A Minimum Description Length–Based Image Segmentation Procedure, and Its Comparison With a Cross-Validation–Based Segmentation Procedure

Thomas C. M. LEE

Image segmentation is a very important problem in image analysis, as quite often it is a key component of a good practical solution to a real-life imaging problem. It aims to partition a digital image into a set of nonoverlapping homogeneous regions. One approach to segmenting an image is to fit a piecewise constant function to the image and define the segmentation by the discontinuity points of the fitted function. The article's first contribution is to present a new and automatic segmentation procedure which follows this piecewise constant function fitting approach. This procedure is based on Rissanen's minimum description length (MDL) principle and consists of two components: (a) an MDL-based criterion in which the "best" segmentation (i.e., the "best" fitted piecewise constant function) is defined as its minimizer and (b) a fast-merging algorithm that attempts to locate this minimizer. As a second contribution, the new MDL-based procedure is compared with a cross-validation based segmentation procedure. Empirical results from a simulation study suggest the new MDL-based procedure is superior. Some possible extensions of the MDL-based procedure are also described.

KEY WORDS: Cross-validation; Image segmentation; Minimum description length; Piecewise constant function fitting; Region merging; Segmentation quality evaluation.

1. INTRODUCTION

1.1 Why Segmentation?

This article concerns the problem of image segmentation. Loosely speaking, the aim of segmentation is to locate the boundaries of objects captured in images. It is a very important problem in image analysis, as it is the step that changes the basic elements that one can work with from the highly localized pixels to the more meaningful and global segmented objects. If this segmentation step is performed well, then subsequent image analysis steps are made simpler and easier. Here I present two real-life examples that require segmentation.

Example 1. Figure 1(a) shows a digitized micrograph of grains in rolled aluminium coils. The grains can be differentiated visually by changes in shades of gray. This image is one of a series provided to the CSIRO Mathematical and Information Sciences by an industrial client. According to the client, it is extremely useful if various grain characteristics such as the distributions of their areas, perimeters, and orientations can be automatically measured, as these quantities are good predictors of macroscopic properties of interest. The first step in the measurement process is the automated location of grain boundaries.

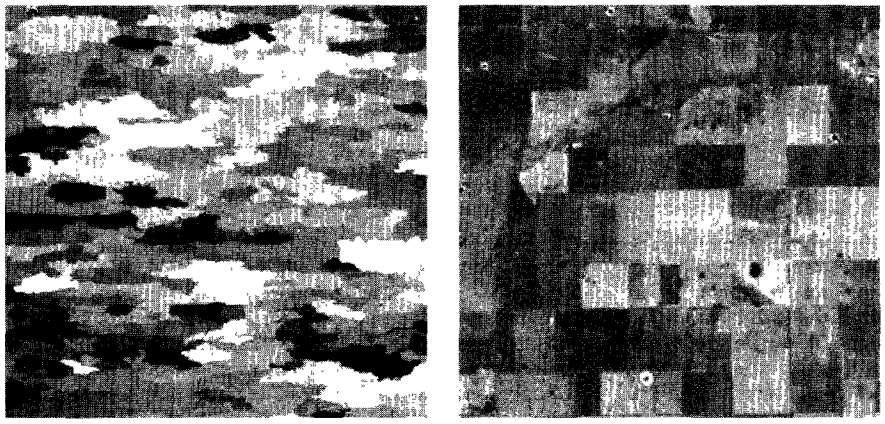
Example 2. Figure 1(b) displays an image of the fifth band of a Landsat Thematic Mapper scene of the Esperance region of Western Australia. Farmers farming around that region are interested in the productivity of their paddocks, and the ultimate aim of this "paddock project" is to report productivity measures on a per paddock basis. Therefore, paddock boundaries need to be obtained.

1.2 Background

More formally, image segmentation aims to partition a digital image into a set of nonoverlapping regions, so that (a) pixels within the same region are homogeneous with respect to some characteristic (e.g., grayvalue or texture) and (b) pixels of neighboring regions are significantly different with respect to the same characteristic. (For general background readings on the topic, see, e.g., Haralick and Shapiro 1992, chap. 10, and Glasbey and Horgan 1995, chap. 4.) The main image characteristic as the basis for segmentation with which this article is concerned is grayvalue. Segmentation based on other image characteristics can often be fulfilled by first applying a preprocessing step to the image so that each pixel is assigned a value called the characteristic index, and then using a suitable grayvalue segmentation technique to segment the "characteristic indexed image." A good example is the segmentation of images based on roughness; one can use estimates of, say, fractal dimension as the characteristic indices.

There are many approaches to segmenting a grayscale image; one approach is to fit a two-dimensional piecewise constant function to the image and define the segmentation (i.e., region boundaries) by the discontinuity points of the fitted piecewise constant function. This article's primary goal is to present a new segmentation procedure that follows this approach. This new procedure is based on the minimum description length (MDL) principle (see, e.g., Rissanen 1989). In brief, the MDL principle defines the best segmentation

Thomas C. M. Lee is Assistant Professor, Department of Statistics, Colorado State University, Fort Collins, CO 80523 (E-mail: *tlee@stat.colostate.edu*). Most of this work was completed while the author was a Ph.D. student jointly at the Statistics Department, Macquarie University, and CSIRO Mathematical and Information Sciences, Australia. The author would like to express his gratitude to his supervisors, Mark Berman and Victor Solo, for their continuous encouragement, guidance, and patience. He would also like to thank Murray Cameron for a fruitful discussion and providing a preprint of Cameron, Hannan, and Speed (1995); Byron Dom for providing a preprint of Kanungo, Dom, Niblack, Steele, and Sheinvald (1995); and the reviewers, associate editor, and editor for many helpful comments. Revision of this article was completed while the author was visiting the Department of Statistics, University of Chicago.



(a) (b)

Figure 1. An Aluminium Grain Image (a) and a Landsat Image (b).

as the one that enables the best encoding (or compression) of the image.

Another focus of this article is a direct empirical comparison between the new MDL-based segmentation procedure and a cross-validation (CV)–based segmentation procedure recently proposed by Bose and O'Sullivan (1997). The idea behind this CV-based procedure is that it attempts to construct an estimator for the mean integrated squared error by using the "leave-one-out" mechanism and defines the best segmentation as the one that minimizes this estimator. Thus it is clear that the MDL- and CV-based procedures are based on two very different philosophies, and that a direct head-to-head comparison between them would be interesting. Also, it is worth noting that very few direct comparisons between these two methodologies have appeared in the literature.

Both the MDL-based and the CV-based procedures define their "best" segmentations as the minimizer of some objective criterion. For an image of reasonable size, practical global minimization of such an objective criterion is very difficult. To overcome this difficult minimization problem, I use the popular *region merging* strategy.

The article is organized as follows. Section 2 describes region merging. Section 3 derives the MDL-based segmentation criterion, and Section 4 discusses how to (approximately) minimize this criterion. Section 5 briefly reviews the CV-based procedure of Bose and O'Sullivan (1997), and Section 6 discusses evaluating segmentation quality. Section 7 reports results of a direct comparison between the MDL-based and CV-based procedures, and Section 9 discusses other issues and extensions related to the MDL-based procedure. Section 10 gives a conclusion, and two Appendixes provide mathematical details.

2. REGION MERGING: BASIC CONCEPTS

A region merging strategy is often applied to approximate the "best" segmentation, which is implicitly defined as the minimizer of some objective function, Q say. The general idea is as follows. It starts with computing the value of Q of an *oversegmentation* (see Sec. 4.1 for details) of the image being segmented. Then at each time step it chooses two neighboring regions, merges them to form a new region, and updates the corresponding Q value. Usually these two neighboring regions are chosen in a "greedy" way: When they are merged, it provides the largest reduction in (or smallest addition to) the current value of Q among all other possible merges. The region merging process continues until only one region is left. If K initial regions are in the original oversegmentation, then, when the merging process is done, a sequence of K nested segmentations is produced, and each of these segmentations is associated with a Q value. The one with the smallest Q value is then chosen as the final segmentation.

All the foregoing ideas is well illustrated by the example given in Figure 2. Figure 2(a) shows the "raw data"; that is, the image to be segmented. For the moment, disregard Figure 2(b). Figure 2(c) is one possible oversegmentation, from which a nested sequence of segmentations will be generated. Three different segmentations from such a nested sequence are displayed in Figures 2(d), (e), and (f). The segmentation in Figure 2(d) is unsatisfactory, as it is "undermerged" (i.e., not enough regions were merged), and the one in Figure 2(f) is also unsatisfactory, as it is "overmerged" (i.e., too many regions were merged). But the segmentation in Figure 2(e) is "just about right" and should have the smallest Q value among all other nested segmentations, and hence it should be selected.

Clearly, the form of the objective function Q will have a great impact on the quality of the final segmentation chosen. This is why methods like MDL and CV are used in image segmentation problems, as they provide a means for constructing such objective functions. Other merging (and splitting) methods have been proposed by Besl and Jain (1988), Beaulieu and Goldberg (1989), Chen, Lin, and Chen (1991), and Chang and Li (1994).

3. IMAGE SEGMENTATION BY THE MINIMUM DESCRIPTION LENGTH PRINCIPLE

In this section I define the problem that this article considers, and demonstrate how to construct an objective segmentation criterion using the MDL principle.

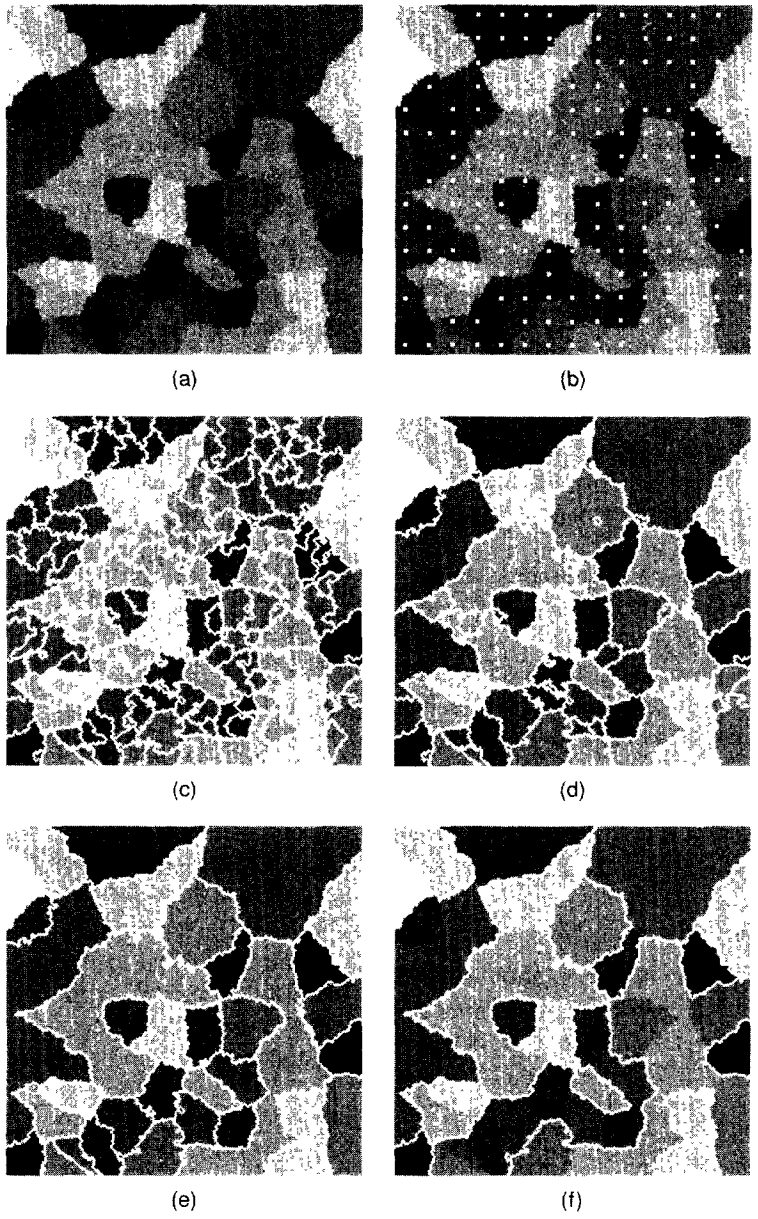


Figure 2. Region Merging. (a) Observed image; (b) regularly spaced 5 × 5 seeds (notice that some seeds have been rejected for having a high grayvalue variance); (c) oversegmentation obtained by applying SRG; (d) "undermerged" segmentation; (e) "about right" segmentation; (f) "overmerged" segmentation.

3.1 Image Segmentation as Model Selection

In digital image processing, an image can be viewed as a regularly sampled two-dimensional function; that is, a matrix of heights or grayvalues. Elements of such a matrix are called pixels. In this article I am interested mainly in images that can be well approximated by two-dimensional piecewise constant functions.

Let f be an arbitrary two-dimensional piecewise constant function with k constant regions. Denote the grayvalue of the jth region by μ_j $(j = 1, \ldots, k)$. Then a regularly sampled discrete version $\mathbf{f} = (f_1, \ldots, f_n)^T$ of f can be represented by

$$f_i = \sum_{j=1}^{k} \mu_j I_{\{i \in r_j\}}, \qquad i = 1, \ldots, n,$$

where n is the total number of pixels, $i \in r_j$ means "the ith pixel is in the jth region," and I_E is the indicator function for the event E. Notice that I have chosen to use single indexing rather than double indexing for labeling pixel coordinates. Also notice that in my formulation for $\mathbf{f}$, (discretized) region boundaries are composed of horizontal and vertical "edges" between pixels. Let $\boldsymbol{\Omega} = \{r_1, \ldots, r_k\}$; that is, $\boldsymbol{\Omega}$ defines a partition of the image. I also write $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)^T$.

The problem that I consider is as follows. Given an observed image $\mathbf{y} = (y_1, \ldots, y_n)^T$ satisfying $y_i = f_i + \varepsilon_i, \varepsilon_i \sim$ iid $\mathrm{N}(0, \sigma^2), i = 1, \ldots, n$, I want to obtain an estimate $\hat{\mathbf{f}}$ of $\mathbf{f}$ (or, equivalently, estimates of $k, \boldsymbol{\Omega}$ and $\boldsymbol{\mu}$), and define the segmentation of the image by the discontinuity points of $\hat{\mathbf{f}}$. Because of the equivalence between $\mathbf{f}$ and $k, \boldsymbol{\Omega}$ and $\boldsymbol{\mu}$, the problem of estimating $\mathbf{f}$ can be seen as a model selection problem, with each model $\boldsymbol{\theta}_k = \{k, \boldsymbol{\Omega}, \boldsymbol{\mu}\}$ specified by three

vector-valued parameters. Later I write $\hat{\boldsymbol{\theta}}_k = \{\hat{k}, \hat{\boldsymbol{\Omega}}, \hat{\mu}\}$ as an estimate of $\boldsymbol{\theta}_k = \{k, \boldsymbol{\Omega}, \boldsymbol{\mu}\}$, and I use Rissanen (1989)'s MDL principle to select the "best" model.

3.2 Model Selection by the Minimum Description Length Principle

The MDL principle defines the best model as the one that enables the best encoding (or compression) of the data; that is, the best fitted model is the one that produces the shortest code length of the data. For this article, one can treat the code length of the data as the amount of memory space required to store the data. Typically, the code length for a dataset has two parts: a fitted model plus the data "conditioned on" the fitted model (i.e., the residuals). For the present case, the data is the image $\mathbf{y}$ to be segmented, and a fitted model is simply a fitted two-dimensional piecewise constant function $\hat{\boldsymbol{\theta}}_k$ of the image.

Therefore, to apply the MDL principle to tackle the segmentation problem, first a code length expression that calculates the amount of space required to store an arbitrarily fitted two-dimensional piecewise constant function plus the corresponding residuals must be constructed. Then the best model, or segmentation, is defined as the minimizer of this code length expression. But before such a code length expression can be constructed, methods to produce encodable representations for the estimated region boundaries $\hat{\boldsymbol{\Omega}}$ are needed.

3.3 Freeman's Chain Code

A simple but effective boundary representation method that captures all spatial information is *Freeman's chain code* (see, e.g., Haralick and Shapiro 1992, chap. 18). Imagine that one is traveling along a region's boundary. Because region boundaries are composed of horizontal and vertical "edges" between pixels, at each step one can travel in only one of three directions: forward (F), left (L), or right (R). By utilizing this fact, Freeman's chain code for representing a single region consists of a *starting pixel edge,* a prescribed global travel direction (either clockwise or counterclockwise), and a *direction chain.* The direction chain indicates the local direction for each move. As an example, the direction chain of the gray region in Figure 3 is "FLFFLLRRLLFLRL" in the counterclockwise direction.

Therefore, by using Freeman's chain code, the encoding of a fitted two-dimensional piecewise constant function is equivalent to the encoding of the following:

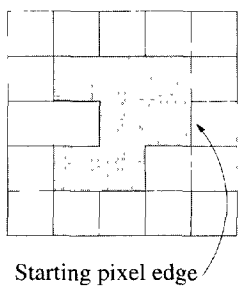


Figure 3. Freeman's Chain Code.

- $\hat{k}$, the estimated number of regions
- all starting pixel edges, which can be specified by the coordinates of the starting pixels (This is because the decoder and the encoder can agree beforehand on using the rightmost edge, say, of a starting pixel as the starting pixel edge.)
- direction chains of all regions
- $\hat{\mu}_j$'s, estimated grayvalues of the $\hat{k}$ regions defined by (A.1) in Appendix A.

I have ignored the encoding of the global travel direction of Freeman's chain code and the estimate of the noise variance σ^2, as they both produce constant code lengths.

3.4 Minimum Description Length Criterion for Image Segmentation

If a_j and b_j are the area (in terms of number of pixels) and perimeter (in terms of number of pixel edges) of the jth region of $\hat{\boldsymbol{\Omega}}$, then it is shown in Appendix A that the MDL criterion

$$\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}}) = \hat{k} \log n + \frac{\log 3}{2} \sum_{j=1}^{\hat{k}} b_j + \frac{1}{2} \sum_{j=1}^{\hat{k}} \log a_j + \frac{n}{2} \log \left(\frac{\mathrm{RSS}_{\hat{k}}}{n} \right) \quad (1)$$

is an approximation to the total code length of $\mathbf{y}$ (including both the fitted model part and the residual part). Here $\mathrm{RSS}_{\hat{k}} = \sum_{i=1}^{n} (y_i - \hat{f}_i)^2$ is the residual sum of squares with $\hat{f}_i$ defined by (A.1) in Appendix A. I propose to select the "best" fitted model as the minimizer of $\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$, and define our segmentation by its discontinuity points (i.e., boundaries separating all constant regions).

4. MERGING ALGORITHM AND SOME COMPUTATIONAL ISSUES

Global minimization of $\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$ is infeasible for images of a reasonable size. Here I recommend using the greedy region merging strategy described in Section 2 to approximate the minimizer of $\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$. This section describes a method for obtaining an *oversegmentation* for the merging algorithm to start with, and offers suggestions for efficient implementation of the merging strategy.

4.1 Oversegmentation

An ideal oversegmentation should be easy and fast to obtain, contain not too many segmented regions, and have its region boundaries be a superset of the true image region boundaries. How one obtains an oversegmentation depends on the nature of the image. Here we describe one ad hoc but reliable method that uses the *seeded region growing* (SRG) segmentation technique developed by Adams and Bischof (1994) to achieve our aim.

SRG begins with identifying, either manually or automatically, a set of *seeds* from the image to be segmented. Each seed can be a single pixel or a set of connected pixels. Then SRG grows seeds outward in a way that maintains the grayscale homogeneity of each growing seed/region; grow-

ing a region is equivalent to successively adding neighboring pixels to that region. In other words, at each time step all neighboring pixels of the growing regions are examined, and the pixel with a grayvalue closest to the mean grayvalue of its neighboring region is added to that region. A segmentation is obtained when the whole image is engulfed by the growing regions.

Thus one way to achieve a segmentation of an image is to identify one and only one seed for each true region in the image and simply apply the growing mechanism. However, such perfect seed identification is not always possible without human interaction. Nevertheless, if it can be safely assumed that the true regions of an observed image are reasonably regular in shape and of size greater than, say, 25×25 pixels, then, if seeds are regularly placed in such an observed image with these seeds 20 pixels apart (in both directions), it is very likely that at least one seed will be placed in each true region. Thus by applying SRG with these regularly spaced seeds, an oversegmented image can be obtained for the greedy region merging strategy to start with.

When using SRG, if the signal-to-noise ratio is low, then it is advisable to use a set of connected pixels (e.g., a square of 5×5 pixels) rather than a single pixel as one seed. This is to stabilize the initial estimates of the seed grayvalues. The price for having this stabilizing effect is that some of these regularly spaced seeds may fall on a true region boundary and have adverse effects on subsequent processing. To overcome this, a further step of seed rejection can be applied before using SRG. For example, one can reject a seed if the variance of its individual pixel grayvalues is too high (see Fig. 2).

4.2 Graph and Updating Formulas

Despite the fact that the greedy region merging strategy described earlier drastically cuts down the search space of $\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$, naive implementations of the algorithm could result in an unduly long computation time for large-sized images. Here I suggest methods for efficient implementation of the algorithm.

In the spatial context, a region typically has more than two neighboring regions. This multiplies the number of comparisons of MDL reductions in our greedy merging strategy and presents the need for efficient representation and manipulation of the neighboring system of the regions and fast computation of different MDL reductions. To meet the first need, I have used graphs. Figure 4 provides a simple illustration.

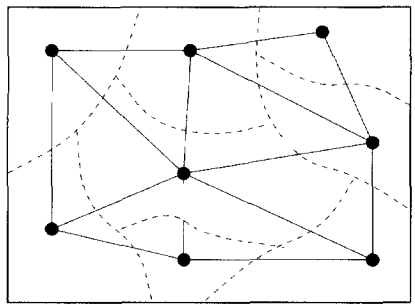


Figure 4 A Simple Graph. (- - - -, region boundaries; ———, graph links; •, graph nodes)

For the second need, updating formulas can be used to speed up MDL reduction computations. Some notation is needed to proceed. Suppose that after completion of a particular step of the algorithm, a fitted model $\hat{\boldsymbol{\theta}}_k$ with $\hat{k}$ regions is obtained. Denote the $\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$ value by $\mathrm{MDL}_{\hat{k}}$ and the RSS of $\hat{\boldsymbol{\theta}}_k$ by $\mathrm{RSS}_{\hat{k}}$. Also denote the $\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$ value and the RSS of the model obtained from merging the ith and the jth regions $(1 \le i < j \le \hat{k})$ of $\hat{\boldsymbol{\theta}}_k$ by $\mathrm{MDL}_{\hat{k}-1,i,j}$ and $\mathrm{RSS}_{\hat{k}-1,i,j}$. Then it can be shown that

$$\mathrm{RSS}_{\hat{k}-1,i,j} = \mathrm{RSS}_{\hat{k}} + a_i \hat{\mu}_i^2 + a_j \hat{\mu}_j^2 - (a_i \hat{\mu}_i + a_j \hat{\mu}_j)^2/(a_i + a_j)$$

and

$$\begin{aligned} \Delta\mathrm{MDL}_{\hat{k}-1,i,j} &= \mathrm{MDL}_{\hat{k}} - \mathrm{MDL}_{\hat{k}-1,i,j} \\ &= \log n + \frac{1}{2}\log(a_i a_j/(a_i + a_j)) + \frac{\log 3}{2}\, b_{ij} \\ &\quad + \frac{n}{2}\log(\mathrm{RSS}_{\hat{k}}/\mathrm{RSS}_{\hat{k}-1,i,j}), \end{aligned}$$

where b_{ij} is the number of boundary pixel edges common to both the ith and the jth regions.

Thus efficient implementation of the merging algorithm can be achieved by using the graph data structure, allocating memory space to store different values of $\mathrm{MDL}_{\hat{k}}$, $\mathrm{RSS}_{\hat{k}}, a_j, b_{ij}$, and $\hat{\mu}_j$, and applying the foregoing updating formulas to compute all related values. For an image of dimension 512×512 with an initial oversegmentation of about 800 initial regions, my implementation is usually completed in less than 20 seconds on a Sparc-10 machine.

5. IMAGE SEGMENTATION BY CROSS-VALIDATION

Bose and O'Sullivan (1997) recently proposed a segmentation procedure in which the "best" model is defined as the minimizer of

$$\mathrm{CV}_\lambda(\hat{k}, \hat{\boldsymbol{\Omega}}) = \mathrm{RSS}_{\hat{k}} + \lambda \hat{k}.$$

Compare this to the MDL criterion $\mathrm{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$ and observe that the penalty is the number of regions $\hat{k}$. Here λ is a "smoothing" parameter chosen by V-fold CV (i.e., leave out n/V observations each time). To approximate the minimizer of $\mathrm{CV}_\lambda(\hat{k}, \hat{\boldsymbol{\Omega}})$, Bose and O'Sullivan (1997) proposed a *recursive merging* algorithm.

I have empirically evaluated the performance of $\mathrm{CV}_\lambda(\hat{k}, \hat{\boldsymbol{\Omega}})$ with the "leave-one-out" choice of λ via a simulation study (Sec. 7). However, I did not use the same recursive merging algorithm suggested by Bose and O'Sullivan (1997) to approximate the minimizer of $\mathrm{CV}_\lambda(\hat{k}, \hat{\boldsymbol{\Omega}})$. (I was unable to obtain the relevant codes.) Instead, I used a greedy merging algorithm similar to the one discussed in Section 2, starting with oversegmentations of the images obtained by the SRG-based approach described in Section 4.1. I believe that the results obtained by the recursive and the greedy merging algorithms will be (slightly) different, but the general empirical conclusions to be reported in Section 7.2 will remain the same.

Besides MDL and CV, other statistical methods have also been applied to define "best" segmentations. These include

the Akaike information criterion (AIC) (Zhang and Modestino 1990) and Bayesian approaches (see, e.g., Johnson 1994; LaValle and Hutchinson 1995).

6. EVALUATION OF SEGMENTATION QUALITY

This section discusses the problem of automatic evaluation of segmentation quality.

6.1 Comparing Region Boundaries

Perhaps the simplest and the most widely used distance measure for comparing a true image f and a "fitted" image $\hat{\mathbf{f}}$ is the mean integrated squared error (MISE) between the grayvalues,

$$\text{MISE}_{\text{gray}} = \frac{1}{n}\sum_{i=1}^{n}(f_i - \hat{f}_i)^2,$$

which can also be used here to evaluate segmentation quality, as both the MDL-based and CV-based procedures also produce fitted images. However, it is widely known that $\text{MISE}_{\text{gray}}$ is not appropriate for image segmentation.

We can achieve better methods for evaluating segmentation quality by refocusing our target: to compare region boundaries but not grayvalues. This can be done by comparing a binary image containing true boundaries with another binary image containing "segmented boundaries." Because in my formulation boundaries are composed of horizontal and vertical "edges" between pixels, the actual boundaries cannot be exactly represented by binary images. To circumvent this, I propose comparing binary images consisting of *boundary pixels*. (A boundary pixel is a pixel having at least one of its four edges as part of a boundary.) Thus all that is needed now is a good distance measure for comparing such binary "boundary pixel images," and I suggest using Baddeley (1992)'s Δ_w^p as the distance measure. Appendix B gives a brief description of Δ_w^p.

6.2 Comparing Area and Perimeter Densities

I mentioned earlier that the original motivation for segmenting the aluminium grain image was an interest in measuring area and perimeter distributions, as such distributions may be related to aluminium product quality. Therefore, another meaningful method for evaluating segmentation results is to compare an estimated area (or perimeter) density of the segmented regions with an estimated area (or perimeter) density of the true regions. Hence in my simulation I also use discretized approximations of the following two quantities as quality measures:

$$\text{MISE}_{\text{area}} = \int \{p_{\text{area}}(x) - \hat{p}_{\text{area}}(x)\}^2\,dx$$

and

$$\text{MISE}_{\text{per}} = \int \{p_{\text{per}}(x) - \hat{p}_{\text{per}}(x)\}^2\,dx,$$

where p_{area} is a "true" area density function (estimated from the "true" segmentation) and $\hat{p}_{\text{area}}$ is an estimate of p_{area} calculated from a segmented image. In the simulation study $\hat{p}_{\text{area}}$ is obtained by kernel smoothing with a normal-scale choice of smoothing parameter (see, e.g., Wand and Jones 1995, chap. 3). The perimeter densities p_{per} and $\hat{p}_{\text{per}}$ are defined in a similar fashion.

Note that there are two problems with using $\text{MISE}_{\text{area}}$ (or MISE_{per}), however. First, $\text{MISE}_{\text{area}}$ (or MISE_{per}) is not a proper distance measure for comparing images, as two remarkably different images can have the same region area (or perimeter) density. Second, suitable statistical adjustments should be made to the raw measurements of those regions that are truncated (or censored) by image edges; otherwise, these raw measurements would induce bias in the density estimates. But it is not straightforward to calculate the required adjustment in the present situation, and also because this issue of adjusting truncated measurements is a relatively minor point in this work, I decided not to investigate this effect.

7. IMAGE SEGMENTATION SIMULATION STUDY

This section reports results of a simulation study which was conducted to evaluate the performance of the proposed MDL-based procedure and the CV-based procedure of Bose and O'Sullivan (1997).

7.1 Settings

Four test images (i.e., the true but unknown images) were used in the simulation study. They are all of dimension 256×256, and Figure 5 displays one of them. These test images are realizations generated from a model constructed in an attempt to simulate real aluminium grain images such as the one in Figure 1(a). Details were given in an earlier work (Lee 1999).

I used three different levels of signal-to-noise ratio (SNR): high SNR $= 8$, medium SNR $= 4$, and low SNR $= 2$. Here SNR is defined as the ratio of the variance of the test image to the noise variance; that is, $\text{SNR} = \text{var}(\mathbf{f})/\sigma^2$.

For each combination of test image and SNR (i.e., a total of $4 \times 3 = 12$ combinations), I generated 100 noisy observed images, and then segmented these observed images with the following steps. I first used the oversegmentation approach (based on SRG with regularly spaced seeds, where the seeds are squares of 5×5 pixels and are 20 pixels apart) discussed in Section 4.1 to obtain initial oversegmentations of the generated noisy images. I then applied the merging algorithms to find the (local) minima of the criteria $\text{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$ and

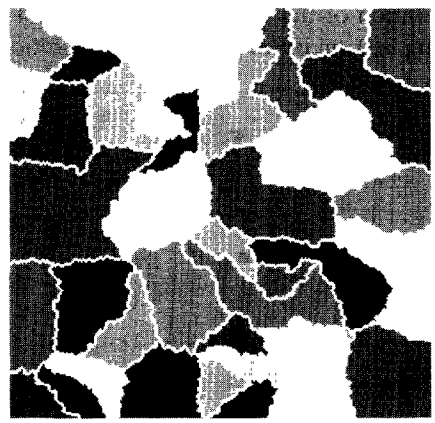


Figure 5. Test Image 1.

$CV_\lambda(\hat{k}, \hat{\Omega})$. That is, for each generated noisy image, I obtained two different segmentations (one MDL-based, the other CV-based) originating from the same initial oversegmentation.

Then for each segmentation result, I computed the values of Δ_w^p, $\text{MISE}_{\text{gray}}$, $\text{MISE}_{\text{area}}$ and MISE_{per}. Boxplots of of these values for test image 1, in a log scale, are displayed in Figure 6. Results for test images 2, 3, and 4 are similar and hence are omitted. To visually evaluate the segmentation results, I sorted, for each combination of test image 1 and the three SNRs, the Δ_w^p values of the MDL-based results in ascending order. Figure 7 displays the MDL-based results corresponding to the 50th smallest Δ_w^p values for test image 1. Figure 7 also displays are the CV-based results for the same images. Results for test images 2, 3, and 4 are similar and hence again are omitted.

7.2 Empirical Conclusions

Some obvious empirical conclusions can be drawn: The MDL-based procedure is clearly superior. The CV-based procedure always produces heavily oversegmented results, and in fact, the lower the SNR, the larger the number of segmented regions seems to be.

Figure 7 shows that the proposed MDL-based segmentation procedure performed well for both high and medium snrs. However, for low SNR, it produced "overmerged" re-

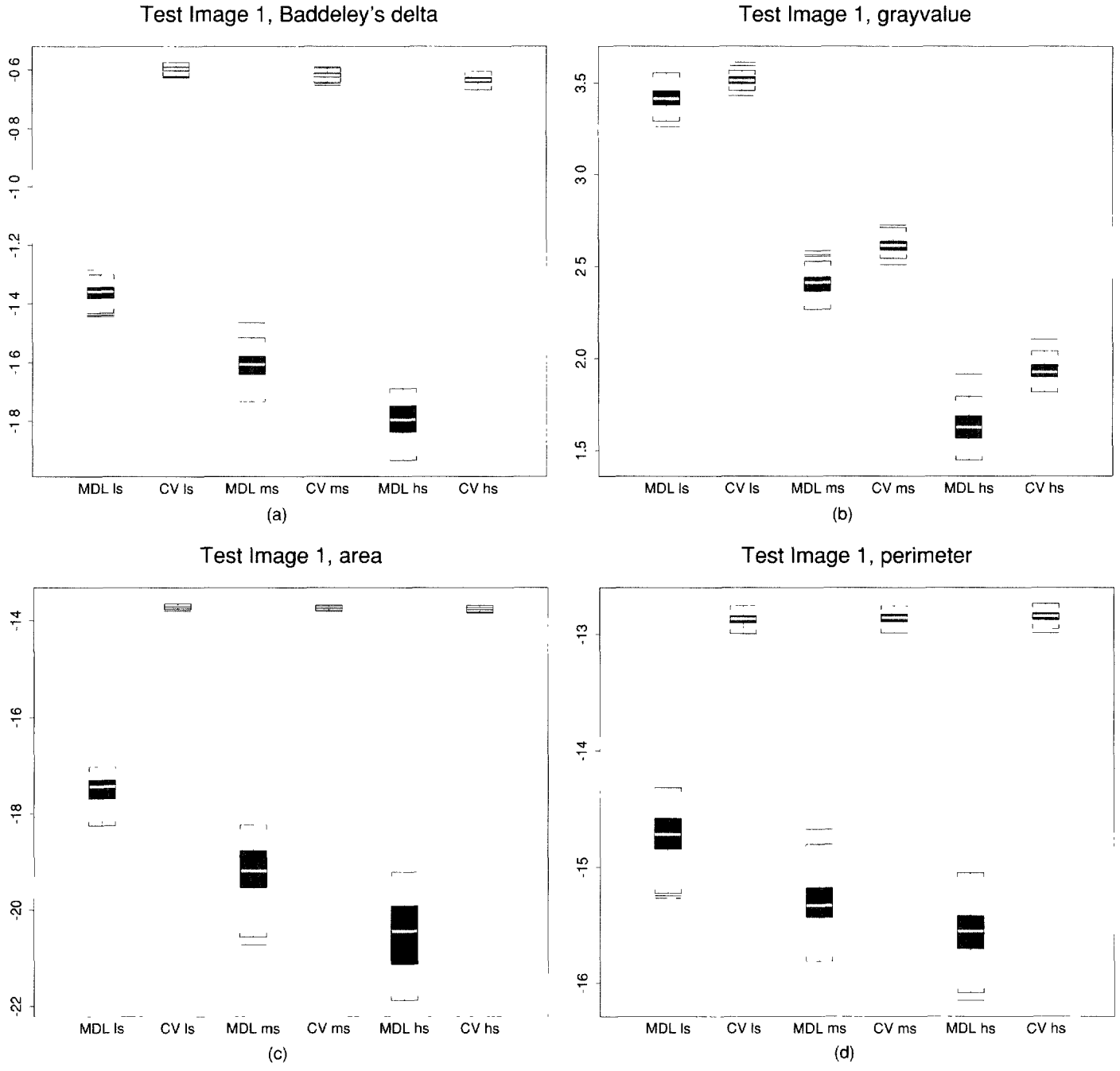


Figure 6. Boxplots of Δ_w^p, $MISE_{gray}$, $MISE_{area}$, and $MISE_{per}$, in a Log Scale, for the MDL-Based and CV-Based Segmentations for Test Image 1 Abbreviations used in the boxplots are ls = low SNR, ms = medium SNR, and hs = high SNR.

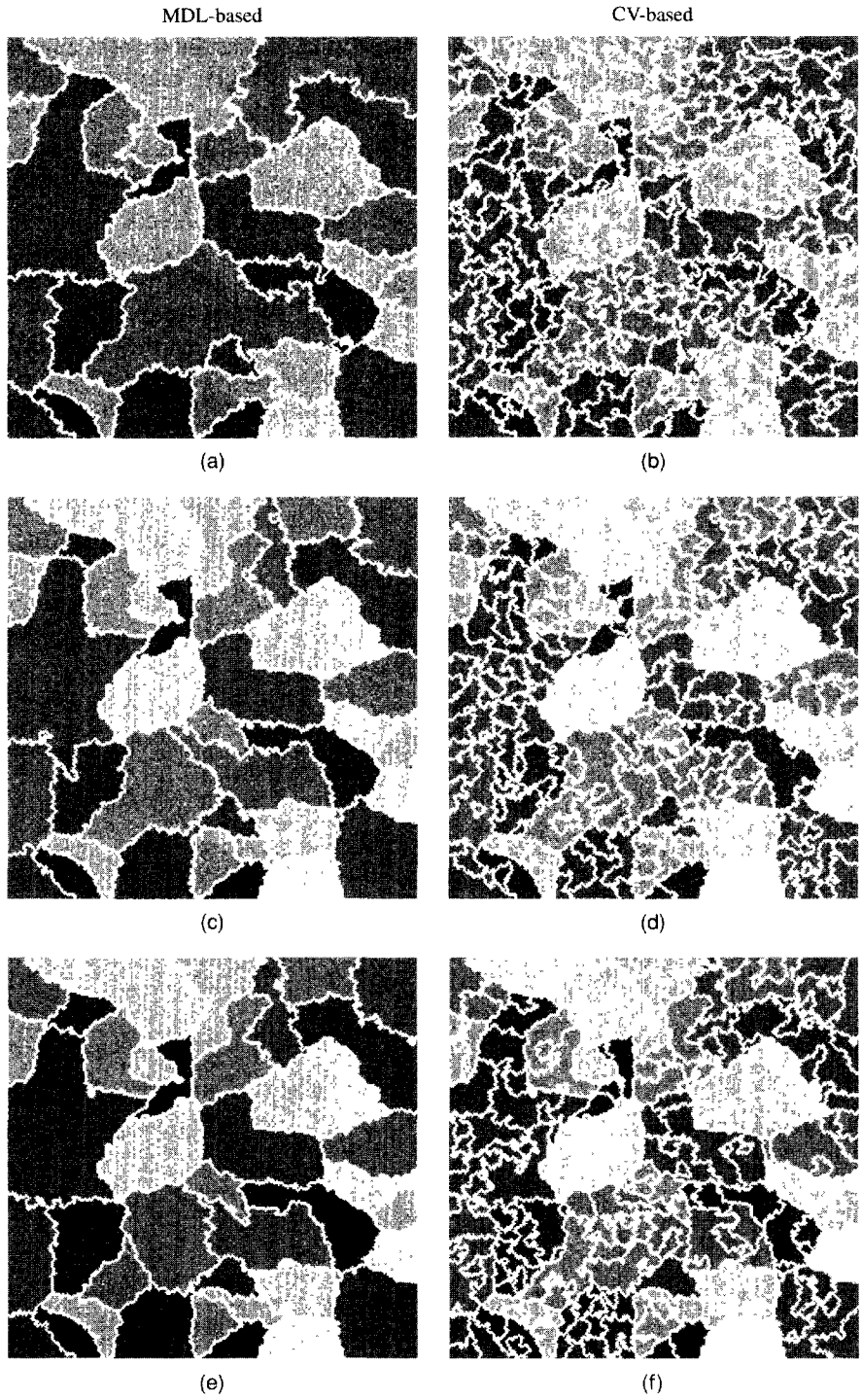


Figure 7. Segmentation Results for Test Image 1. Images in the left column are MDL-based results; those in the right column are CV-based results In the top row, the observed image corresponds to low SNR with segmented region boundaries (in white) overlaid. The middle row is similar to the top row but for medium SNR. The bottom row is similar to top row but for high SNR.

sults. This is not surprising, as the standard deviation of the noise was larger than some grayvalue differences between adjacent regions. As a result, these adjacent regions were merged together to form larger regions. Also, the lower the SNR, the rougher the segmented region boundaries.

8. REAL EXAMPLES REVISITED

Despite the fact the my piecewise constant model with additive independent noise may not be appropriate, I applied the MDL-based and the CV-based procedures to segment the two real images presented in Section 1. Results are displayed in Figure 8.

The MDL segmented result of the aluminium grain image seems to be a little oversegmented. I suspect that this is because the noise is not quite identical and independent, and I described methods for obtaining better results in my doctoral dissertation (Lee 1997). For the Landsat image, the MDL result is reasonable, but there are artifacts near

some of the small white "annuli" occurring at various points in the image. The CV-based procedure produced heavily oversegmented results.

9. OTHER ISSUES AND POSSIBLE EXTENSIONS

9.1 Comparison With Other Minimum Description Length–Based Procedures

Various MDL-based segmentation procedures have been proposed in the literature (see, e.g., Leclerc 1989; Kanungo, Dom, Niblack, Steele, and Sheinvald 1995; Zhu and Yuille 1996 and references therein). Here I compare my procedure to the MDL-based procedures closest to mine.

Kanungo et al. (1995) and Leclerc (1989) applied the MDL principle to the problem of image segmentation. Both allow the underlying true image to be piecewise polynomial and different regions to have different noise variances (i.e., spatial varying noise). They also claim that a maximal degree of 2 (i.e., quadratic) is sufficient for piecewise polynomial surfaces to approximate most images. But they do not provide automatic methods for choosing such a maximal degree, however.

The piecewise constant model that is being considered in this article can be seen as a special case of theirs. But when reduced to this special case, Kanungo et al. (1995) and Leclerc (1989) have different MDL criteria to (1). This is because I used different approximations and methods for encoding region boundaries. The use of different approximating and encoding methods is somewhat analogous to the case of adopting different priors when applying the Bayesian approach to tackling the same problem.

To approximate the minimum of her MDL criterion, Leclerc (1989) used a minimization procedure that is continuous in nature while Kanungo et al. (1995) used a stepwise region merging algorithm similar to ours (but did not suggest methods for obtaining oversegmentations). Leclerc's procedure is less likely to miss the global minimum but is more time consuming. Using different minimization procedures is analogous to using different sampling schemes to maximize the posterior density in Bayesian problems.

9.2 Non-Gaussian Noise

In certain circumstances the noise corrupting the true image is not Gaussian. If the distribution of such non-Gaussian noise is known, then an MDL criterion specific to this can be derived. Here I give an example.

In the synthetic aperture radar imaging context, it is known that Poisson or exponential noise models are more appropriate than Gaussian models (see, e.g., Derin, Kelly, Vézina, and Labitt 1990). Now suppose that the noise is Poisson and the true image $\mathbf{f}$ is a piecewise constant function. Denote the observed image by $\mathbf{y}$. A typical Poisson noise model is $y_i \sim$ independent Poisson$(f_i), i = 1, \ldots, n$,

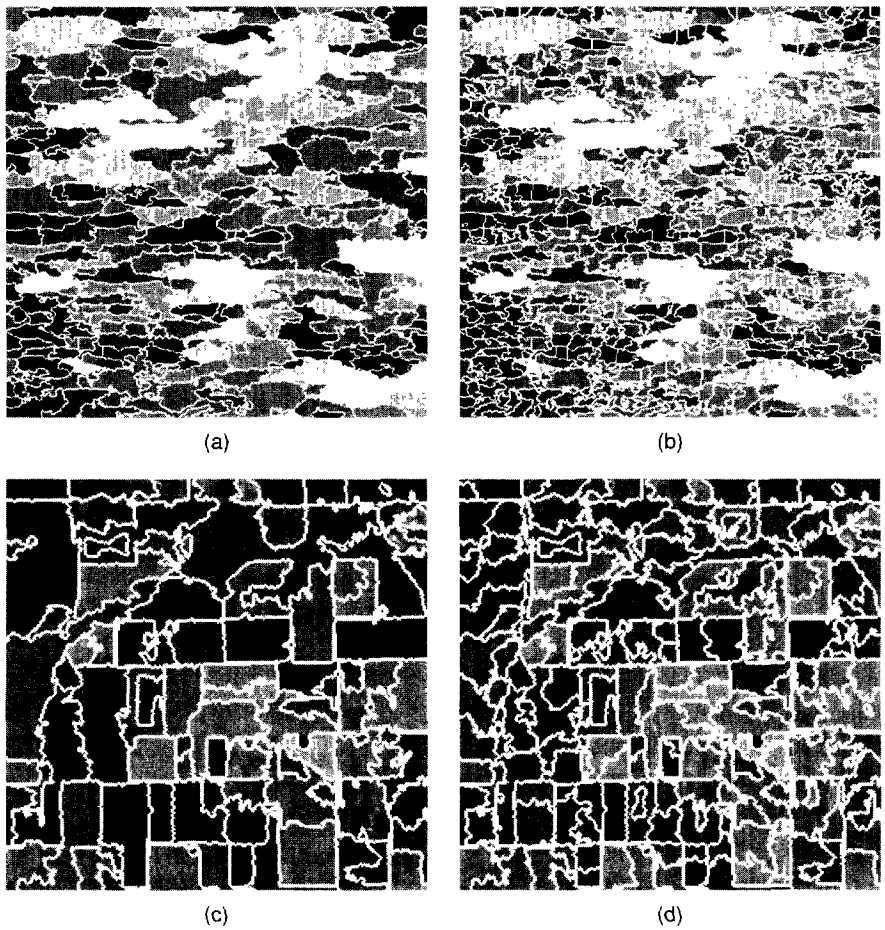


Figure 8. Segmentation Results of the Real Images Displayed in Figure 1. (a) and (c) MDL-based results; (b) and (d) CV-based results.

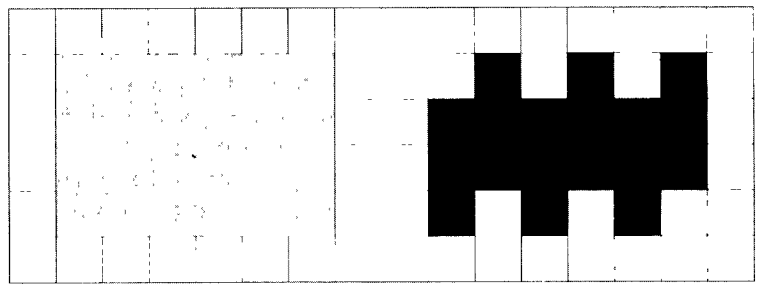


Figure 9. A Smooth Boundary Gray Region and a Rough Boundary Black Region.

where Poisson(x) means a Poisson random variable with mean x. For this Poisson model, estimates of f_i and μ_j are also given by (A.1). Now the negative of the log-likelihood of $\mathbf{y}$ given a fitted model is (ignoring a constant) $-\sum(y_i \log \hat{f}_i - \hat{f}_i)$. Because $\sum \hat{f}_i = \sum y_i$ is a constant and using similar arguments to those given earlier, one can derive the following MDL criterion for the Poisson noise model:

$$\hat{k} \log n + \frac{\log 3}{2} \sum_{j=1}^{\hat{k}} b_j + \frac{1}{2} \sum_{j=1}^{\hat{k}} \log a_j - \sum_{i=1}^{n} y_i \log \hat{f}_i.$$

9.3 Smoother Boundaries by Other Boundary Encoding Methods

One empirical observation obtained from the simulation study is that the lower the snr, the rougher the segmented region boundaries tend to be. If it is reasonable to assume a priori that the true region boundaries are smooth, then it is desirable to modify my MDL criterion to encourage smoother region boundaries.

One way to achieve this is to use other boundary encoding methods that generally produce shorter code lengths for smooth boundaries. For example, instead of using Freeman's chain code, one can use some sort of "run-length" type representation to represent region boundaries (see, e.g., Jain 1989, chap. 11). Consider the problem of representing the boundary of the gray region in Figure 9. Assume that the starting pixel edge is the bottom edge of the bottom leftmost pixel, and that the global travel direction is counterclockwise. Then Freeman's chain code gives: C_{F} = "FFFFFLFF-FLFFFFFLFFF", and a typical "run length"–type representation gives C_{R} = "F5L1F3L1F5L1F3", where the integers were used to indicate the number of repetitions of the previous direction. It is not hard to see C_{R} requires a shorter code length than C_{F}. On the other hand, such a run-length representation would produce longer code length than Freeman's chain code for the black region in Figure 9. Thus it is apparent that such a run-length representation "prefers" regions with smooth boundaries. It is also apparent that by using this run-length approach for boundary encoding (or any other boundary encoding methods that "prefer" smooth boundaries), one can obtain modified MDL criteria that encourage smoother horizontal and vertical boundaries.

10. CONCLUSIONS

In this article I presented an MDL-based segmentation procedure. Empirical results suggest that the procedure performs well for simulated data and is superior to an existing CV-based procedure. The MDL-based procedure is also applied to two real images, producing reasonable results. Extensions for non-Gaussian noise and smoother boundaries have been described. The procedure also can be extended to handle images corrupted by correlated noise and handle multiband images. (Details were given in Lee 1997, 1998.)

As a final remark, the reader may now recognize a hidden but important message of this article: Statisticians could play a more dominant role in tackling the very important, interesting, and challenging problem of image segmentation.

APPENDIX A: DERIVATION OF MDL($\hat{k}$, $\hat{\Omega}$)

This appendix provides the derivation of the MDL criterion MDL$(\hat{k}, \hat{\Omega})$ (1), which is an approximation of the code length $L(\mathbf{y})$ of the data $\mathbf{y}$. (Generally I use $L(z)$ to denote the code length of the object z.)

Recall that, using Freeman's chain code for boundary representation (see Sec. 3.3), a fitted model is completely characterized by (a) $\hat{k}$, (b) the coordinates of all *starting pixels*, (c) the direction *chains* of all regions, and (d) $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \ldots, \hat{\mu}_{\hat{k}})^T$. Thus one way to encode $\mathbf{y}$ is to first encode $\hat{k}$ and "starting pixels," then "chains" given $\hat{k}$ and "starting pixels," and then $\hat{\boldsymbol{\mu}}$ given $\hat{k}$, "starting pixels" and "chains," and finally $\mathbf{y}$ given $\hat{k}$, "starting pixels," "chains," and $\hat{\boldsymbol{\mu}}$. This leads to the following expression for the data code length $L(\mathbf{y})$:

$$\begin{aligned} L(\mathrm{y}) \;=\; & L(\hat{k}, \text{starting pixels}) + L(\text{chains}|\hat{k}, \text{starting pixels}) \\ & + L(\hat{\boldsymbol{\mu}}|\hat{k}, \text{starting pixels, chains}) \\ & + L(\mathbf{y}|\hat{k}, \text{starting pixels, chains}, \hat{\boldsymbol{\mu}}). \end{aligned}$$

I begin with $L(\hat{k}$, starting pixels). Because I use single indexing for pixel coordinate labeling, a compact way to specify all starting pixel coordinates is to specify their successive differences. That means that $\hat{k}$ positive integers with sum a little less than n can be used to specify all starting pixel coordinates. The code length for encoding $\hat{k}$ such positive integers ($\hat{k}$ is unknown), which is also $L(\hat{k}$, starting pixels), is given by the first two terms of equation (23) of Rissanen (1989):

$$L(\hat{k}, \text{starting pixels}) = L^*(n) + \log_2 \frac{(n+\hat{k})!}{n!(\hat{k}-1)!},$$

with $L^*(\cdot)$ defined by $L^*(n) = \log_2 c + \log_2 n + \log_2 \log_2 n + \ldots$, and the sum includes only positive terms and $c \approx 2.865$.

If b_j is the perimeter (in terms of number of pixel edges) of the jth region, then the code length for encoding the direction chain of the jth region is $L^*(b_j) + b_j \log_2 3$. The first term is required to ensure the prefix property (e.g., Rissanen 1989, sec.

2.2) of the codes, and the second term is used for storing the actual "directions." The appearance of $\log_2 3$ is due to the fact that, on average, a direction needs $\log_2 3$ bits to encode. Using the foregoing expression, one can deduce

$$L(\text{chains}|\hat{k}, \text{starting pixels}) = \frac{1}{2}\left[\sum_{j=1}^{\hat{k}} L^*(b_j) + \log_2 3 \sum_{j=1}^{\hat{k}} b_j\right].$$

The factor $\frac{1}{2}$ is necessary here, because otherwise all boundary edges will be encoded twice.

To encode the real numbers $\hat{\mu}_j$'s, I need to impose a prior $\pi(\hat{\boldsymbol{\mu}})$ on $\hat{\boldsymbol{\mu}}$ and specify a precision $\delta(\hat{\mu}_j)$ for each $\hat{\mu}_j$. Following Rissanen (1989, pp. 53–54), the code length $L(\hat{\boldsymbol{\mu}}|\hat{k}$, starting pixels, chains) for encoding $\hat{\mu}_j$'s conditional on $\hat{k}$, starting pixels and chains is

$$L(\hat{\boldsymbol{\mu}}|\hat{k}, \text{starting pixels, chains}) = -\log_2 \pi(\hat{\boldsymbol{\mu}}) - \sum_{j=1}^{\hat{k}} \log_2 \delta(\hat{\mu}_j).$$

The code length $L(\mathbf{y}|\hat{k}$, starting pixels, chains, $\hat{\boldsymbol{\mu}})$ of the data $\mathbf{y}$ conditioning on the fitted model is given by the negative of the log of the likelihood of $\mathbf{y}$ conditioning on the fitted model (see Rissanen 1989, pp. 54–55). In the present situation it simplifies to

$$L(\mathbf{y}|\hat{k}, \text{starting pixels, chains}, \hat{\boldsymbol{\mu}}) = \frac{n}{2}\log_2\left(\frac{\text{RSS}_{\hat{k}}}{n}\right) + C,$$

where C is a negligible term and $\text{RSS}_{\hat{k}} = \sum_{i=1}^{n}(y_i - \hat{f}_i)^2$ is the residual sum of squares with

$$\hat{f}_i = \sum_{j=1}^{\hat{k}} \hat{\boldsymbol{\mu}}_j I_{\{i \in \hat{r}_j\}}, \qquad \hat{\boldsymbol{\mu}}_j = \frac{1}{a_j}\sum_{s \in \hat{r}_j} y_s, \qquad i = 1, \ldots, n. \quad \text{(A.1)}$$

Here $i \in \hat{r}_j$ means "the ith pixel is in the jth region of a fitted model." Thus the overall code length $L(\mathbf{y})$ of the data $\mathbf{y}$ is

$$\begin{aligned} L(\mathbf{y}) &= L(\hat{k}, \text{starting pixels}) + L(\text{chains}|\hat{k}, \text{starting pixels}) \\ &\quad + L(\hat{\boldsymbol{\mu}}|\hat{k}, \text{starting pixels, chains}) \\ &\quad + L(\mathbf{y}|\hat{k}, \text{starting pixels, chains}, \hat{\boldsymbol{\mu}}) \\ &= L^*(n) + \log_2 \frac{(n+\hat{k})!}{n!(\hat{k}-1)!} \\ &\quad + \frac{1}{2}\left[\sum_{j=1}^{\hat{k}} L^*(b_j) + \log_2 3 \sum_{j=1}^{\hat{k}} b_j\right] \\ &\quad - \log_2 \pi(\hat{\boldsymbol{\mu}}) - \sum_{j=1}^{\hat{k}} \log_2 \delta(\hat{\mu}_j) + \frac{n}{2}\log_2\left(\frac{\text{RSS}_{\hat{k}}}{n}\right), \end{aligned}$$

which should be minimized by the "best" fitted model.

Direct minimization of the foregoing expression for $L(\mathbf{y})$ is practically infeasible, and a simpler approximation to $L(\mathbf{y})$ is needed. My approach for simplifying $L(\mathbf{y})$ is similar to that of Cameron, Hannan, and Speed (1995). If I assume that both n and $\hat{k}$ are large and $n >> \hat{k}$, use Stirling's formula to approximate factorials, and ignore constant and negligible terms, then the first two terms can be well approximated by $\hat{k}\log_2 n$. I ignore the term $\sum_{j=1}^{\hat{k}} L^*(b_j)$, as $b_j >> L^*(b_j)$ when b_j is large. I also ignore the term $-\log_2 \pi(\hat{\boldsymbol{\mu}})$, because $\pi(\hat{\boldsymbol{\mu}})$ can be well chosen to make $-\log_2 \pi(\hat{\boldsymbol{\mu}})$ small enough to be ignored (see, e.g., Rissanen 1989, p. 56). I apply the result of Rissanen (1989, pp. 55–56) to handle the term regarding the precisions $\delta(\hat{\mu}_j)$'s: Briefly, if $\hat{\mu}_j$ is the (conditional) maximum likelihood estimate of μ_j estimated from a_j data points and if a_j is large, then $\delta(\hat{\mu}_j)$ can be effectively encoded with $\frac{1}{2}\log_2 a_j$ bits. This means that we can replace the term $-\sum_{j=1}^{\hat{k}} \log_2 \delta(\hat{\mu}_j)$ in the foregoing expression for $L(\mathbf{y})$ by $1/2\sum_{j=1}^{\hat{k}} \log_2 a_j$. When all of these approximations are taken and all $\log_2$'s are replaced by the natural log, then the MDL criterion $\text{MDL}(\hat{k}, \hat{\boldsymbol{\Omega}})$ (1) is obtained.

APPENDIX B: A BINARY IMAGE DISTANCE MEASURE: BADDELEY'S Δ_w^p

This appendix gives the definition of Δ_w^p. (The reader is referred to Baddeley 1992 for a thorough discussion of the measure.) To follow Baddeley's notation, let X be a pixel grid with N pixels, and let $A \subset X$ and $B \subset X$ be the set of all "black pixels" of a true and a fitted binary image. For a pixel $x \in X$, define $d(x, A)$ as the smallest distance from x to A:

$$d(x, A) = \min_{a \in A} \rho(x, a),$$

where ρ is a distance function between two pixels and is taken as the Euclidean distance in this article. Baddeley defined the new distance measure as

$$\Delta_w^p(A, B) = \left[\frac{1}{N}\sum_{x \in X} |w(d(x, A)) - w(d(x, B))|^p\right]^{1/p}, \quad \text{(B.1)}$$

where $w(t) = \min(t, c)$ is a threshold function and p and c are parameters supplied by the user. (Other forms for $w(t)$ are also possible; see Baddeley 1992.) I followed Baddeley and set $p = 2$ and $c = 5$.

[Received February 1997. Revised May 1999.]

REFERENCES

Adams, R., and Bischof, L. (1994), "Seeded Region Growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 641–647.

Baddeley, A. J. (1992), "Errors in Binary Images and an L^p Version of the Hausdorff Metric," *Nieuw Archief voor Wiskunde*, 10, 157–183.

Beaulieu, J-M., and Goldberg, M. (1989), "Hierarchy in Picture Segmentation: A Stepwise Optimization Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 150–163.

Besl, P. J., and Jain, R. C. (1988), "Segmentation Through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, 167–192.

Bose, S., and O'Sullivan, F. (1997), "A Region-Based Image Segmentation Method for Multichannel Data," *Journal of the American Statistical Association*, 92, 92–106.

Cameron, M. A., Hannan, E. J., and Speed, T. P. (1995), "Estimating Spectra and Prediction Variance," unpublished manuscript.

Chang, Y-L., and Li, X. (1994), "Adaptive Image Region-Growing," *IEEE Transactions on Image Processing*, 3, 868–872.

Chen, S-Y, Lin, W-C., and Chen, C-T. (1991), "Split-and-Merge Image Segmentation Based on Localized Feature Analysis and Statistical Tests," *CVGIP: Graphical Models and Image Processing*, 53, 457–475.

Derin, H., Kelly, P. A., Vézina, G., and Labitt, S. G. (1990), "Modeling and Segmentation of Speckled Images Using Complex Data," *IEEE Transactions on Geoscience and Remote Sensing*, 28, 76–87.

Glasbey, C. A., and Horgan, G. W. (1995), *Image Analysis for the Biological Sciences*, New York: Wiley.

Haralick, R. M., and Shapiro, L. G. (1992), *Computer and Robot Vision*, Reading, MA: Addison-Wesley.

Jain, A. K. (1989), *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice-Hall.

Johnson, V. E. (1994), "A Model for Segmentation and Analysis of Noisy Images," *Journal of the American Statistical Association*, 89, 230–241.

Kanungo, T., Dom, B., Niblack, W., Steele, D., and Sheinvald, J. (1995), "MDL-Based Multi-Band Image Segmentation Using a Fast Region Merging Scheme," Technical Report RJ 9960 (87919), IBM Corp., Research Division.

LaValle, S. M., and Hutchinson, S. A. (1995), "A Bayesian Segmentation Methodology for Parametric Image Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 211–217.

Leclerc, Y. G. (1989), "Constructing Simple Stable Descriptions for Image Partitioning," *International Journal of Computer Vision*, 3, 73–102.

Lee, T. C. M. (1997), "Some Models and Methods in Image Segmentation," doctoral dissertation, Macquarie University, Sydney, Australia.

——— (1998), "Segmenting Images Corrupted by Correlated Noise," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 481–492.

——— (1999), "A Stochastic Tessellation for Modelling and Simulating Colour Aluminium Grain Images," *Journal of Microscopy*, 193, 109–126.

Rissanen, J. (1989), *Stochastic Complexity in Statistical Inquiry*, Singapore: World Scientific.

Wand, M. P., and Jones, M. C. (1995), *Kernel Smoothing*, London: Chapman and Hall.

Zhang, J., and Modestino, J. W. (1990), "A Model-Fitting Approach to Cluster Validation With Application to Stochastic Model-Based Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 1009–1017.

Zhu, S. C., and Yuille, A. (1996), "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, 884–900.