Fast Piecewise Constant Approximation of Images

Hayder Radha* **
Martin Vetterli**
Riccardo Leonardi*

*** AT&T Bell Laboratories**
2L313 Crawfords Corner Rd, Holmdel, New Jersey 07733
(201) 949-2799

**** Center for Telecom. Research and Dept. of Elec. Eng.**
Columbia University, New York, New York 10027
(212) 854-3109

ABSTRACT

In this work, we present a Least-Square-Error (LSE), recursive method for generating piecewise-constant approximations of images. The method is developed using an optimization approach to minimize a cost function. The cost function, proposed here, is based on segmenting the image, recursively, using Binary Space Partitionings (BSPs) of the image domain. We derive a LSE necessary condition for the optimum piecewise-constant approximation, and use this condition to develop an algorithm for generating the LSE, BSP-based approximation. The proposed algorithm provides a significant reduction in the computational expense when compared with a brute force method. As shown in the paper, the LSE algorithm generates efficient segmentations of simple as well as complex images. This shows the potential of the LSE approximation approach for image coding applications. Moreover, the BSP-based segmentation provides a very simple (yet flexible) description of the regions resulting from the partitioning. This makes the proposed approximation method useful for performing image affine transformations (e.g., rotation and scaling) which are common in computer graphics applications.

1. INTRODUCTION

Many image processing and computer vision problems have been approached, in recent years, using segmentation-based image representation methods. For example, contour-texture segmentation techniques are used in [Kunt] and [Kocher] for high compression image coding applications. In [Leclerc] the author explains the importance of solving the *image partitioning problem* as a first step toward solving the *scene partitioning problem* which arises in image understanding applications. Leclerc solves the image partitioning problem in the framework of the minimum-description-length (MDL) principle [Rissanen] using a segmentation-based language (contours and regions). The work by Mumford and Shah [Mumford] approaches boundary detection as a segmentation problem, and provides an excellent framework for this problem using optimization of functionals. A similar functional method defined on a discrete domain was first introduced in [Geman]. The works presented in [Marroquin] and [Besl] are examples of approaching the image segmentation problem with a regularization solution.

These and many other segmentation-based image representation and description methods share two common aspects. First, the desired image is modeled as a *piecewise-smooth* function. This piecewise-smooth model is usually represented by (1) the geometry (e.g., shape or boundary) of the regions resulting from the segmentation, and (2) smooth and continuous functions (e.g., low order polynomials) representing the interiors of these regions. The second common aspect of segmentation-based approaches is the requirement to segment the image into a *minimum number* of regions (to achieve efficient representation) such that the piecewise-smooth approximation (or the reconstructed image from the segmentation-based description) is a *minimum distance* from the original image. These two desired features provide the right ingredients for any segmentation-based description method to be solved as an optimization problem.

Only few of the papers mentioned above approach the image segmentation problem as one of optimization. Good examples of optimization-based treatment to the problem are [Mumford] and [Leclerc]. Our objective in this paper is to develop, within the framework of optimization theory, a fast algorithm for constructing a *simple* segmentation-based description of an image using a piecewise-constant approximation model (which is a special case of the piecewise-smooth model).

The first step needed for an optimization-based treatment, is the formulation of a cost function (or a functional in variational analysis). The cost function derived for any segmentation-based image representation approach is dependent on the particular language selected for describing the geometry and the interiors of the piecewise-smooth approximation model. In our recent work [Radha 90b] [Radha 91b], we introduced a *Binary Space Partitioning* tree representation of images. We have shown that a BSP tree-based description of images provides efficient representation useful for image coding and manipulation applications. In the next section we explain, briefly, the BSP tree representation of images, and formulate the appropriate cost function, which is based on the BSP tree description language. In Section 3 we describe how our cost function fits within the framework of optimal approximation theory, and use this cost function to derive the main results of this work. The main results are *global* and *local* necessary conditions for the piecewise-constant approximation which minimizes our cost function. Based on these results, we outline in Section 4 an algorithm for constructing a piecewise-constant approximation of an arbitrary image using the BSP tree representation method. Simulation results are shown in Section 5.

2. LSE-BASED BSP TREE REPRESENTATION OF IMAGES

In [Radha 90b] and [Radha 91b], we described an image representation method based on Binary Space Partitioning (BSP) [Thibault]. This method provides an efficient representation (useful for image coding), a simple data structure (binary tree), and a very flexible description of the geometry of the regions resulting from the segmentation. Both the simple data structure and the flexible geometric description make the BSP approach useful for performing affine transformations (e.g. rotation and scaling) that are common in computer graphics applications.

The BSP approach partitions the desired image, recursively, by straight lines in a hierarchical manner. First, a line is selected (based on an appropriate criterion) to partition the whole image into two sub-images. Using the same criterion, two lines are selected to split the two sub-images resulting from the first partitioning. This procedure is repeated until a terminating criterion is reached. The outcome of this recursive partitioning is a set of (unpartitioned) convex regions which are referred to as the *cells* of the segmented image. A good segmentation is obtained when the pixel values within each cell are homogeneous. The recursive partitioning generates a binary tree representation of the image known as the *Binary Space Partitioning tree* (BSP tree). The non-leaf nodes of the BSP tree represent the partitioning lines, and the leaves represent the cells (unpartitioned regions) of the image.

The most critical aspect of the BSP representation approach is the criterion used for selecting the partitioning lines. In our previous work [Radha 91b], we based the partitioning on the image boundary information (edges). This criterion provides, in general, very good segmentation. However, the accuracy of this boundary-based BSP tree representation is constrained by the number of edge points one can detect from the original image. This can be a serious problem for images with week boundaries and low contrast [Radha 91a] [Radha 91b]. Another criterion for selecting the partitioning lines could be to minimize some error function.

In this work we develop a Least-Square-Error (LSE) method for generating a BSP-based, piecewise-constant approximation of images. When partitioning the image by a straight line into two sub-images, we approximate the pixels' intensities within these two sub-images by their respective mean values. We use the Square Error (SE) function $e(x, y)$, which is the square of the difference between the original image $I(x, y)$ and the piecewise-constant approximation $m(x, y)$. As shown in Figure 1, $m(x, y)$ consists of (1) two constant values corresponding to the two sub-images' means, and (2) a discontinuity (between the two constant values) along the partitioning line. For each line h passing through the image domain there is a SE function $e(x,y; h)$ and a mean function $m(x,y; h)$ (which is the piecewise-constant approximation) associated with that line. By integrating the SE function $e(x,y; h)$ over the whole image domain, one gets the total square error $E(h)$ resulting from approximating the two sub-images by their respective mean values.

Therefore, $E(h)$ is the cost function of the recursive, BSP tree representation method. At every step of the BSP recursion, our objective is to find the (optimum) line h_0 which minimizes our cost function $E(h)$. We refer to h_0 as the *LSE line*. For example, if H denotes the set of straight lines that pass through the domain of the desired image $I(x, y)$, then the (first) line h_0 selected to partition $I(x, y)$ has to meet the following condition:

$$h_0 = \min_{h \in H} E(h)$$

Using a brute-force method, h_0 can be determined by computing the total error $E(h)$ for *all* possible lines (i.e. for all $h \in H$) that pass through the image domain. However, this exhaustive search is very computationally intensive. As an example, for an image of size NxN pixels, the brute-force method requires on the order of N^4 operations to detect the optimum line.

Using the cost function $E(h)$, in this paper we derive a *necessary condition* for the optimum partitioning line h_0. We refer to this condition as the *LSE test*. Performing the LSE test on every line h that passes through the image domain produces a list of LSE *candidates* which meet the LSE necessary condition. Now h_0 can be determined by computing the total square error $E(h)$ only for the lines in the LSE candidate list, and selecting the candidate with the minimum square error. Using the LSE test when searching for h_0 in the set of all possible lines H can reduce the amount of computations significantly. For an NxN image, the amount of computations required to detect h_0 is on the order of N^3 which represents a reduction by a factor of N when compared with the brute-force method. Even for small images (e.g. N=128) this represents a saving of about two orders of magnitude. At the end of Section 4, we provide a rough estimate for the computational advantage of using our proposed LSE algorithm versus employing the brute force method.

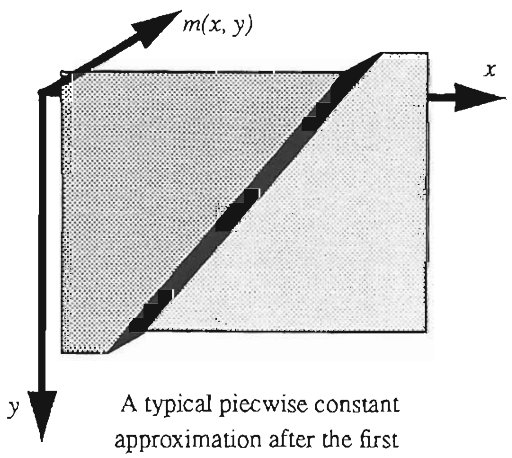
After detecting the first LSE partitioning line, the same process is repeated on the two resulting sub-images. In other words, for each sub-image, one has to detect the optimum partitioning line which minimizes the total square error of the piecewise-constant (mean) approximation of that sub-image. The LSE test is also used when searching for the two partitioning lines of the two sub-images. The process is repeated recursively until the total or average square error (for approximating a given region by its pixels' mean) is smaller than some threshold. When this happens, the region under consideration will not be partitioned, and it becomes a cell of the BSP-based representation of the original image.

As shown below, a given line $y=h(x)$ is a LSE candidate if the image function *along that line* (i.e. the function $I(x, y) = I(x, h)$) satisfies a simple condition (which is the LSE necessary condition mentioned above). This result is the main contribution of our work, and will be derived in the next section. Since the LSE test has to be performed on lines that pass through arbitrary convex regions of the image, it is shown that our result is valid for *any convex domain* $D \subset R^2$. We also show that the LSE test can be applied not only to straight lines but also to more general partitioning curves that satisfy certain conditions.

3. LSE BINARY PARTITIONING OF CONTINUOUS FUNCTIONS

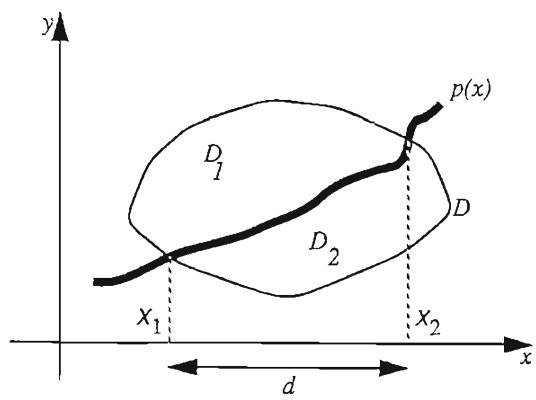
In this section, we derive a necessary condition for the optimum (in the least-square sense) piecewise-constant approximation $m(x,y)$ of a given function $I(x,y)$, when $m(x,y)$ consists of two constant values only. We assume that $I(x,y)$ is defined over a *bounded convex* region D ($\subset R^2$). In other words, D is enclosed within a convex curve c(x,y) = 0 as shown in Figure 2.

We denote D_1 and D_2 to the two subdomains resulting from partitioning D by a continuous curve $y = p(x;\alpha)$, where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in R^n$ is a vector in n-dimensional space.



A typical piecewise constant approximation after the first partitioning of the whole image.

Figure 1



Binary Partitioning of the domain D by the curve $p(x)$.

Figure 2

The elements of α represent n independent variables that parameterize the partitioning curve p *uniquely*. In other words, a given α defines a unique curve $p(x;\alpha)$. For example, the straight line $y = \alpha_1 + \alpha_2 x$ is uniquely defined by the two parameters α_1, which represents the y-intercept, and α_2, which represents the slope of the line.

Let m_1 and m_2 be the constant (with respect to x and y), approximations of $I(x,y)$ over D_1 and D_2, respectively. Although m_1 and m_2 are constants with respect to the space variables (x,y), they (m_1 and m_2) are functions of the parameter variables $(\alpha_1, \alpha_2, \ldots, \alpha_n)$. For a given convex contour $c(x,y)$ and a class of partitioning curves P satisfying $y = p(x)$ such that p and c intersect in two points only, our objective is to find $m_1(p)$, $m_2(p)$, and p which minimize the following error (or cost) function:

$$E(m_1, m_2;\ p) = \int\int_{D_1(p)} \left[I(x,y) - m_1(p)\right]^2 dxdy + \int\int_{D_2(p)} \left[I(x,y) - m_2(p)\right]^2 dxdy \qquad (1)$$

It can be easily shown that, for a given p, the m_1 and m_2 which minimize $E(m_1, m_2;\ p)$ can be expressed as follows:

$$m_i(p) = \frac{\int\int_{D_i(p)} I(x,y)\,dxdy}{\int\int_{D_i(p)} dxdy} = \frac{I_i(p)}{A_i(p)} \qquad (2)$$

for $i = 1,2$. Therefore, m_1 and m_2 that minimize E are the mean values of $I(x,y)$ over the subdomains D_1 and D_2.

It is important to note that the optimum constant values (i.e., m_1 and m_2) and the subdomains D_1 and D_2 are functions of the partitioning curve p. Therefore, one can express the error function E as a function of p only (i.e., $E(p)$ instead of $E(m_1, m_2;\ p)$). Moreover, since p is uniquely defined by the parameter vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, the above optimization problem is equivalent to finding $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ that minimizes the following square error (cost) function:

$$E(\alpha) = \int\int_{D_1(p(x;\alpha))} \left[I(x,y) - m_1(\alpha)\right]^2 dxdy + \int\int_{D_2(p(x;\alpha))} \left[I(x,y) - m_2(\alpha)\right]^2 dxdy \qquad (3)$$

Before proceeding, it is important to take a closer look at the error functions $e_1 = [I(x,y) - m_1]^2$ and $e_2 = [I(x,y) - m_2]^2$. Throughout the rest of the paper, we refer to these functions as the error *density* functions because they measure the amount of error per unit area over D_1 and D_2, respectively. Since each of e_1 and e_2 is a functional of $p(x;\alpha)$, any change in one or more of the parameter variables $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ will cause both e_1 and e_2 to change. This change is quantified by the partial derivatives $\partial e_1/\partial\alpha_i$ and $\partial e_2/\partial\alpha_i$, where $\alpha_i \in \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. It can be shown that the integral of $\partial e_1/\partial\alpha_i$ and $\partial e_2/\partial\alpha_i$ over their respective domains D_1 and D_2 are *always* zero. This is an important result and we state it as a lemma:

- **Lemma 1**

  Let $I(x,y)$ be a square integrable function over a domain $D \subset R^2$. Let $p(x;\alpha)$ partition D into the two subdomains D_1 and D_2. If e_1 and e_2 are the error density functions resulting from approximating $I(x,y)$ by its mean values m_1 and m_2 over D_1 and D_2, respectively, then e_1 and e_2 satisfy the following, for all $\alpha_i \in \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$:

$$\int\int_{D_1} \frac{\partial e_1}{\partial\alpha_i} dydx = 0 \quad \text{and} \quad \int\int_{D_2} \frac{\partial e_2}{\partial\alpha_i} dydx = 0 \qquad (4)$$

The proof to this lemma is given in [Radha 90a]. Although we will use Lemma 1 to derive the necessary condition of the LSE partitioning curve p which divides a *convex* domain D, the result of this lemma is valid for *concave* domains also.

Moreover, this result is valid when D is partitioned into more than two subdomains. However, the more general case is beyond the scope of this paper.

In the following sub-section we use Lemma 1 to derive a necessary condition for the parameter vector α that minimizes $E(\alpha)$. As will be shown, one form of this necessary condition is a relationship between the two error density functions $e_1(x,y;\ \alpha)$ and $e_2(x,y;\ \alpha)$ evaluated along the partitioning curve $y = p(x;\alpha)$.

3.1 The LSE Necessary Condition

To simplify the minimization of the error function E as expressed in equation (3), we restrict D to be a convex domain.

- **Theorem 1**

Let $I(x,y)$ be a square-integrable, continuous function over a bounded convex domain $D \subset R^2$. Let P_α be a class of continuous differentiable curves that satisfy $y = p(x;\alpha)$. A given p, which is uniquely defined by the parameter vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, partitions D into the two subdomains D_1 and D_2. If $\alpha_o = (\alpha_{01}, \alpha_{02}, \ldots, \alpha_{0n})$ is a point of relative (local) minimum for the error function

$$E(\alpha) = \iint_{D_1(p(x;\alpha))} \left[I(x,y) - m_1(\alpha)\right]^2 dxdy + \iint_{D_2(p(x;\alpha))} \left[I(x,y) - m_2(\alpha)\right]^2 dxdy \qquad (5a)$$

then α_o must satisfy the following:

$$\int_d p'_{\alpha_i}(\alpha_o)\ e_1(x,p;\ \alpha_0)\ dx = \int_d p'_{\alpha_i}(\alpha_o)\ e_2(x,p;\ \alpha_0)\ dx \qquad (5b)$$

for $i = 1,2,\ldots,n$, where d is the domain of the partitioning curve $p(x;\alpha_o)$ over the x-axis and within the convex region D, and $p'_{\alpha_i} = \partial p / \partial \alpha_i$.

Theorem 1 shows that the error density functions e_1 and e_2 have to meet the above condition *along the LSE partitioning curve*. In other words, a knowledge of the error function along a given p can determine if $m(x,y;\ p)$ is a potential LSE approximation of $I(x,y)$ or not. Therefore, if $e_1(x,y;\ \alpha)$ and $e_2(x,y;\ \alpha)$ do not satisfy equation (5b) for a given curve $p(x;\alpha)$, then we know that this p does not minimize the error function $E(\alpha)$. (Due to the lack of space, the proof is omitted here. A complete proof for the theorem is given in [Radha 90a].)

It should be clear that a knowledge of the error density functions e_1 and e_2 along the partitioning curve p requires a knowledge of (i) the constant approximations m_1 and m_2, and (ii) the original function $I(x,y)$ along p (i.e. $I(x,p)$). This leads to a more intersting form of the condition in equation 5b.

- **Corollary 1.1**

The necessary condition of Theorem 1 is equivalent to the following necessary condition for the partitioning curve $p(x;\alpha_0)$ which minimizes the error function $E(\alpha)$:

$$\int_d p'_{\alpha_i}(\alpha_o)\ \tfrac{1}{2}[m_1(\alpha_o) + m_2(\alpha_o)]\ dx = \int_d p'_{\alpha_i}(\alpha_o)\ I(x,p(x;\alpha_o))\ dx \qquad (5c)$$

If one thinks of p'_{α_i} as a *weighting* function (or a probability density in x over the domain d) for the original signal $I(x,y)$ along the partitioning curve p, then this corollary shows that the average value of $I(x,p)$ has to be equal to the average value of the two constant approximations m_1 and m_2 when p minimizes the error functional $E(p)$. It is important to note that this condition (of equal averages) has to be met for *all n* weighting functions p'_{α_i}, $\alpha_i \in \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

Since $[m_1(\alpha_o) + m_2(\alpha_o)]$ is constant with respect to the space variable x, one can write Eq. 5c as follows:

$$½[m_1(\alpha_o) + m_2(\alpha_o)] \int_d p'_{\alpha_i}(\alpha_o)\, dx = \int_d p'_{\alpha_i}(\alpha_o)\, I(x,p(x;\alpha_o))\, dx \quad (5d)$$

(In Eq. 5c we have selected to insert this constant expression inside the integral to show the analogy between $I(x,p)$ and $½[m_1(\alpha_0) + m_2(\alpha_0)]$.)

If $I(x,y)$ represents the z value in R^3 (i.e., $z = I(x,y)$), then the function $I(x,p)$ is the 3-D curve resulting from the intersection of the surface $z = I(x,y)$ and the infinite sheet $y - p(x;\alpha) = 0$. By projecting this 3-D curve into the (x,z) plane, one gets a function that (for a given p and $I(x,y)$) depends on x only. Throughout the rest of the paper we will denote this function by $I_p(x)$ (which is the same as $I(x,p)$).

Although Eq. 5d expresses the LSE necessary condition in terms of the function $I_p(x)$, one needs to perform some computation on $I(x,y)$ over the *whole* domain D in order to test if the condition of Corollary 1.1 is met or not. This computation is needed to evaluate the two values m_1 and m_2. In other words, a knowledge of $I_p(x)$ is not sufficient to help us determine if Eq. 5d is satisfied or not. Therefore, we refer to Eq. 5d as the *global* LSE necessary condition, since a knowledge (and processing) of the original function $I(x,y)$ over the whole region D under consideration is required to determine if the LSE condition holds.

In the next subsection we derive, from the global LSE condition, a *local* necessary condition for the optimum partitioning curve. As will be shown, one can determine if a given curve p is a LSE candidate by testing the behavior of $I(x,y)$ along that partitioning curve only, i.e. along $I(x,p)$. We derive the local condition for the special case when the partitioning curve p is a straight line h.

3.2 A Local Condition for the LSE Partitioning Lines

One way to uniquely define a straight line in the (x,y) plane is by the slope and y-intercept parameterization: $y = h(x) = \alpha_1 + \alpha_2 x$. From this line representation, one can derive the necessary conditions for the parameters α_1 and α_2 which minimize the error function $E(\alpha)$, where α in this case consists of α_1 and α_2 only. Using Eq. 5d, these two necessary conditions can be expressed as follows:

$$½[m_1(\alpha) + m_2(\alpha)] = \frac{\int_d I(x,h(x))\, dx}{(x_2 - x_1)} \quad (6a)$$

$$½[m_1(\alpha) + m_2(\alpha)] = \frac{\int_d x\, I(x,h(x))\, dx}{½(x_2^2 - x_1^2)} \quad (6b)$$

where x_1 and x_2 are the two end points of the domain d over the x-axis (see Figure 2).

Since the left hand sides of these two equations are the same, the right hand sides have to be equal. This leads to the following form of the LSE necessary condition for the optimum partitioning line h:

$$½(x_2 + x_1) \int_d I(x,h(x))\, dx = \int_d x\, I(x,h(x))\, dx \quad (7)$$

We refer to Eq. 7 as the *local* LSE condition for the optimum partitioning line h, since a knowledge of the function $I(x,y)$ along the line h (i.e., $I(x,h)$) is sufficient to determine if h is a potential LSE line. The local necessary condition of Eq. 7 is the key element of our proposed algorithm for generating a piecewise constant approximation of an image using the recursive BSP approach (described in Section 2). This algorithm is explained in detail in Section 4.

Although we have derived the above local condition for the slope and y-intercept (i.e., $h(x) = \alpha_1 + \alpha_2 x$) representation of straight lines, the necessary condition of Eq. 7 is valid for *any* desired line parameterization. In addition, it is important to note that there several ways to test and interpret the condition of Eq. 7. In the following subsection, we take a closer look at this necessary condition, and define a normalized transform useful for computing Eq. 7.

3.3 The LSE Partitioning Line (LPL) Transform

As explained in the previous subsection, a line h which minimizes the error function E (as expressed in Eq. (3)), has to satisfy the condition of Eq. 7. This necessary condition can be expressed as follows:

$$\int_{x_1}^{x_2} [\ x - \tfrac{1}{2}(x_2+x_1)\ ]\ I(x,h(x))\ dx = 0 \tag{8}$$

Using a change of variables: $t = x - \tfrac{1}{2}(x_2+x_1)$, and $x_c = \tfrac{1}{2}(x_2+x_1)$ it can be shown that Eq. (8) is equivalent to the following expression:

$$\int_{-t_0}^{t_0} t\ f(t)\ dt = 0 \tag{9}$$

where $f(t) = I(t+x_c, h(t+x_c))$, and $t_0 = \tfrac{1}{2}[x_2-x_1]$.

If one thinks of $f(t)$ as a weighting function for t, then Eq. (9) states that the weighted average of t has to be zero. Since $t = x-\tfrac{1}{2}[x_2+x_1]$, this is equivalent to having the average value of x (weighted by the image function $I(x,h)$ along the line h) equals the midpoint $x_c = \tfrac{1}{2}[x_2+x_1]$ between x_1 and x_2.

It should be clear from equations (8) and (9) that if the image function $I_h(x) = I(x,h)$ is even with respect to $\tfrac{1}{2}[x_2+x_1]$ (i.e., symmetric around x_c), then h is a potential LSE line. This class of even $I_h(x)$'s (which is one instance of the classes of functions that satisfy the above conditions) includes the case when the image function is constant along the line h. Moreover, if $I_h(x)$ has a nonnegative, constant value along one half of the interval $[x_1, x_2]$ (e.g., $[x_1, x_c]$), and has *another* nonnegative, constant value (or even several constant values) along the other half, then the above LSE condition is not satisfied. These observations lead to the following important (yet intuitive) result.

If the image function $I(x,y)$ over a given domain D (under consideration) has a simple step (or a ramp) edge, then the local LSE condition stated above eliminates all lines crossing the edge from being LSE candidates, and admits all lines parallel to the edge as being potential candidates for minimizing the error function E.

Based on the LSE local condition, we define the following image transform:

- **Definition**

  The LSE Partitioning Line (LPL) transform, $L(h)$, of a 2-D continuous function $I(x,y)$ over a domain D is defined as follows:

$$L(h) = \frac{\int_{x_1}^{x_2} x\ I(x,h)\ dx}{\int_{x_1}^{x_2} I(x,h)\ dx} \tag{10}$$

  where h is a straight line intersecting the boundary of the domain D in two points p_1 and p_2, and x_1 and x_2 are the x-coordinates of p_1 and p_2, respectively.

From this definition, the LPL Transform (LPLT) measures the average value of x within the domain $[x_1, x_2]$ weighted by the function $I(x,h)$. The denominator of the LPLT provides the desired normalization of the weighting function $I(x,h)$. Moreover, the LPLT transforms the image I over D from a function of the spatial variables (x,y) to a function $(L(h))$ of the straight lines passing through D. It is important to note that the above definition for $L(h)$ is independent of the specific line parameterization selected for representing h.

4. A RECURSIVE ALGORITHM FOR A LSE PIECEWISE-CONSTANT APPROXIMATION

Our proposed LSE-based algorithm requires two *strictly positive* thresholds: the average square error threshold T_e (which is needed to terminate the recursive Binary Space Partitioning of the image), and the line criterion threshold T_c (which is needed to select the LSE candidates as explained below).

To detect the LSE line, ideally one should look for all lines satisfying $L(h) = x_c$. In practice, however, if h is a LSE line, $L(h)$ could be very close to x_c (but not exactly equals to x_c). This explains the need for the line criterion threshold T_c which provides an upper bound for the distance between $L(h)$ and x_c as shown in the LSE algorithm outlined below.

Moreover, one needs to quantize the parameter space of h to consider a finite number of straight lines passing through the domain D. We denote H_D to the set of quantized straight lines passing through D, where the number of elements in H_D is proportional to the area of D. For example, if the image size is NxN pixels then the number of lines needed to be considered is on the order of N^2.

In addition to the two thresholds (i.e., T_e and T_c), the LPL transform $L(h)$, and the set of partitioning lines passing through D (i.e., H_D), we use the following variables for our proposed LSE algorithm:

1. D_1 and D_2 are the two subdomains resulting from partitioning D by h.
2. A_1 and A_2 are the areas of D_1 and D_2.
3. m_1 and m_2 are the mean values of the image function $I(x,y)$ over D_1 and D_2.
4. E_1 and E_2 are the total square errors resulting from approximating $I(x,y)$ by m_1 and m_2 over D_1 and D_2, i.e. $E = E_1 + E_2$.
5. S_1 and S_2 are the average square errors over D_1 and D_2, i.e. $S_1 = E_1/A_1$ and $S_2 = E_2/A_2$.

The LSE algorithm for generating a piecewise-constant approximation of an arbitrary image $I(x,y)$ consists of the following steps:

I. Set the domain D to the whole image domain, and begin the recursive partitioning.

II. Compute $L(h)$ and $x_c(h)$ for all $h \in H_D$.

III. Generate a set C_D (or a list) of the LSE candidate lines that satisfy the following:

$$|L(h) - x_c(h)| < T_c, \tag{11}$$

where T_c is a small positive number.

IV. Compute the total square error $E(h)$ for all lines $h \in C_D$. Select the line h_0 with the minimum value of $E(h)$, and use h_0 to partition D into D_1 and D_2.

V. If the average error $S_1 > T_e$, set $D = D_1$ and go to step (II). Otherwise, consider D_1 as a BSP cell (unpartitioned region) and use m_1 for approximating $I(x,y)$ over D_1.

VI. If the average error $S_2 > T_e$, set $D = D_2$ and go to step (II). Otherwise, consider D_2 as a BSP cell (unpartitioned region) and use m_2 for approximating $I(x,y)$ over D_2.

VII. End the recursion.

It is important to note that, in practice and due to computational errors when measuring $L(h)$, the above algorithm does *not always* guarantee that the selected lines are the LSE partitionings of their respective domains. However, as shown in the next section, this LSE-based algorithm provides very satisfactory results when tested on simple as well as complicated images. (From our experience, a selected line is either exactly or very close to the actual LSE partitioning line.)

It should be clear that the thresholds T_c and T_e have a significant impact on the efficiency (measured by the total number of unpartitioned regions) and the accuracy of the resulting piecewise-constant approximation. The higher T_c, the more partitioning lines are admitted to the set C_D. This increases the probability of detecting the actual LSE partitioning lines. On the other hand, the lower T_c, the less lines are admitted into C_D, and consequently, the faster one can detect the partitioning lines. Moreover, the lower T_e, the more accurate the approximation. However, the higher T_e, the less regions result from the partitioning, and therefore the more efficient the representation.

Before leaving this section, let us provide a rough estimate for the computational advantage of using the proposed LSE algorithm versus the brute force method for detecting the optimum partitioning line. With the brute force scenario one has to compute the error function E for all lines in the set H_D. For now let assume that H_D contains N_D discrete lines. Since the amount of computation required for evaluating E is proportional to the area A_D of the domain (under consideration) D, the computational expense of detecting the optimum partitioning line is on the order of (N_D A_D).

Using the LSE algorithm, however, one needs to compute the LPL transform $L(h)$ for all N_D lines. Since computing $L(h)$ is proportional to the length b of the domain D boundary, the expense of computing $L(h)$ for all N_D lines is on the order of N_D b. In addition, one needs to compute the square error E for all N_C lines which satisfy the LSE line criterion (i.e., for all $h \in C_D$). Therefore, the computational expense of using the LSE algorithm is on the order of (N_D b + N_C A_D). Based on these numbers, the computational advantage ratio R_c can be expressed as follows:

$$R_c = O\left[\frac{N_D\ A_D}{N_D\ b + N_C\ A_D}\right]. \tag{12}$$

If the number of LSE candidates (i.e., N_C) that satisfy the line criterion of equation (11) is very small (which is the case for most images) such that $N_D b >> N_C A_D$, the computational advantage R_c will be on the order of (A_D / b). For an image with NxN pixels this means that $R_c = O(N)$. If $N = 256$, which is the case for the images shown in the next section, this represents a computational saving of two orders of magnitudes. For example, using the brute force method, it takes about six hours (on a Sun-4 machine) to detect *only one* (the first) LSE partitioning line for the image shown in Figure 4, whereas using our LSE-based algorithm, it takes about 45 minutes (on the same machine) to generate the *complete* piecewise-constant approximation shown in Figure 7a. (See the next section for more details regarding our simulation results.)

5. SIMULATION RESULTS

The algorithm described is Section 4 was simulated in the *C* language, and tested on the original images shown in Figures 3, 4, and 5. These images represent simple, low-contrast, and highly-textured types of scenes, respectively. Throughout this section, we refer to Figures 3, 4, and 5 by the *cross, Mona Lisa*, and *girl* images, each with a size of 256x256 pixels. The main objective of presenting the results of the *cross* image is to demonstrate how the LSE algorithm works. Due to its simplicity, one would expect that a good segmentation-based representation of the *cross* image should generate a *perfect* approximation (i.e., zero error) with a minimum number of regions.

For all three images, we use a hierarchical method for sampling the parameter space of the straight lines passing through a given region D. In other words, for any region D (under consideration at a given step of the recursive partitioning) the number of lines tested against the LSE criterion of Eq. 11 is proportional to the area of D. For example, the number of lines considered for partitioning the whole image (i.e., at the first step of the recursion) is 2x256x256. As the partitioning progresses, this number is decreased to as little as four lines only. In addition to its computational advantage, this hierarchical approach for sampling the parameter space provides an efficient representation of the partitioning lines. This efficient representation is very important for image coding applications as explained in [Radha 90b] [Radha 91b]. (The detailed description on how to implement the hierarchical sampling of the

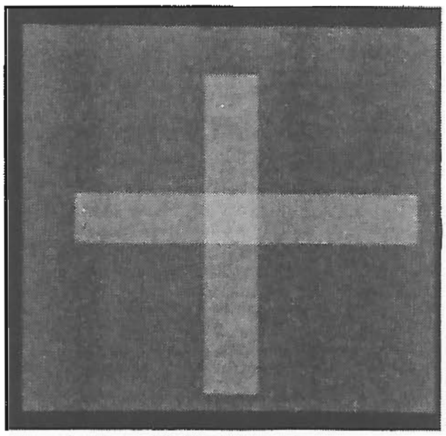


Figure 3: The *cross* image.

straight lines passing through a polygon with an arbitrary shape is beyond the scope of this paper, and is the subject of a future work regarding BSP-based image coding.)

Figure 6 shows the piecewise-constant approximation of the *cross* at different stages of the LSE recursive algorithm. As seen in the figure, at every step, the selected LSE line coincides exactly with one of the straight lines at the boundary of the *cross*. The final result (shown in Figure 6) is a perfect approximation of the *cross* with only 13 regions (i.e., 13 BSP tree cells). This demonstrates the efficiency of the LSE-based partitioning when applied on simple images containing strong edges.

For the *Mona Lisa* image, we set the line criterion threshold $T_c = 0.05$, and the average square error threshold $T_e = 100$. Figures 7a and 7c show an intermediate and the final approximations with the selected partitioning lines, respectively. The intermediate image (Figure 7a) represents the piecewise-constant approximation when $T_e = 200$. Figures 7b and 7d show the corresponding piecewise-constant approximations without the partitioning lines. As seen from the figures, the LSE algorithm succeeded in selecting straight lines that pass through the boundaries of the objects in the image despite the fact that this image has very low contrasts among its different regions. This results in an efficient partitioning of the image, where the total number of polygons (cells) are about 450 and 1000, for the intermediate and final images, respectively. It is clear also that by lowering the error threshold, one can obtain more accurate representation of the original image on the expense of more cells.

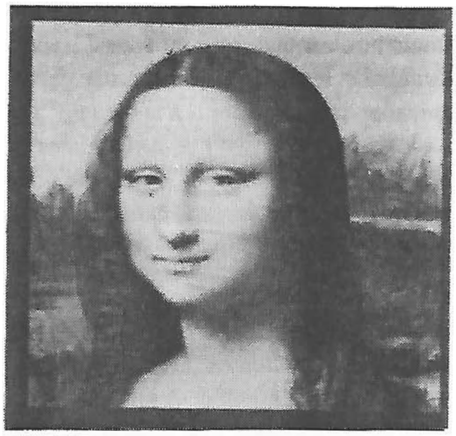


Figure 4: The *Mona Lisa* image.

Figure 5: The *girl* image.

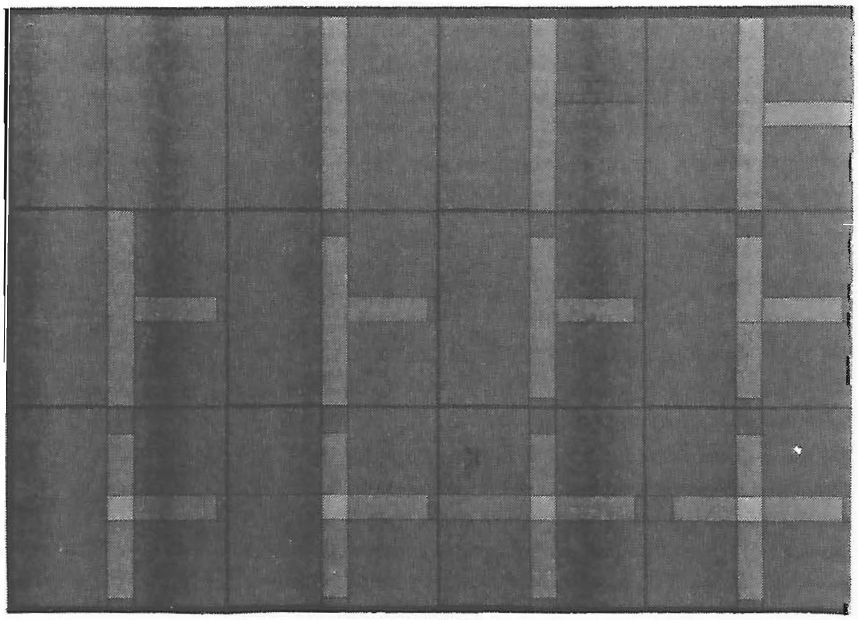


Figure 6: Intermediate and final approximations of the *cross* image.

Figure 7: Intermediate and final approximations of the *Mona Lisa* image.

a	b
c	d

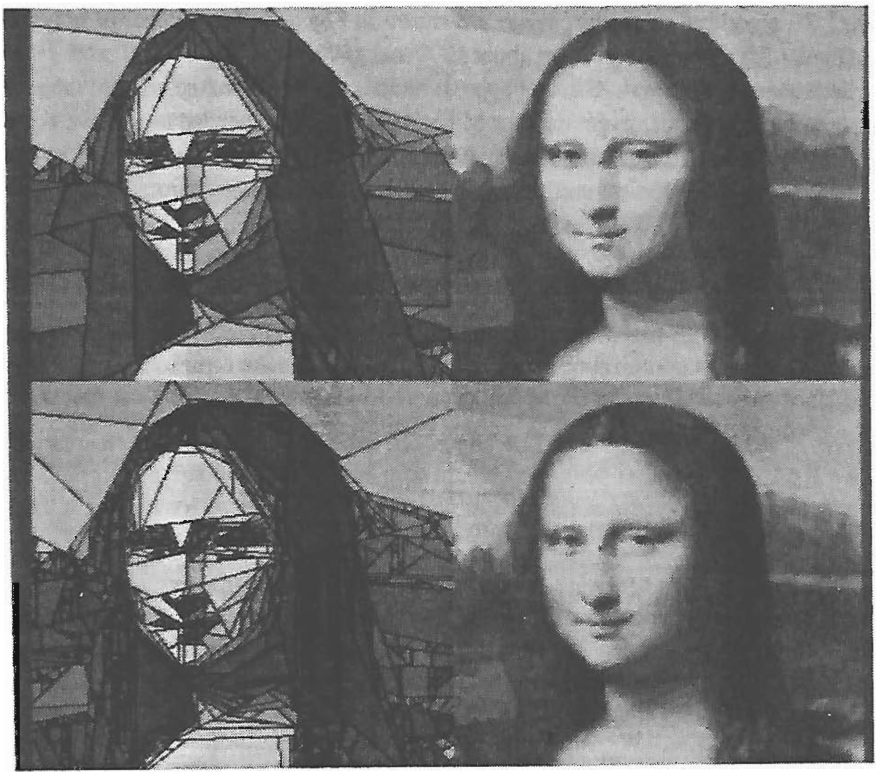
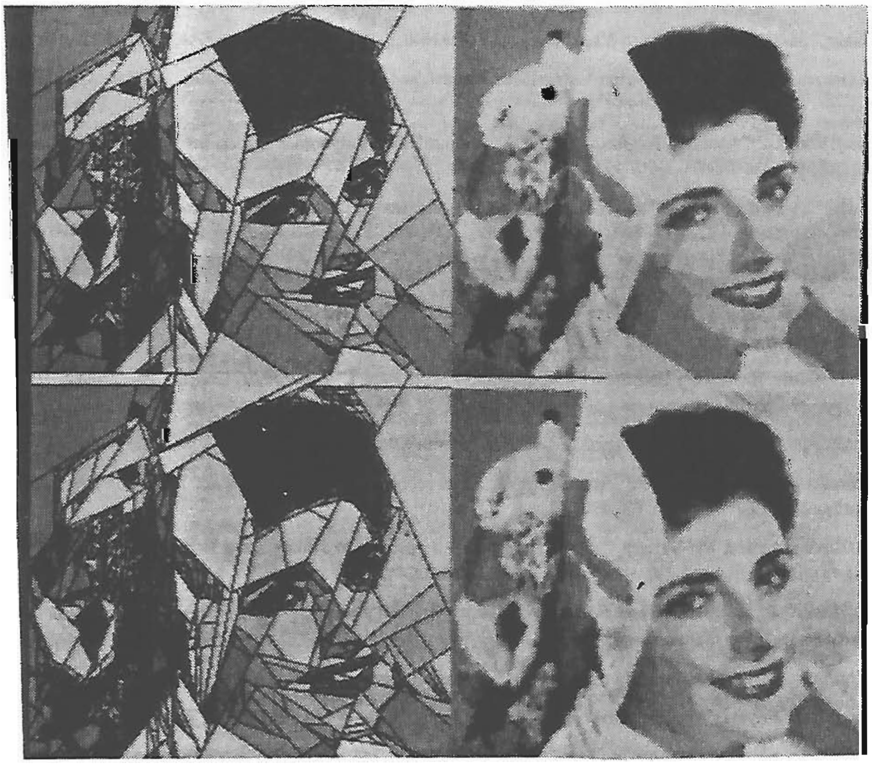


Figure 8: Intermediate and final approximations of the *girl* image.

a	b
c	d



Similar to the *Mona Lisa* image, we set the line criterion threshold T_c to 0.05 for the *girl* image, and used $T_e = 100$. The results of applying the LSE algorithm on the *girl* image are shown in Figure 8. The number of polygons for the intermediate (Figure 8a) and final (Figure 8c) approximations are about 1000 and 1500, when $T_e = 200$ and 100, respectively. Figures 8b and 8d show the corresponding piecewise-constant approximations of the *girl* image without the partitioning lines. Due to the large textured areas in the *girl* image, larger number of partitionings where needed to achieve the same level of accuracy as in the *Mona Lisa* piecewise-constant approximations (shown in Figure 7). Again, in this case, one can notice that the LSE algorithm succeeded in producing an efficient partitioning by selecting lines passing through the boundaries (or edges) of the objects in the image.

6. CONCLUSION AND FUTURE WORK

In this work, we have presented a LSE-based, recursive method for generating piecewise-constant approximations of images. The method was developed using an optimization approach to minimize a square error (cost) function. We have derived both a global and local LSE necessary condition for the optimum curve which minimizes this error function.

As shown in the previous section, the proposed LSE algorithm provides an efficient segmentation of simple as well as complex images. This shows the potential of this approximation approach for image coding applications. In addition, the proposed algorithm provides a significant reduction in the computational expense for detecting the LSE partitioning lines when compared with a brute force method.

Currently, we are working on using this LSE-based and a boundary-based (proposed in our previous work [Radha 91b]) BSP tree representation method to develop an efficient image coding algorithm. Moreover, further improvements can be made to the LSE algorithm. For example, a better way to detect the partitioning lines that satisfy the LSE condition (see Eq. 11), is to look for the zero-crossings of $|L(h) - x_c(h)|$ in the parameter space rather than using a threshold value (i.e., the line criterion threshold T_c). This new searching method can eliminate a large number of LSE candidate lines, and therefore, can improve the speed of the algorithm. We also considering a hybrid approach where both boundary and LSE-based partitionings are used to generate the piecewise-constant approximation of images. It is our belief that this hybrid approach will provide a better segmentation than either method applied alone.

References

[Besl] P. J. Besl and R. C. Jain, "Segmentation Through Variable-Order Surface Fitting," *IEEE Trans. on PAMI*, Vol. 10 No. 2, 1988.

[Geman] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and The Bayesian Restoration of images," *IEEE Trans. on PAMI*, Vol. 6 No. 6, pp. 721-741, 1984.

[Kocher] M. Kocher and R. Leonardi, "Adaptive Region Growing Technique Using Polynomial Functions for Image Approximation," *Signal Processing*, Vol. 11 No. 1, pp. 47-60, July 1986.

[Kunt] M. Kunt, M. Benard, R. Leonardi, "Recent Results in High-Compression Image Coding," IEEE Trans. on Circ. and Syst., Vol. CAS-34, No. 11, pp. 1306-1336 (Nov. 1987).

[Leclerc] Y. G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning," *International Journal of Computer Vision*, 3, 73-102, 1989.

[Marroquin] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic Solution of Ill-Posed Problems in Computational Vision," *Journal of the American Statistical Association*, pp.76-89, Vol. 82, March 1987.

[Mumford] D. Mumford and J. Shah, "Boundary Detection by Minimizing Functionals, I," *Proc. IEEE Comp. Society Conf. on Computer Vision and Pattern Recognition*, San Francisco, pp. 22-26, 1985.

[Radha 90a] H. Radha, "Least-Square Binary Partitioning of L_2 Functions with Convex Domains," *AT&T Bell Labs. Tech. Memo, Draft*,, Sept. 1990.

[Radha 90b] H. Radha, R. Leonardi, B. Naylor, and M. Vetterli, "Image Representation Using Binary Space Partitioning Trees," *Visual Communications and Image Processing V Proceedings*, October 1990, pp. 639-650.

[Radha 91a] H. Radha, R. Leonardi, and M. Vetterli, "A Multiresolution Approach to Binary Tree Representations of Images," *ICASSP Proceedings*, pp.2653-2656, May 1991.

[Radha 91b] H. Radha, R. Leonardi, M. Vetterli, and B. Naylor, "Binary Space Partitioning (BSP) Tree Representation of Images," *Journal of Visual Communication and Image Representation*, September 1991.

[Rissanen] J. Rissanen, "Minimum-Description-Length Principle," In *Encyclopedia of Statistical Sciences*, Wiley, New York, pp. 523-527, 1987.

[Thibault] W. C. Thibault and B. F. Naylor, "Set Operations on Polyhedra Using Binary Space Partitioning Trees," *SIGGRAPH*, July 1987, pp. 153-162.