

# Symmetrical Segmentation-Based Image Coding

C. Saraceno and R. Leonardi\*  
Signals & Communications Lab.  
Department of Electronics for Automation,  
University of Brescia,  
Brescia, I-25123, Italy.

## Abstract

An image coding technique based on symmetry extraction and Binary Space Partitioning (BSP) Tree representation for still pictures is presented. Axes of symmetry, detected through a principal axis of inertia approach and a coefficient of symmetry measure, are used to divide recursively an input image into a finite number of convex regions. This recursive partitioning results in the BSP tree representation of the image data. The iterative partition occurs whenever the current left/right node of the tree cannot be represented "symmetrically" by its counterpart, i.e. the right/left node. This splitting process may also end whenever the region associated with a given node has homogeneous characteristics or its size falls below a certain threshold. Given a BSP tree partition for an image, and the "seed" leaf nodes (i.e. those that cannot be generated by mirroring their counterparts), the remaining leaf nodes of the tree are reconstructed using an iterative predictive scheme.

## 1 Introduction

Pixel-based representation of images have shown their limitation for most image processing applications ranging from image manipulation to image compression. Several image processing, computer vision and computer graphics applications have resorted to employ more efficient representations of the pictorial information. In particular, waveform representation which is regularly considered for image compression, lacks of flexibility, as it cannot handle the non-statistical or at least non stationary characteristics of visual information. This limits not only the ease of processing pictorial data, but also the performance to achieve a desired goal (for example,  $R(D)$  bound). Since a decade, segmentation-based representation have been suggested as they are more able to adapt to the physical nature of the image data (piecewise uniform characteristics, within the objects of a natural scene). It is essential to widen the horizon of this kind of representation, with the hope of breaking the asymptotic behavior reached by waveform coding representation.

As an example, tree-based representations have shown promising potential due to the simplicity of the data structure and navigation through such structure. Originally  $2^n$  trees have been considered to represent data in an  $n$ -dimensional space [1]. For simplicity, the partitioning of the  $n$ -dimensional signal has been performed with hyperplane boundaries whose direction remains parallel to the basis of the  $n$ -dimensional space, throughout the partitioning. As an example, images are represented using quadrees by recursively partitioning the image plane into squares whose side are parallel to line and column directions (see Fig. 1).

\*Research partly supported by a grant from Società Italiana per le Telecomunicazioni Italtel S.p.A.

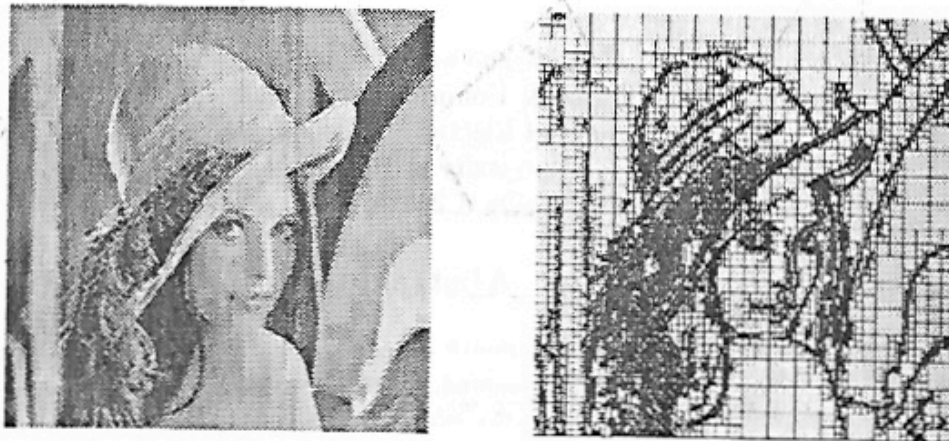


Figure 1: Quadtree based decomposition of Image Lenna

Recently, it has been suggested to use Binary Space Partitioning Trees to represent visual information [2]. This way, hyperplane boundaries can have any direction and recursively partition the image plane, while keeping a binary decomposition. The advantage of such a partitioning lies also in the fact that leaves of the tree represent any kind of convex region. In the context of image compression, it has been possible to encode image data at 0.1 bpp with an acceptable quality [3-4]. Noticeable gain can be achieved in this case due to the simplicity of both the boundary information representation (arbitrarily oriented lines+binary tree) and the texture information representation (simple parametric description).

Most often the decomposition is achieved on the basis of contour information or maximally constant behavior of the image data. The former technique uses a modified recursive Hough transform procedure [2], while the latter uses an least-square optimization approach [4].

This paper presents a method for adaptively segmenting an image in a recursive way to obtain a BSP tree on the basis of symmetry. From a recent proposal to use symmetry information in image data as a relevant feature (due to the frequent occurrence of symmetrical or quasi-symmetrical properties of objects in natural scenes) [5], it is proposed to recursively construct a BSP tree by identifying the line boundary which partitions a given region with the axis showing the highest symmetry. If the symmetry characteristics is sufficiently high, the partitioning is further iterated only to one side of the binary tree, so that the other side can be represented by mirroring the information in the one side. The partitioning process may also stop when reaching a near constant luminance region. Using this symmetry based decomposition of an image, a coding algorithm is developed.

In what follows, section 2 outlines the segmentation procedure, which implies the BSP representation, a region based symmetry extraction procedure, and a termination criterion for the recursive partitioning. Given a segmentation result, section 3 discusses some aspects related to the coding of the information, i.e. the tree structure, the partitioning lines, the "seed" nodes and "symmetrical prediction" [6]. Section 4 concludes the presentation with some future directions.

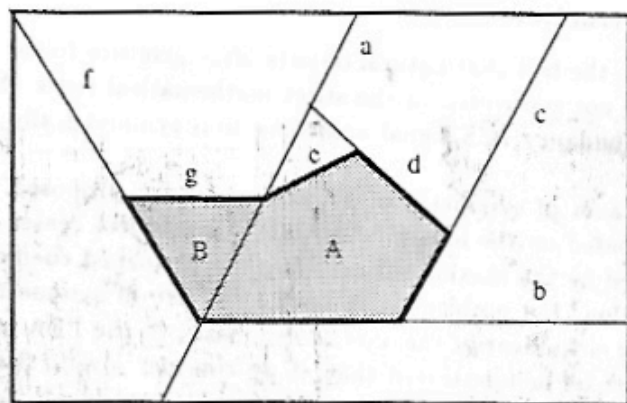


Fig. 2a

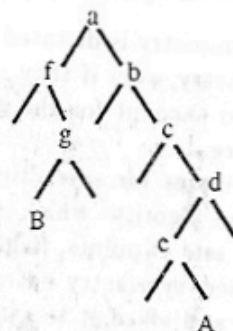


Fig. 2b

Figure 2: Simple BSP tree representation of a concave hexagon

## 2 Segmentation procedure

### 2.1 The BSP Tree Representation

One of the most promising approach for a very high compression is the so-called segmentation-based coding technique [7]. It involves three processing steps: segmentation, which split the image into various homogeneous components, coding of contours and coding of each region texture. Several studies have been reported in literature but a major drawback remains the segmentation operation.

An recent technique which reduces considerably the complexity of segmentation is achieved through the Binary Space Partition (BSP) of the image plane.

A BSP Tree is an abstract data type that provides a representation of a n-dimensional space through the use of recursive subdivision using arbitrarily oriented hyperplanes. In the 2D case, the subdivision is created using arbitrarily oriented lines. As the partition is obtained recursively on each half space, a binary tree is therefore generated.

Fig. 2a shows an example of a partition while Fig. 2b shows the corresponding BSP Tree. The root node represents the entire plane. A binary partition of the plane is formed by a labeled line ( $h$ ), resulting in a negative ( $h^-$ ) and a positive ( $h^+$ ) halfplane. These two halfplanes are represented respectively by left and right children of the root. A recursive binary partition of each of these two halfplanes may be performed (see Fig. 2). The non-leaf nodes of the tree will hold the equation of partitioning lines whereas the leaf nodes will represent an unpartitioned convex region, called cell. For any node of the tree, the corresponding region is defined by the union of the two halfplanes given by its left child and right child.

BSP tree achieves a simple and very flexible description of the image regions. BSP representation of images was introduced as a powerful tool for image manipulation and compression [2]. The decomposition was then based on identifying linear contour elements to the line based partitioning. In what follows, we wish to investigate the BSP decomposition on the basis of line symmetry. In this case, the partitioning lines should match axis of symmetry. If the region to be decomposed has no particular symmetry, the partitioning line will be chosen so as to distribute in the most symmetric fashion the luminance signal within the region.

## 2.2 Region based symmetry extraction

The idea to use symmetry is dictated by the fact that natural objects often give rise to the human sensation of symmetry, even if they are not symmetric in the strict mathematical sense. Symmetry can be used to account for the redundancy in a signal according to a symmetrically defined correlation measure.

Various techniques for extracting axes of symmetry of objects have been proposed. In [8], Marola presents an algorithm which is based on the identification of a given object center of mass and other related sets of points, followed by the maximization of a specially defined coefficient.

In region-based symmetry extraction, the problem is to detect the axis of symmetry of an object. A simple and efficient technique is to identify the axes of symmetry to the Principal Axes of Inertia (PAI) of a rigid body. It can be demonstrated that, if we consider a rigid 3D object, its Principal Axes of Inertia are identified by the eigenvectors of its inertia matrix. If the object is perfectly symmetric, these axes coincide with the axes of symmetry; if the object is not strictly symmetric these axes give the "best" partitioning according to a measure of symmetry.

The technique uses the inertia matrix and its eigenvectors to find the PAIs of a rigid body to detect the symmetry inside an image region  $D$  (2D object with a mass distribution defined by the luminance function inside the region). The inertia matrix is defined by

$$\begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \quad (1)$$

where  $I_{ij}$  are the moments of inertia with respect to 2 reference axes ( $i, j \in \{x, y\}$ ). If  $I(x, y)$  represents the luminance function we have

$$I_{ij} = \sum_{(x,y) \in D} i \cdot j I(x, y) \quad (2)$$

This approach allows to consider an image region as a 2D object, rather than a 3D object, defining this way a 2x2 inertia matrix rather than a 3x3 as in [5], which greatly reduces the computational complexity. It can be demonstrated that the eigenvector associated with the smallest eigenvalue of the 2D inertia matrix represents the direction of most likely axis symmetry of the image region. The symmetry axis location is defined by the fact that it passes through the center of mass of the 2D object.

Another way to select the axis of symmetry can be obtained using of a coefficient of symmetry measure. This measure gives an estimate of the degree of symmetry associate to each PAI as suggested in [8].

Let  $d$  be a symmetry axis and  $I(x, y)$  the luminance function. If  $P(x, y)$  and  $\bar{P}(\bar{x}, \bar{y})$  are two points symmetrically taken with respect to  $d$ , we define the symmetry coefficient  $\beta$  as follows:

$$-1 \leq \beta = \frac{\iint I(x, y) I(\bar{x}, \bar{y})}{\iint I(x, y)^2} \leq 1 \quad (3)$$

- If  $\beta = 0$  there is no symmetry.
- If  $\beta = 1$  there is even symmetry.
- If  $\beta = -1$  there is odd symmetry.

The higher the absolute value of  $\beta$  the higher the symmetry present in the image region with respect to that axis. With this approach however problems may occur when the ( $P$ ) point falls outside the

region of interest. Because the symmetry axis can be obtained through the PAI approach by choosing the eigenvector associated with the smallest eigenvalue of the inertia matrix, it is unnecessary to compute the value of  $\beta$ . Certainly, this measure could be modified to deal with non symmetric regions, but this falls beyond the scope of this work. This measure provides however a good way to distinguish odd symmetry from even symmetry.

### 2.3 Complete segmentation procedure

To speed up the computational complexity without losing the symmetry property, the input image is filtered using a Sobel operator. Pixels for which the gradient magnitude exceed a certain threshold are used in the computation of the inertia matrix. Once the PAI related to the smallest eigenvalue is detected, its equation is stored in the current tree node as the partitioning line of that node.

- If the axis splits the current region into two regions with a sufficient high symmetry characteristics, then only one of them is going to be further processed. In our implementation the recursion is applied to the left child (i.e., the negative region), while the right child will hold the information about the type of symmetry (even or odd, so as to use a proper prediction scheme for coding purposes). When the symmetry is odd, the PAI approach generates the direction of symmetry, but the location of the symmetry axis has to be shifted with respect to the center of mass of the image region.
- If the axis splits the current region into two regions with low degree of symmetry, both the associated children (right and left) are recursively processed.

The algorithm stops when a uniform or quasi-uniform luminance function has been reached in the corresponding region. (The algorithm could also have been stopped whenever the region size falls below a certain threshold.) Given the uniformity characteristics of the last layer cells (left and right), the mean value is the only parameter value assigned to such nodes. These leaf nodes are distinguished from the others as they become the seeds for the reconstruction algorithm.

At the end a BSP tree has been generated where the non-leaf nodes contain the equation of the axis of symmetry where pair of leaf nodes (i.e. the "seed" leaf nodes) are associated the mean value of the unfiltered original image luminance function. The unpartitioned right nodes (i.e. the non "seed" leaf nodes) contain a 1 bit flag specifying the type of symmetry (even or odd).

## 3 Coding strategy

Once the BSP tree representation is obtained, the coding procedure will be performed in 3 stages:

- Data structure coding: that is the binary tree structure defining the segmentation graph. It involves therefore 1 bit/tree node (Some bit could be spared if the region size gets smaller than a predetermined threshold so that no information is required for further division).
- The partitioning line equation coding: These lines will be used in the symmetrical prediction process. They can be coded by quantizing a parametric representation of such lines. Due to the symmetry characteristics and the limited size of the region that they intersect, bounds on the parameter exist which can be efficiently used in the quantization process.
- The leaf node coding: Here "seed" leaf node coding has to be distinguished from the other leaf node coding. At the "seed" leaf node level, only a mean value coefficient is quantized. As it is unlikely that pair of "seed" leaf nodes with same parent node have very different mean values, the right "seed" leaf node value is coded differentially with respect to his father.

For non leaf nodes, a symmetrical linear prediction scheme [6] is chosen, with no coding at this point of the quantization error. The prediction filter coefficients were obtained so as to minimize the square error between the input information (associated with the left node) and the symmetrically predicted output information. Both even and odd symmetries were considered. Quantization was then performed on the prediction coefficients.

On the 512x512 lena image, a sufficiently accurate BSP representation could be obtained with 1000 cell regions. Using this non optimized coding strategy, a 0.5 bpp rate could be estimated. Noticeable quantization artefacts are present due to the reconstruction procedure performed on the basis of symmetry information and no coding of the prediction error.

#### 4 Concluding remarks and future directions

This paper introduces a new approach to segmentation based coding, where the segmentation is obtained by identifying recursively symmetries in an image, with the use of a binary space partitioning tree. It tries to find more complex characteristics of the source data than contour information or uniform texture. This approach can be however attractive when local symmetries exist, but noticeable artefacts in the reconstruction process can occur if prediction errors are not coded, due to the breaking of contour lines across symmetry axis.

Many topics are currently being investigated: (1) better symmetrical prediction (use of alternating prediction with symmetrical filters [9]), (2) combining symmetrical based segmentation with contour based segmentation, (3) robust ways of detecting and making use of odd symmetry, (4) pruning of the symmetrically generated BSP tree on the basis of a rate-distortion criterion, to optimize the coding efficiency, (5) changing the stopping criterion in the partitioning process so as to freeze the decomposition whenever a uniform texture is reached, rather than having only uniform luminance regions at the "seed" leaf node level.

#### References

- [1] R. Leonardi & M. Kunt, "Adaptive Split and Merge for Image Analysis and Coding", *Proc. of the SPIE Conf. on Image Coding*, Cannes, 594: 2-9, Dec. 1985.
- [2] H. Radha, R. Leonardi, M. Vetterli, & B. naylor, "Binary Space Partitioning Tree Representation of Images", *Journal of Visual Communication and Image Representation*, 2(3):201-221, 1991.
- [3] H. Radha, M. Vetterli, & R. Leonardi, "Coding of Images Using BSP Trees", *Proceedings of the 1993 Picture Coding Symposium*, Mar. 1993.
- [4] H. Radha, *Efficient Image Representation Using Binary Space Partitioning Trees*, Ph.D. Thesis, Center for Telecommunications Research and Dept of Electrical Engineering, Columbia Univ., 1993.
- [5] P. Cicconi, R. Leonardi, & M. Kunt, "Symmetry-based Image Coding", *Proc. of the SPIE Conf. on Visual Communications and Image Processing*, 1818(III):1312-1323, Nov. 1992.
- [6] C. Saraceno, & R. Leonardi, "Symmetrical Linear Prediction", *Proceedings of the 8th Workshop on Image and Multidimensional Signal Processing*, Sep. 1993.
- [7] M. Kunt, A. Ikononopoulos and M. Kocher, "Second generation image-coding techniques", *Proceeding of IEEE*, 73: 549-574, Apr. 1985.
- [8] G. Marola, "On the Detection of the Axes of Symmetry of Symmetric and Almost Symmetric Planar Images", *IEEE Trans. on Pattern Anal. and Machine Intell.*, PAMI-11(1), Jan. 1989.
- [9] C. Saraceno & R. Leonardi, "Symmetrical Filters", submitted at the 1994 ICASSP Conference.