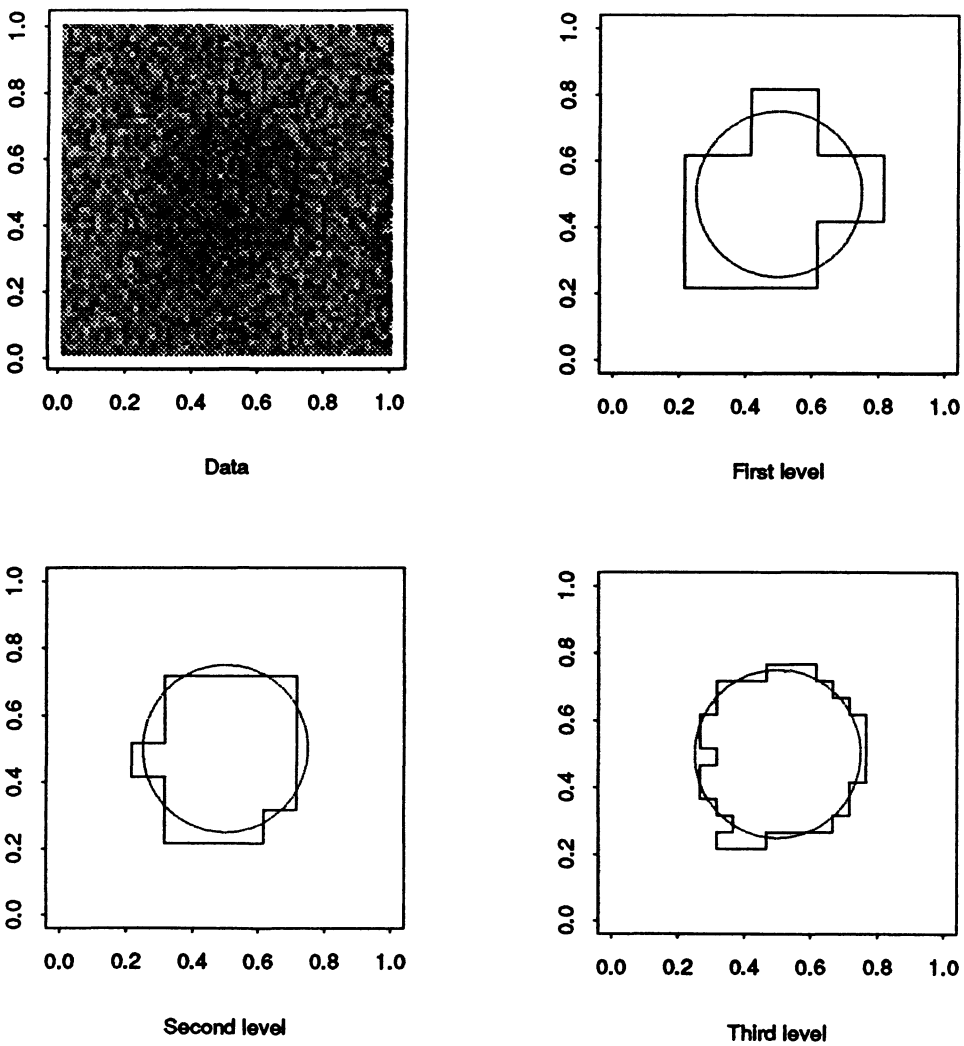

Figure 2: Cusp algorithm for $\Delta = 1.0$

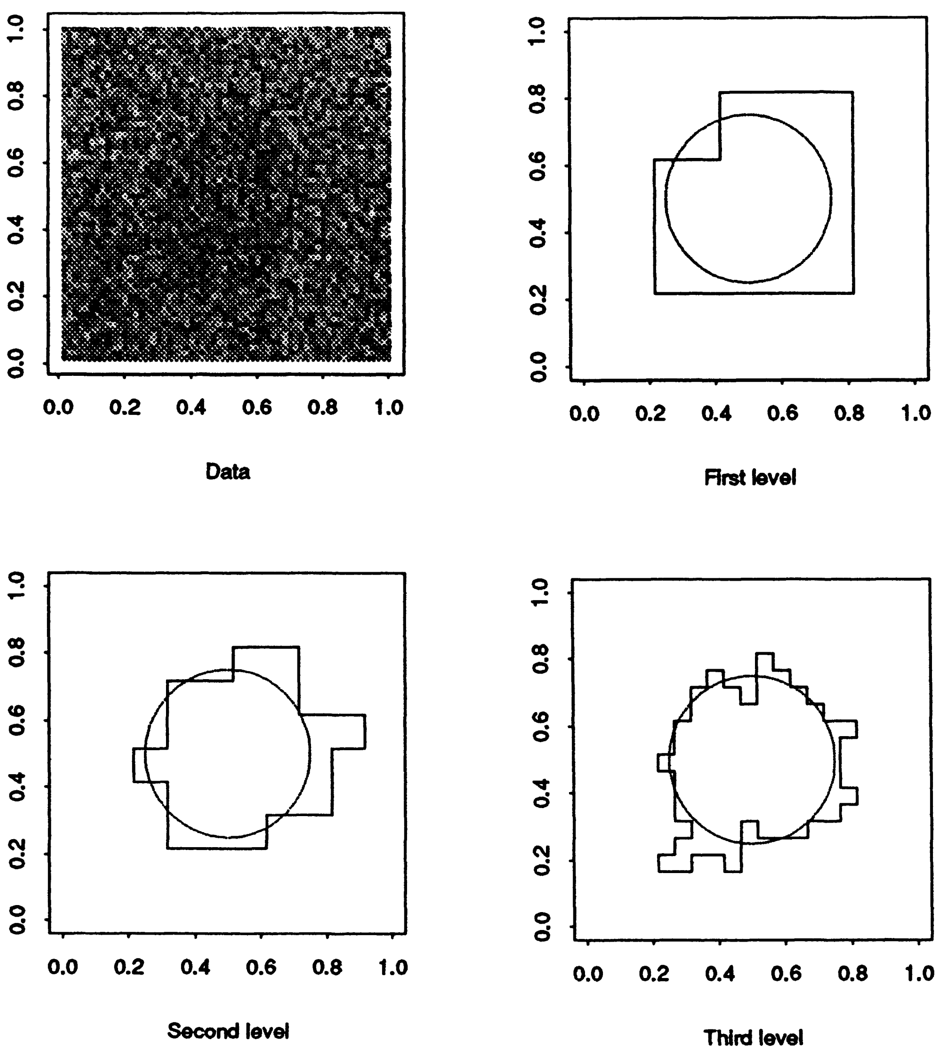

Figure 3: Cusp algorithm for $\Delta = 0.5$

Defining $\bar{\epsilon}_i = \epsilon_i 1_{\{|\epsilon_i| \le (in)^{1/r}\}}$, where r is as in (A5), and $W_{i,C} = (nb^d)^{-1} 1_{\{x_i \in C\}}$ for $C \in \mathcal{C}_n$, we observe

$$\sup_{C \in \mathcal{C}_n} |\frac{1}{\#C} \sum_{x_i \in C} \epsilon_i| \le \sup_{C \in \mathcal{C}_n} |\sum W_{i,C}(\epsilon_i - \bar{\epsilon}_i)| + \sup_{C \in \mathcal{C}_n} |\sum W_{i,C}\, \bar{\epsilon}_i| = I + II.$$

For term I, we apply the same arguments as in Lemma 5.2 in Müller and Stadtmüller (1987) to show that

$$\sup_{C \in \mathcal{C}_n} |\sum W_{i,C}(\epsilon_i - \bar{\epsilon}_i)| = O\left(n^{2/r} \sup_{C \in \mathcal{C}_n} |W_{i,C}|\right) \; a.s.,$$

and observing $\sup_{C,i} |W_{i,C}| \le 1/(nb^d)$, we obtain from (A5)

$$I = O(\alpha_n) \qquad a.s.$$

Using the fact that $\sup_{C \in \mathcal{C}_n} \# \{W_{i,C} \ne 0\} = O(nb^d)$, we find $\sup_{C \in \mathcal{C}_n} [\sum W_{i,C}^2 \log n]^{1/2} = O(\alpha_n)$, which implies

$$II = O(\alpha_n) \qquad a.s.$$

as in Müller and Stadtmüller (1987). Observing that by (13),

$$\sup_{A \in \mathcal{A}_n} |\frac{1}{\#A} \sum_{x_i \in A} \epsilon_i| \le \frac{\#\{C_{i,n}^{(m)} \subset A\}}{\#A/\#C} O(\alpha_n) \qquad a.s.,$$

the result follows from (A2). ∎

LEMMA 5.2. *For any sets $A, B \in \mathcal{B}([0,1]^d)$,*

$$T_n(A) = \rho(A) \left\{ \frac{\#(AB)}{\#A} - \frac{\#(A^cB)}{\#A^c} \right\} \Delta + R(A). \tag{14}$$

PROOF. By direct calculation, (1) and (2) yield

$$T_n(A) = \rho(A) \Bigg\{ c_2 \left[\frac{\#A^c - \#(A^cB)}{\#A^c} - \frac{\#A - \#(AB)}{\#A} \right] - c_1 \left[\frac{\#(AB)}{\#A} - \frac{\#(A^cB)}{\#A^c} \right] \Bigg\} + R(A),$$

which implies (14). For more details, see Müller and Song (1993). ∎

Define for $A, B \in \mathcal{B}([0,1]^d)$,

$$Q_0(A,B) = \rho(B) - [\rho(A)]^{-1} |\lambda(A^c)\lambda(AB) - \lambda(A^cB)\lambda(A)|.$$

LEMMA 5.3.

$$|T_n(B_n)| - |T_n(\tilde{A}_n)| \;=\; |\Delta| Q_0(\tilde{A}_n, B_n) + O(\alpha_n) \;\; a.s. \tag{15}$$

PROOF. Applying Lemma 5.1 and Lemma 5.2 for $A = \tilde{A}_n, B = B_n$, one finds

$$|T_n(B_n)| - |T_n(\tilde{A}_n)| = \rho(B_n)|\Delta| - \rho(\tilde{A}_n)|\frac{\lambda(\tilde{A}_n B_n)}{\lambda(\tilde{A}_n)} - \frac{\lambda(\tilde{A}_n^c B_n)}{\lambda(\tilde{A}_n^c)}| \, |\Delta| \\ + \; O(\alpha_n) \qquad a.s.,$$

and therefore (15). ∎

To prove the Theorem, we find by some set algebra for any sets A, $B \in \mathcal{B}([0,1]^d)$, defining $Q(A,B) = \lambda(AB^c)\lambda(A^cB)\lambda(A^cB^c) + \lambda(AB)\lambda(A^cB)\lambda(A^cB^c)$ $+ \lambda(AB)\lambda(AB^c)\lambda(A^cB^c) + \lambda(AB)\lambda(AB^c)\lambda(A^cB)$, that $Q(A,B) + [\lambda(AB) - \lambda(A)\lambda(B)]^2 = \lambda(A)\lambda(A^c)\lambda(B)\lambda(B^c)$, and furthermore,

$$Q_0(A,B) = \frac{Q(A,B)}{\rho(A)\,[\rho(A)\rho(B) + |\lambda(AB) - \lambda(A)\lambda(B)|]} \geq \frac{Q(A,B)}{2}. \tag{16}$$

Observing (A1), (A2), we obtain

$$Q(A,B) \;\geq\; 2\eta^2\lambda(A\Delta B).$$

With (14) and (16), this implies,

$$|T_n(B_n)| \;-|T_n(\tilde{A}_n)| \;\geq\; |\,\Delta|\eta^2\lambda(\tilde{A}_n\Delta B_n) + O(\alpha_n) \qquad a.s. \tag{17}$$

As by definition, $|T_n(\tilde{A}_n)| \;\geq\; |T_n(B_n)|$ a.s., it follows that $\lambda(\tilde{A}_n \Delta B_n) = O(\alpha_n/|\Delta|)$ a.s. and thus the Theorem follows with (A4).

REFERENCES

BHATTACHARYA, P. K. and BROCKWELL, P.J . (1976). The minimum of an additive process with applications to signal estimation and storage theory. *Z. Wahrsch. verw. Gebiete* **37**, 51–75.

BRODSKII, B. and DARKHOVSKI, B. (1986). The a posteriori method of detecting the disruption of a random field. In: *Detection of Changes in Random Processes* (L. Telksnys, ed.) 32–38. Optimization Software, New York.

CARLSTEIN, E. and KRISHNAMOORTHY, C. (1992). Boundary estimation. *J. Amer. Statist. Assoc.* **87**, 430–438.

GEMAN, S. and GEMAN, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.* **6**, 721–741.

KOROSTELEV, A. P. and TSYBAKOV, A. B. (1993). *Minimax Theory of Image Reconstruction. Lecture Notes in Statistics* **82**, Springer-Verlag, NY.

MÜLLER, H. G. and SONG, K. S. (1993). Edge estimation in two-region images. Technical Report, Division of Statistics, UC Davis.

MÜLLER, H. G. and STADTMÜLLER, U. (1987). Estimation of heteroscedasticity in regression analysis. *Ann. Statist.* **15**, 610–625.

RUDEMO M. and STRYHN, H. (1991). Approximating the distribution of the maximum likelihood estimator in two-region images. Technical Report, Agricultural University, Copenhagen.

DIVISION OF STATISTICS
UNIVERSITY OF CALIFORNIA
DAVIS, CA 95616