A Fast Algorithm for MDL-Based Multi-Band Image Segmentation

Tapas Kanungo* Byron Dom Wayne Niblack David Steele
IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120

Abstract

We consider the problem of image segmentation and describe an algorithm that is based on the Minimum Description Length (MDL) principle, is fast, is applicable to multi-band images, and guarantees closed regions. We construct an objective function that, when minimized, yields a partitioning of the image into regions where the pixel values in each band of each region are described by a polynomial surface plus noise. The polynomial orders and their coefficients are determined by the algorithm. The minimization is difficult because (1) it involves a search over a very large space and (2) there is extensive computation required at each stage of the search. To address the first of these problems we use a region-merging minimization algorithm. To address the second we use an incremental polynomial regression that uses computations from the previous stage to compute results in the current stage, resulting in a significant speed up over the non-incremental technique. The segmentation result obtained is suboptimal in general but of high quality. Results on real images are shown.

1 Introduction

1.1 The General Image Segmentation Problem

This paper[1] describes a solution to the problem of unsupervised multiband image segmentation. This is an extension of a previous algorithm[22]. More precisely, the problem we address is the following. We are given an image in which the ith pixel has associated with it two vectors: $\mathbf{y}_i \in \Re^d$ and $\mathbf{x}_i \in \mathbf{Z}^q$. The components of $\mathbf{x}_i$ are the pixel coordinates in a discrete, q-dimensional space. The components of $\mathbf{y}_i$ represent the measured intensity levels (greylevels) in d different "bands". These may be spectral bands (e.g. "red", "green" and "blue"), different imaging modalities (e.g. an optical image and a range image), "features" (using pattern recognition parlance) computed from the greylevels of the pixels in some neighborhood of pixel i, [2] and so on. We assume that the image represents a real scene consisting of objects, regions, surfaces, etc. Our task is to divide this image into a number of non-overlapping regions whose union is the entire image. The goal is for these regions to correspond to actual regions, objects, surfaces, etc. in the real scene. We want the regions produced by the algorithm to be homogeneous in some sense. There may be a precise measure of homogeneity corresponding to a particular application or we may want the grouping into regions to be similar to one that would be produced by a human given the same task. Our approach is to propose a precisely defined problem whose solution is, in many cases of interest, consistent with this more general but somewhat imprecise definition. In doing so, we will make certain assumptions about the images being segmented. In practice these frequently may not hold of course, but, taking a pragmatic view, we will judge the algorithm by how well it does in segmenting images of real scenes.

The solution to the segmentation problem that we present uses the Minimum Description Length Principle (MDL) to obtain a complexity-based objective function. The originality of our approach is in its combination of generality (multi-band, high-order polynomial surfaces), speed, and this fixed (no adjustable thresholds) MDL-based objective function.

1.2 Related Work

Before beginning a detailed description of our approach, we will briefly survey related work. We will review here only closely related work. For a more complete survey of image segmentation techniques see [8]. The MDL criterion was first used for the problem of image segmentation by Leclerc[12] where a graylevel image was partitioned into regions, with a two-dimensional polynomial model defined on each region. A continuation minimization procedure led to an algorithm for finding the regions and polynomials. This work is the most closely related to ours. There are, however, important differences between his approach and ours: (1) we use a region-merging-based optimization procedure whereas Leclerc uses a continuation scheme. Although the continuation approach is less likely to result in a suboptimal (local minimum) solution than ours, it is much slower and if not allowed to run to convergence, may leave boundary fragments that aren't closed. Our approach never does this; (2) ours treats multiple band images; (3) we explicitly count the cost of the encoding parameters in our MDL formulation; (4) ours is implemented by a fast, incremental computation; (5) Leclerc's algorithm, through its application of the continuation

*Tapas Kanungo is now at: Intelligent Systems Laboratory; Department of Electrical Engineering, FT-10 University of Washington; Seattle, WA 98195; (tapas@ee.washington.edu)

[1] See [9] for an expanded version.

[2] Some examples are the local mean, median, maximum, gradient magnitude/direction, Laplacian. More complicated measures are, of course, also possible.

approach has a "relaxation" flavor where pixels determine their new (next iteration) parameter values based on the values of those parameters and the grey-values of neighboring pixels (and their own) for the current iteration. Our approach is region based, however, always maintaining and merging regions based on their statistics and boundaries. In more recent work[13] Leclerc applied the same approach to region grouping. Our approach lends itself naturally to that problem as well; see [22].

Other authors have also used MDL for image segmentation. Keeler[10] describes a method in which he segments an image by encoding the topology of the segments (for which he has an efficient encoding), their specific boundaries, and the pixel values in each segment as a noise-corrupted constant grey level. The segmentation is the one for which the encoding length of the topology, boundaries, means, and deviations from means is minimum. Fua and Hanson[7] use MDL for a model-based image segmentation. They use geometric constraints on object boundaries (e.g. they are straight lines), allow certain outlying pixels to be excluded to account for shadows, etc., and model only the "objects" in a scene, not the background. Pentland, Darrel and Sclaroff have applied MDL in image segmentation[17, 6]. They use part-based models combined with an optimization algorithm that uses a modified Hopfield-Tank network and a continuation scheme.

MDL has been used to achieve segmentation by simple feature-space clustering. See for example the work of Wallace and Kanade[23]. Zhang and Modestino[24] use the AIC (Akaike's information criterion [2]), an information-theoretic criterion that is an alternative to MDL, for image segmentation, also by simple feature-space clustering.

Keren et. al.[11] apply MDL to the problem of 1D waveform segmentation and experiment with extension of the technique to images by operating on 1D projections of those images.

Besl and Jain [5] also addressed image segmentation using polynomial surface fitting, but the criterion function uses a user-specified threshold for acceptable noise variance and does not account for the model complexity as the MDL principle does. Another approach that uses a similar image model (polynomial surfaces plus Gaussian noise) is applied to 2D images in [14] and 3D surfaces in [15]. This work is also not based on MDL however, and uses a different optimization algorithm.

1.3 The Problem We Solve

Similar to many of these approaches, to solve this problem we will use the common general procedure of formulating an *objective function*, whose global minimum (we assume) corresponds to the best segmentation of the image, then devising an *optimization procedure* that attempts to find this minimum. In formulating this objective function we assume that the images to be analyzed come from a certain stochastic process, characterized by a family of stochastic models (probability distributions, $p(\mathbf{y}_i)$). The model we assume for this process consists of an ideal partitioning (the segmentation we seek) of the image into regions, $\{\omega_j\}$ (Denote this segmentation by $\Omega = \{\omega_j\}$.) and a separate probability density $p(\mathbf{Y}_j|\beta_j)$ for each region, where $\mathbf{Y}_j$ represents the collection of $\mathbf{y}_i$'s within region j and β_j is a vector of parameters characterizing the distribution. We will use $\beta \triangleq \{\beta_j\}$ to denote the collection of all the parameters for all the regions in Ω. More specifically, in the work described here we will assume that the pixel values of the regions of the image can be described by polynomial (in spatial coordinates) greyscale surfaces (one per band) to which "white" (spatially uncorrelated) "noise" has been added. We further assume that this noise is Gaussian distributed with (in general) a non-diagonal covariance matrix i.e. there can be correlation among the bands[3]. Let $\mu_j(\mathbf{x})$ be a d-dimensional vector-valued function whose components are the values of the underlying polynomial surfaces mentioned above (these can be treated as the spatially dependent mean of the Gaussian distribution), and Σ_j is the covariance matrix for the region ω_j. Note that for this model β consists of the polynomial coefficients of the greyscale surfaces and the components of the covariance matrices. Notice also that in this description the region boundaries are composed of the "cracks" between the pixels. In many images, for example those where the optical resolution of the imaging system (lenses, etc.), expressed in units of length, is larger than the pixel size this may seem like an unjustified assumption. Our use of polynomial models allows the existence of "edge" regions in such cases, however.

A problem with performing *maximum likelihood* (ML) estimation in a case such as this is that there is no bound on the complexity of the model, M, and the more complex it is made, the better the fit obtained until the ridiculous limit of every pixel being a separate region is reached. We say that such problems are "ill-posed" or "under-constrained". To correct such problems some way of "regularizing" them must be found. The approach we have chosen to address this problem is to apply the Minimum Description Length Principle (MDL)[18, 20]. In this approach the objective function to be minimized is the description length of the data in a suitable "language." We choose MDL for two reasons: (1) It has a strong fundamental grounding, being based on information-theoretic arguments that can be viewed as a formalization of the physicist's *Ockham's razor:* the simplest model explaining the observations is the best [4]; and (2) It results in an objective function with no arbitrary thresholds. To formalize this, the model is used to *encode* (in the sense of data compression) the data in such a way that it can be decoded by a decoder that "knows" only about the model class (the image size, the number of bands and the fact that polynomial

[3] This inter-band correlation will be especially strong in cases where, for example, the "noise" actually corresponds to material texture in the scene

[4] Attributed to William of Ockham (1285-1349)

Gaussian models will be used). The model that gives the shortest description length in bits is then chosen as optimum.

There are different ways to reduce this general methodology to an algorithm that can be applied to a given problem (See [20]). The one we use is conceptually straightforward and typically the easiest computationally. It is based on a two-part encoding, where one part consists of an encoding of the model and the other consists of an encoding of the data using the model. Thus the codelength we seek to minimize is:

$$L(\mathbf{Y}, M) = L(M) + L(\mathbf{Y}|M), \quad (1)$$

where, $L(\ldots)$ denotes codelength. This codelength *is* our objective function. In the following section we will derive detailed expressions for the terms in this equation. If the set of possible models were discrete (countable) and we had a prior probability, $P(M)$ on those models, we could let let $L(\mathbf{Y}|M) = -\log P(\mathbf{Y}|M)$. In this case minimizing equation (1) is equivalent to performing Bayesian *maximum a posteriori* (MAP) estimation. If the set of possible models is not countable (the more usual case, which is also the case in this work), however, the situation is more complicated.

2 The Objective Function

As discussed in the introduction, our objective function will be divided into two parts: the codelength of the model, $L(M)$, and the codelength of the data given the model (i.e. encoded using the model), $L(\mathbf{Y}|M)$. In our approach the specification of the model divides naturally into two components, $M = \{\Omega, \boldsymbol{\beta}\}$: the segmentation, Ω, and the distribution parameters, $\boldsymbol{\beta}$. Thus our total code length (equation 1) may be written as:

$$L(\mathbf{Y}, \Omega, \boldsymbol{\beta}) = L(\Omega) + L(\boldsymbol{\beta}|\Omega) + L(\mathbf{Y}|\Omega, \boldsymbol{\beta}) \quad (2)$$

We begin by deriving an expression for $L(\Omega)$.

2.1 Encoding Region Boundaries: $L(\Omega)$

We can encode the boundaries by encoding a graph whose nodes represent the boundaries' intersections, and whose edges represent the boundary branches lying between those intersections. To make the boundaries reconstructable from such a graph, we choose one node from each connected component of this graph to be a reference node. To describe a given connected component we start by specifying the location of the reference node, followed by the number of branches from that node, followed by length of the first boundary branch (corresponding to a graph edge), followed by a chain code representing its path along the rectangular grid between the pixels (this chain-code description was also used in [12].).

Each element of the chain-code description of a branch represents the direction of the next step in the chain. Since the number of possible directions is 3, i.e. the number of adjacent grid points (excluding the last visited grid point), the number of bits required for the chain code is $l_i \log 3$. To encode the length of the boundary segment we use Rissanen's "universal prior" for integers[19], which gives a code length of $L^0(l_i) = \log^*(l_i) + \log(2.865064)$, where $\log^*(x) = \log x + \log\log x + \log\log\log x \cdots$ up to all positive terms. Thus, associated with arc i, whose length is l_i, is an encoding cost of $L^0(l_i) + l_i \log 3$.

When the regions are large, the bulk of the resulting codelength will be the length of description of the branches, so that we can approximate the description length of the boundaries (neglecting the description length of the graph) by $\sum_{\text{all branches}}[l_i \log 3 + L^0(l_i)]$, yielding: $L(\Omega) \approx \sum_i (l_i \log 3 + L^0(l_i))$.

The scheme we have chosen favors segmentations with shorter total boundary length for a given image size. This means it favors a small number of regions with smooth boundaries. This also seems to be a reasonable measure of complexity, though it does differentiate between some cases where the complexity difference is not clear such as charging a heavier penalty for a large square than for a small one.

2.2 Encoding the Parameters: $L(\boldsymbol{\beta}|\Omega)$

For the coding cost of the real-valued parameters, $\boldsymbol{\beta}$, we use the expression derived by Rissanen in his optimal-precision analysis[19]. For encoding K independent real-valued parameters characterizing a distribution used to describe/encode n data points the codelength he derives is: $(K/2)\log n$. Rissanen derives this expression for the encoding cost of real-valued parameters by optimizing the precision to which they are encoded. Encoding them to infinite precision would require an infinite number of bits and there is a tradeoff point at which the gain (i.e. decrease) in the codelength of the data due to increasing the precision of the parameters is exactly offset by increased codelength for the parameters. This codelength corresponds to that optimal precision, but is an asymptotic form for large n. During the writing of this paper we have become aware of recent results in this area[16, 21]. These results derive better expressions valid for small n. In future work we will utilize these new results.

Applying this result in our case we will have one such term for each region, which results in a total parameter codelength of:

$$L(\boldsymbol{\beta}|\Omega) = \frac{1}{2}\sum_j K_{\beta_j} \log n_j, \quad (3)$$

where K_{β_j} is the number of free parameters in $\boldsymbol{\beta}_j$ and n_j is the number of pixels in region j. For our model we have:

$$K_{\beta_j} = \frac{d(d+1)}{2} + dm_j, \quad (4)$$

where m_j is the number of polynomial coefficients per

band in region j. The first term on the right hand side of equation (4) is the number of free parameters in the covariance matrix, $\mathbf{\Sigma}_j$. The second term is the number of polynomial coefficients in the spatially varying mean vector, $\boldsymbol{\mu}_j(\mathbf{x})$, which is equal to the number of terms in Θ. For maximum polynomial degree k_j and a two-dimensional ($q = 2$) image $m_j = (k_j+1)(k_j+2)/2$. Substituting into equation (4) yields, for a two dimensional image:

$$K_{\beta_j} = \frac{d}{2}[(d+1) + (k_j+1)(k_j+2)]. \quad (5)$$

2.3 Encoding the Residuals: $L(\mathbf{Y}|\mathbf{\Omega}, \boldsymbol{\beta})$

In this section we will describe the encoding of the residuals (the data given the model) and derive an expression for $L(\mathbf{Y}|M) = L(\mathbf{Y}|\mathbf{\Omega}, \boldsymbol{\beta})$. Since our model includes polynomial surfaces fit to the greyvalues in each region, we might think of this step as that of encoding the residuals between the polynomial surface and the actual data.

Now let $\mathbf{Y} = [\mathbf{y}_1\mathbf{y}_2 \ldots \mathbf{y}_n]^t$ (the total collection of pixel values for the entire image), let $\mathbf{Y}_j$ denote those belonging to the jth region and let n_j represent the number of pixels in region j. Bear in mind that both n_j and $\mathbf{Y}_j$ are functions of the image partitioning, $\mathbf{\Omega}$. Let $p(\mathbf{y}|\boldsymbol{\beta}_j)$ be conditional distribution of a sample $\mathbf{y}$ belonging to the jth region which is characterized by the parameter vector $\boldsymbol{\beta}_j$ and let the parameter set $\boldsymbol{\beta} \triangleq \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_J\}$, where J is the total number of regions in $\mathbf{\Omega}$. Then, the conditional distribution $p(\mathbf{Y} \mid \mathbf{\Omega}, \boldsymbol{\beta})$ is obtained by forming a product of the individual conditional distributions for all the regions in $\mathbf{\Omega}$.

$$p(\mathbf{Y} \mid \mathbf{\Omega}, \boldsymbol{\beta}) = \prod_j p(\mathbf{Y}_j \mid \boldsymbol{\beta}_j) \quad (6)$$

From Shannon's theorems (see [1]) we know that, when such a distribution is known, the shortest codelength for $\mathbf{Y}$ is given by $L(\mathbf{Y}|M) = -\log p(\mathbf{Y}|\mathbf{\Omega}, \boldsymbol{\beta}) = \sum_j -\log p(\mathbf{Y}_j \mid \boldsymbol{\beta}_j)$, where the logarithms are base-two.

We now derive an expression for this codelength using the specific assumptions of our model. We use the assumed Gaussian distributions but in order to use these, we need an expression for $\boldsymbol{\mu}_j(\mathbf{x})$, which is a vector-valued function whose components are polynomial greyvalue surfaces of the form:

$$\mu_{jl}(\mathbf{x}) = \sum_{k=1}^{m} \theta_{jlk}\phi_k(\mathbf{x}) \quad (7)$$

where μ_{jl} is the l^{th} component of the vector $\boldsymbol{\mu}_j$ and θ_{jlk} is the scalar coefficient for the j^{th} region, the l^{th} band and the kth polynomial basis function. The basis functions $\{\phi_k(\mathbf{x})\}$ are products of various powers of the components of $\mathbf{x}$. (i.e. the two image spatial coordinates). In matrix form this may be written as: $\boldsymbol{\mu}_j = \mathbf{\Phi}_j\mathbf{\Theta}_j$ where $\boldsymbol{\mu}_j$ is an $n_j \times d$ matrix of the fitted polynomial surface values (μ values); one for each of the d bands for each of the n_j points in region ω_j. Also, $\mathbf{\Phi}_j$ is an $n_j \times m$ matrix of basis function values; one for each of the m basis functions for each of the n_j points. The $n_j \times m$ matrix of regression coefficients is represented by $\mathbf{\Theta}_j$. Then, using these definitions, we may rewrite the terms on the right side of Equation (6) as:

$$p(\mathbf{Y}_j \mid \boldsymbol{\beta}_j) = (2\pi)^{-dn_j/2}|\mathbf{\Sigma}_j|^{-n_j/2} \times \exp\left[-\frac{n_j}{2}\text{trace}\left\{\mathbf{\Sigma}_j^{-1}S_j\right\}\right] \quad (8)$$

where $|\ldots|$ denotes the determinant and S_j is the *sample* covariance matrix defined by: $S_j \triangleq \frac{1}{n_j}(\mathbf{Y}_j - \mathbf{\Phi}_j\mathbf{\Theta}_j)(\mathbf{Y}_j - \mathbf{\Phi}_j\mathbf{\Theta}_j)^t$. See [9] for a derivation.

Using the results presented thus far we can write our objective function as follows.

$$L(\mathbf{Y}, \mathbf{\Omega}, \boldsymbol{\beta}) = \sum_i (l_i \log 3 + L^0(l_i)) + \sum_j \frac{K_{\beta_j}}{2}\log n_j + \sum_j \frac{n_j}{2}\left[d\log(2\pi) + \log|\mathbf{\Sigma}_j| + \text{trace}\left\{\mathbf{\Sigma}_j^{-1}S_j\right\}\right], \quad (9)$$

where $\sum_i$ is a sum over all boundary segments and $\sum_j$ is a sum over all regions (not to be confused with the covariance matrix $\mathbf{\Sigma}_j$). The three summations correspond to $L(\mathbf{\Omega})$, $L(\boldsymbol{\beta}|\mathbf{\Omega})$ and $L(\mathbf{Y}|\mathbf{\Omega}, \boldsymbol{\beta})$ from left to right in that order.

Since $\mathbf{Y}$ is fixed, we can think of this as an objective function that must be minimized over $\mathbf{\Omega}$ and $\boldsymbol{\beta}$. Fortunately, part of this minimization can be performed analytically. In fact, for a given $\mathbf{\Omega}$ all the real-valued components of $\boldsymbol{\beta}$ have analytical expressions. For example, for Gaussian distributions the ML estimate (which is also minimum-codelength) for $\mathbf{\Sigma}_j$ is $\hat{\mathbf{\Sigma}}_j = S_j$. Using this result gives: $\text{trace}\left\{\hat{\mathbf{\Sigma}}_j^{-1}S_j\right\} = d$. The remaining components of $\boldsymbol{\beta}$ are the polynomial coefficients, $\{\mathbf{\Theta}_j\}$, which don't appear explicitly in Equation (9), but are required to compute S_j. Expressions for these are derived in the following section.

Further simplifying Equation (9) yields the following objective function that can be minimized over all $\mathbf{\Omega}$. We use the notation, $\mathcal{L}(\mathbf{\Omega})$ (i.e. no functional dependence on $\mathbf{Y}$ and $\boldsymbol{\beta}$) to emphasize the point that during the minimization process, the data, $\mathbf{Y}$, are fixed and the parameters, $\boldsymbol{\beta}$, have analytical expressions in terms of $\mathbf{Y}$ that would appear in an expanded expression for S_j. These are derived in the following

section.

$$\mathcal{L}(\Omega) = \frac{n}{2}d(1+\log 2\pi) + \sum_i [l_i \log 3 + L^0(l_i)] + \frac{1}{2}\sum_j [n_j \log|S_j| + K_{\beta_j} \log n_j] \quad (10)$$

Evaluating this expression (Equation (10) requires computing the sample covariance matrices, $\{S_j\}$.

3 The Regression Problem

In the previous section we obtained an expression for our objective function, ($\mathcal{L}(\Omega)$;Equation (10)), which we would like to minimize over all Ω. The sample covariance matrices, $\{S_j\}$ appear in this equation and their calculation involves the problem of fitting multivariate polynomial functions (surfaces) to discrete multivariate data (i.e. $\mathbf{Y}$). In fact they (the $\mathbf{S}_j$) partially characterize the statistics of the deviations of the data from these surfaces. In this section we treat the general problem of fitting multivariate polynomial functions (surfaces) of the form $f : \mathbf{Z}^q \rightarrow \mathbf{Z}^d$, where $\mathbf{Z}$ is the set of integers. In the case of one band, 2-D grayscale images, $q = 2$ and $d = 1$; if the number of band is two, $d = 2$. A discussion on multivariate regression relevant to MDL can be found in [20].

The multivariate regression data model can be written as

$$[\mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_n]^t = \Phi\cdot[\theta_1\theta_2\cdots\theta_d]+[\psi_1\psi_2\cdots\psi_d] , \quad (11)$$

where the $\mathbf{y}_i$ are $d \times 1$ vectors representing the gray values in the d bands at the i^{th} pixel, θ_i are $m \times 1$ vector of regression coefficients for the i^{th} band, and ψ_i are $n \times 1$ vector of Gaussian noise values in the i^{th} band and distributed as $N(0, \sigma^2 I)$, (I is an $n \times n$ identity matrix and Φ is an $n \times m$ matrix). We can write the above equation in a more compact form as

$$\mathbf{Y} = \Phi \cdot \Theta + \Psi \quad (12)$$

where $\mathbf{Y}$ and Ψ are $n \times d$ matrices, Θ is a $m \times d$ matrix and Φ is a $n \times m$ matrix.

The multivariate regression problem, then, is to find the $\hat{\Theta}$ that minimizes the sum of squared residuals

$$\epsilon^2 = \text{trace}\left\{(\mathbf{Y} - \Phi\cdot\Theta)^t(\mathbf{Y} - \Phi\cdot\Theta)\right\} . \quad (13)$$

The above operator in this case is essentially the sum of squares of all entries of the error matrix $\epsilon = (\mathbf{Y} - \Phi \cdot \Theta)$. The solution to the minimization problem is (see [9]):

$$\hat{\Theta} = [\Phi^t\Phi]^{-1}\Phi^t\mathbf{Y} . \quad (14)$$

Same solution can also be obtained by solving the system of equations

$$\Phi^t\mathbf{Y} = \Phi^t\Phi\hat{\Theta} . \quad (15)$$

Now the sample covariance, S, is given by

$$n \cdot S = [\mathbf{Y} - \Phi\hat{\Theta}]^t[\mathbf{Y} - \Phi\hat{\Theta}] , \quad (16)$$

which can be shown [9] to be equal to,

$$n \cdot S = \mathbf{Y}^t\mathbf{Y} - \hat{\Theta}^t[\Phi^t\mathbf{Y}] . \quad (17)$$

3.1 The Incremental Regression Problem

Because we seek only the image segmentation Ω, we wouldn't need to compute the regression coefficients, Θ (They don't appear explicitly in Equation (24).) if it weren't for the fact that they are required to compute the covariance matrix estimates $\{\hat{\Sigma}_j\}$. For this reason, anything that can be done to improve the efficiency of their calculation and that of the $\{\hat{\Sigma}_j\}$ will be valuable. In this section we derive *incremental* formulas for computing these polynomial regression coefficients and the covariance matrix of a new merged region from those of the two individual regions merged without having to perform an explicit regression on the data of the merged region.

Consider the following two independent multivariate regressions:

$$\mathbf{Y}_1 = \Phi_1 \cdot \Theta_1 + \Psi_1 \quad (18)$$
$$\mathbf{Y}_2 = \Phi_2 \cdot \Theta_2 + \Psi_2 , \quad (19)$$

where $\mathbf{Y}_i$ is a $n_i \times d$ data vector, Φ_i is a $n_i \times m$ regression matrix, Ψ_i is a $n_i \times d$ noise matrix, Θ is a $m \times d$ regression coefficient matrix.

Assume that the optimal $\hat{\Theta}_1$ and $\hat{\Theta}_2$ have already been computed. Now consider the following "concatenated" problem:

$$\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} \cdot \Theta + \Psi . \quad (20)$$

Let $\mathbf{Y} = [\mathbf{Y}_1^t \quad \mathbf{Y}_2^t]^t$ and let $\Phi = [\Phi_1^t \quad \Phi_2^t]^t$. Then the above equation can be written as $\mathbf{Y} = \Phi\Theta + \Psi$. The problem: find a computationally efficient method for computing $\hat{\Theta}$ and $(n_1 + n_2)S = [\mathbf{Y} - \Phi\hat{\Theta}]^t[\mathbf{Y} - \Phi\hat{\Theta}]$ for the concatenated system.

Incremental Computation of $\hat{\Theta}$: Matrix inverse computations can be unstable and it is better to compute $\hat{\Theta}$ by solving the linear system of equations $\Phi^t\mathbf{Y} = \Phi^t\Phi\Theta$. Expanding Φ and $\mathbf{Y}$ we get

$$\Phi^t\mathbf{Y} = [\Phi_1^t\mathbf{Y}_1 + \Phi_2^t\mathbf{Y}_2] \quad (21)$$

and,

$$\Phi^t\Phi = [\Phi_1^t\Phi_1 + \Phi_2^t\Phi_2] . \quad (22)$$

Notice that the matrix products $\mathbf{\Phi}_i^t\mathbf{Y}_i$ and $\mathbf{\Phi}_i^t\mathbf{\Phi}_i$ are available since they must have been computed for the individual systems. Moreover, although the matrices $\mathbf{\Phi}$ and $\mathbf{Y}$ are of varying dimensions, the matrix products $\mathbf{\Phi}^t\mathbf{Y}$ and $\mathbf{\Phi}^t\mathbf{\Phi}$ are of always of constant small (relative to the number of pixels in most regions) dimensions, independent of the dimensions of $\mathbf{\Phi}$ and $\mathbf{Y}$, which change with the number of pixels in a region. That is, the matrix product $\mathbf{\Phi}^t\mathbf{Y}$ is $m \times d$ and $\mathbf{\Phi}^t\mathbf{\Phi}$ is $m \times m$.

Incremental Computation of Covariances Matrices S_j: From Equation (17) we have:

$$nS = \mathbf{Y}^t\mathbf{Y} - \hat{\mathbf{\Theta}}^t[\mathbf{\Phi}^t\mathbf{Y}]. \quad (23)$$

In the incremental computation we utilize the fact that $\mathbf{Y}^t\mathbf{Y} = \mathbf{Y}_1^t\mathbf{Y}_1 + \mathbf{Y}_2^t\mathbf{Y}_2$, and $\mathbf{\Phi}^t\mathbf{Y} = [\mathbf{\Phi}_1^t\mathbf{Y}_1 + \mathbf{\Phi}_2^t\mathbf{Y}_2]$, which reduces the number of computations in the incremental computation of the covariance matrix. Furthermore, all the matrices involved in the computation of S ($\hat{\mathbf{\Sigma}}$) and $\mathbf{\Theta}$ have fixed dimensions and therefore the bookkeeping involved with dynamically changing region sizes is reduced.

4 Segmentation Algorithm

The problem our algorithm must solve is one of finding the minimum of Equation (10) over all $\mathbf{\Omega}$. Obtaining the absolute (global) minimum is infeasible because the search space is so large. For this reason we use a hierarchical algorithm similar to that used in [22, 4] to find a good, though perhaps local, minimum. It starts with an initial segmentation of the image. This may be just the image itself, with each pixel considered to be a separate region, or it may consist of larger regions produced by some heuristic device. Starting with this initial segmentation, the algorithm successively merges pairs of neighboring regions provided that the mergers decrease the total code-length. At each step the pair of regions producing the greatest codelength decrease are merged.

The MDL codelength decrease, δ_{tv}, due to a merger between two neighboring regions, ω_t and ω_v, can be deduced from Equation (10):

$$\begin{aligned}\delta_{tv} = {} & [l_{tv}\log 3 + \log^* l_{tv}] \quad (24)\\ & + \tfrac{1}{2}\left[n_t \log|S_t| + n_v \log|S_v| - (n_t + n_v)\log|S_{tv}|\right]\\ & + \tfrac{1}{2}\left[K_{\beta_t}\log n_t + K_{\beta_v}\log n_v - K_{\beta_{tv}}\log(n_t + n_v)\right]\end{aligned}$$

where S_{tv} denotes the sample covariance matrix of the combined region $\omega_t \cup \omega_v$. As mentioned above, at each step in the algorithm we search for the two regions ω_t,ω_v that yield the greatest code-length decrease δ_{tv} when merged. The first term expresses the savings due to the fact that a boundary branch drops when the merger occurs. The second term is the increase in the codelength of the actual data values themselves. This results from going to a single distribution from a separate distribution for each region. The third term is the savings associated with the fact that we have fewer model parameters to describe after the merger. K_{β_t}, K_{β_v} and $K_{\beta_{tv}}$ represent the number of parameters in the models representing the regions t, v, and tv, respectively.

The total number of merger steps needed to reach the final classification equals the number of initial regions r_0 minus the final number of regions R (usually $R \ll r_0$). The regions are ordered with a *heap-based priority queue* to select the best merger and at each step a time proportional to $\log r_0$ is required to maintain the queue, thus making the run time proportional to $r_0 \log r_0$. The memory size required by the algorithm equals the total number of regions (both the initial and the newly created, summing up, in the worst case, to $2r_0$) multiplied by the memory size required by the data set of a single region, which is roughly proportional to the average number of neighbors of a single region. This last number is usually much smaller than r_0, and therefore the memory requirements of this hierarchical algorithm are also proportional to r_0, being modest when compared to conventional hierarchical clustering procedures that try to merge every possible pair regardless of spatial location, and therefore requiring a memory size proportional to $\frac{1}{2}r_0(r_0 - 1)$. Moreover, some of the items of the data sets of inactive regions may be erased to save memory space.

The algorithm is run by first fixing the maximum degree of regression polynomials to 0. That is, region greyvalues are represented by piece-wise constant functions. After the algorithm converges to a segmentation (because it is more expensive to encode the image if any further merging is done), merging is attempted with first order polynomials representing the merged regions. This is continued until the merging converges. The process of incrementing the regression polynomial order and merging until convergence is continued until no merging is accomplished for a particular degree of the polynomial. Note that this process can be stopped at any degree of fit and will still result in closed region boundaries.

5 Experimental Results

To test the algorithm, we implemented it in C on an IBM RISC-System/6000 model 970. Here we show results from running it on two real images. The segmentation results for various maximum polynomial order fits are shown. The computation time for images of size 128×128 was on the order of 180 seconds.

Our first real image test case (Figure 1) was a real 128×128 two-band (red and blue) image of a small fragment of an electronic circuit. Figure 2 is the segmentation result when the maximum degree of polynomial allowed is two.

Our second real image test case was the real 128×128 pixel image of a house in Figure 3. This is a single-band grayscale image. The segmentation result allowing 2^{nd}-order polynomials is shown in Figure 4. Note that when maximum allowed degree of fit is two (quadratic surfaces), some of the regions still can have piece-wise constant and linear surfaces since the MDL criterion might find it cheaper to encode those regions

that way. This model selection is, of course, done automatically using the MDL criterion – there are no heuristic thresholds.

6 Discussion and Conclusions

We have developed an MDL-based objective function for multi-band image segmentation and an efficient segmentation algorithm that performs a suboptimal minimization of this criterion. The algorithm is incremental and makes use of computations performed in previous stages. The algorithm was tested on both synthetic and real images. The speed and performance of the algorithm on the test images were quite good and no manually adjusted thresholds were required. It should be mentioned that this algorithm can be used to treat texture-based segmentation by using the appropriate texture operators to compute the input bands for this algorithm. Natural extensions of this algorithm/work are discussed in [9].

Acknowledgement: The authors would like to thank Jacob Sheinvald, Nimrod Meggidio, Jorma Rissanen, Myron Flickner and Harpreet Sawhney for illuminating discussions.

References

[1] N. Abramson. *Information Theory and Coding.* McGraw-Hill, 1963.

[2] H. Akaike. A new look at statistical model identification. *IEEE Trans.*, AC-19:716–723, 1974.

[3] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis.* Wiley, New York, second edition, 1984.

[4] J. M. Beaulieu and M. Goldberg. Hierarchy in picture segmentation: A stepwise optimization approach. *IEEE PAMI*, 11(2):150–163, February 1989.

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

[6] Trevor Darrel, Stan Sclaroff, and Alex Pentland. Segmentation by minimal description. In *ICCV 90, Osaka, Japan*, pages 112–116, 1990.

[7] P. Fua and A. J. Hanson. Extracting generic shapes using model driven optimization. In *Proceedings of the Image Understanding Workshop*, pages 994–1004, Boston, 1988.

[8] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.

[9] T. Kanungo, B. Dom, W. Niblack, and D. Steele. A fast algorithm for MDL-based multi-band image segmentation. Technical Report 9754, IBM Research Division, March 1994.

[10] K. Keeler. Minimal length encoding of planar subdivision topologies with application to image segmentation. In *AAAI 1990 Spring Symposium of the Theory and Application of Minimal Length Encoding*, 1990.

[11] D. Keren, R. Marcus, M. Werman, and S. Peleg. Segmentation by minimum length encoding. In *ICPR 90*, pages 681–683, 1990.

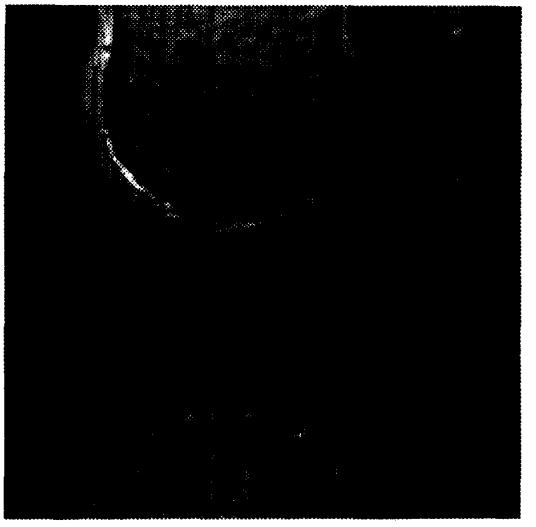

Figure 1: The grayscale image of the red band of a real, two-band (red and blue), image of a small fragment of an electronic circuit.

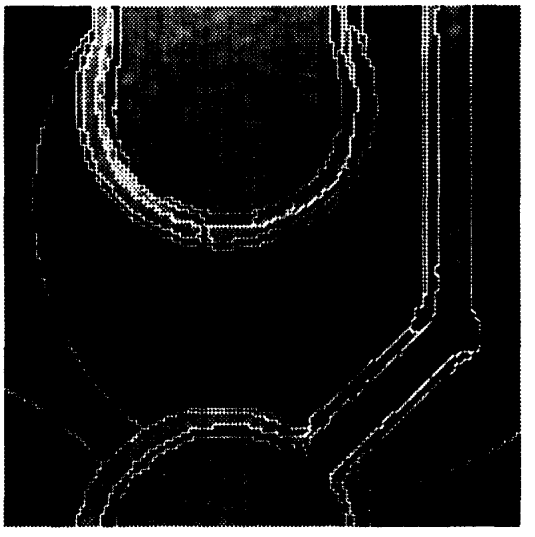

Figure 2: Segmentation result for the electronic circuit image. In this run, the maximum degree of polynomial was two. That is, a piece-wise quadratic model was used for each region. Some of the regions in the piecewise linear result have been merged in this result.

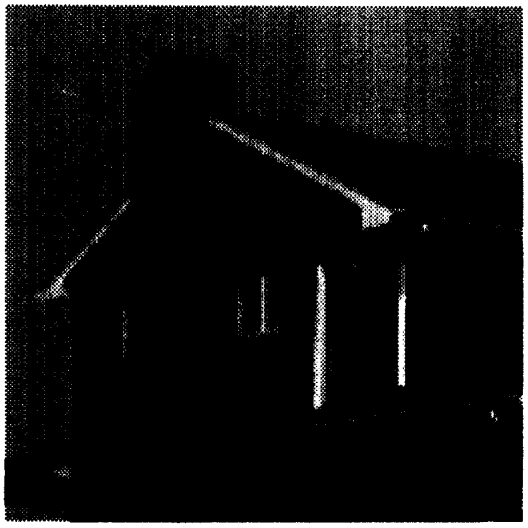

Figure 3: A real, 128×128, grayscale image of a house.

Figure 4: Segmentation result for the house image. In this run, the maximum degree of polynomial was one. That is, a piece-wise linear model was used for each region. Notice that some of the regions in piece-wise constant result been merged in this result.

[12] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[13] Y. G. Leclerc. Region grouping using the minimum-description-length principle. In *DARPA Image Understanding Workshop*, 1990.

[14] S. Liou, A. H. Chin, and R. Jain. A parallel technique for signal-level perceptual organization. *IEEE Trans PAMI*, 13(4):317–325, 1991.

[15] S. Liou and R. Jain. An approach to three-dimensional image segmentation. *CVGIP: Image Understanding*, 53(3):237–252, 1991.

[16] R. Nohre. *Topics in Descriptive Complexity*. PhD thesis, Technical University of Linkoping, 1993. See Chapt. 2 Coding Small Data Sets.

[17] A. Pentland. Part segmentation for objects recognition. *Neural Computation*, 1:82–91, 1989.

[18] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[19] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 2(11):211–222, 1983.

[20] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15. World Scientific Series in Computer Science, 1989.

[21] J. Rissanen. Fisher information and stochastic complexity. Research Report RJ 9547, IBM Research Division, 1993.

[22] J. Sheinvald, B. Dom, W. Niblack, and D. Steele. Unsupervised image segmentation using the minimum description length principle. In *Proceedings of ICPR 92*, August 1992. For an expanded version see: *IBM Research Report RJ 8474 (76695)*, (11/1/91).

[23] R. S. Wallace and T. Kanade. Finding natural clusters having minimum description length. In *AAAI 1990 Spring Symposium on the Theory and Application of Minimum Length Methods*, Stanford University, Stanford, CA, 1990.

[24] J. Zhang and J. W. Modestino. A model fitting approach to cluster validation with application to stochastic model-based image segmentation. *IEEE PAMI*, 12(10):1009–1017, October 1990.