SIGNAL PROCESSING VII: Theories and Applications
M. Holt, C. Cowan, P. Grant, W. Sandham (Eds.)

BSP Tree Coding of Images Using Symmetry Information

C. Saraceno & R. Leonardi

Signals and Communications Lab.,
Dept. of Electronics for Automation,
University of Brescia, I-25123, Italy
Phone: + (39-30) 3715-434
FAX: + (39-30) 380-014
email: leonardi@icil64.cilea.it

Abstract. An image coding technique based on symmetry information and Binary Spaces Partitioning (BSP) Tree representation for still pictures is here presented. A good encoding strategy that may find useful application for very low bit-rate coding is proposed. The segmentation process combines a boundary based partition of the image with an axial decomposition which uses instead symmetry information.

1. Introduction

There has been a growing interest in creating binary tree partitions of the image domain, for image manipulation, representation and compression. Many reasons justify this choice: the simple data structure, that allows an easy coding procedure; the nice ability to navigate through this data structure; the capability to relate the physical nature of information with this representation of images.

On the other hand, efforts have been made to use local features that exhibit certain geometrical properties.

In particular, the use of symmetry information present in images has been suggested as a way of reducing redundancy [1-2]. This seems quite adequate as symmetric or almost symmetric features are frequently present in nature or in most man-made objects, even if these are not necessarily symmetric in a strict mathematical sense.

This article presents a new technique to code images using symmetry information in conjunction with a segmentation process that uses such symmetry information and which leads to a BSP Tree data structure.

The advantages and limits of such a technique are presented below.

2. Segmentation procedure

Prior to any segmentation process, it is necessary to address the issue of extracting local symmetry information in an image.

2.1 Symmetry extraction

Various techniques for extracting axes of symmetry from objects have been proposed. For a region-based symmetry extraction procedure, the problem can be easily solved by detecting the Principal Axes of Inertia (PAI's) of a rigid body. In fact, it can be demonstrated that if we consider a 3-D symmetric object, at least one of its two PAI's coincides with the axis of symmetry of the object. Even if the object is not symmetric in a strict mathematical sense, sometimes it can be considered as "almost" symmetric. In such a case at least one of its two PAI's divides the object into two mostly symmetric parts.

It is known that the PAI's of a rigid body are identified by the eigenvectors of its inertia matrix.

We can consider an image region of support D as a 2-D object with a mass distribution defined by the luminance function of the image within such region. In this case, the inner matrix can be expressed by:

$$I=\begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

where $I_{ij}=\sum_{(x,y)\in D} i\cdot j\cdot I(x,y)$ are the moments of inertia of the image with respect to the two reference axes $i,j\in\{x,y\}$, $I(x,y)$ is the luminance function of the image and D is its support.

In practice, PAI's are calculated on the basis of the edge information rather than the luminance function. This choice has been dictated so as to extract a contour symmetry of the object while avoiding any disturbance introduced by some shadowing artefact. The PAI's extraction procedure defines two orthogonal directions (the eigenvectors) which coincide with the PAI's directions with respect to the chosen reference axes. When the two reference axes go through the center of the mass of the image then at least one of the two PAI's coincides with the most likely axis of symmetry.

To decide which of the two PAI's is the most likely axis of symmetry and whether a symmetry exists indeed, a measure of symmetry has been defined and a tolerance error measure must be proposed to specify the meaning of "almost" symmetric.

The measure of symmetry used in this work is given by the Marola coefficient of symmetry [3] associated to a given axis and calculated on the basis of the luminance function. In general it is defined as follows:

$$-1\le\beta=\frac{\iint_{(x,y)\in D}\left[I(x,y)I(\bar{x},\bar{y})\right]}{\iint_{(x,y)\in D}\left[I(x,y)^2\right]}\le 1$$

if d is the considered axis, let $P(x,y)$ be a point of the region and $P(\bar{x},\bar{y})$ its symmetric one with respect to d, then $I(x,y)$ and $I(\bar{x},\bar{y})$ will be their respective luminance values. In particular:

- if $\beta = 0$ there is no symmetry;
- if $\beta = 1$ there is an even symmetry;
- if $\beta = -1$ there is an odd symmetry.

The higher the absolute value of β the higher the symmetry present in the image region with respect to that axis.

Since PAI's detect only even symmetries we have $0 \leq \beta \leq 1$.

With this approach, however, problems may occur when symmetric points ($P(\bar{x},\bar{y})$) fall outside the region of interest (as in **Figure 1**).

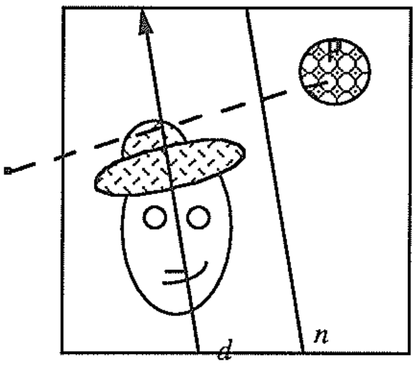

Figure 1. An example of a local symmetry with parts of one region not having symmetric counterparts belonging to the support D. *n* defines the new axis that limits the left-hand region so as to obtain two symmetric regions with respect to the symmetry axis *d*.

In this case one sub-region has many symmetric points falling outside. Another axis has to be calculated to bind the symmetric area to the non-symmetric one. This new axis splits one of the region into two parts: a symmetric one with respect to the PAI axis, while the other does not contain any symmetry. This new axis is selected parallely to the axis of symmetry, so as to define a sub-region between itself and the axis of symmetry *d* which is indeed symmetric with respect to *d*.

At this point, a segmentation algorithm can be suggested.

The procedure starts to consider the whole picture as an object. It detects the axis of symmetry and the associate coefficient of symmetry by dividing the picture into two regions. If the two regions are symmetric enough, only the bigger one is processed, while the other one will be obtained by some "symmetrical" prediction (see section 3). On the other hand, if the axis has a low coefficient of symmetry, the two regions cannot be considered as symmetric; the algorithm will continue to process recursively the two regions, searching for eventual local symmetries. The procedure ends when a homogeneous area is found or the region is too small.

In practice, it is not suitable to divide regions if no symmetry is found, so the recursion is stopped if the last N recursions are not leading to any new symmetric region (typically with N=5).

This technique using PAI's to detect symmetry information in an image works well if there is a global symmetry, i.e. the algorithm detects a symmetry at the first iteration. If there are only local symmetries, a bias is introduced in the recursion. In fact the first PAI divides the image incorrectly, and this initial bias leads further in the recursion to an incorrect estimation of the remaining local symmetries.

2.2 BSP Tree data structure

In the previous section we discussed how a region is divided into two sub-regions by an axis, using symmetry information. The most natural data type to represent this kind of line based decomposition is a Binary Space Partitioning (BSP) tree.

A BSP tree is an abstract data type that provides a representation of an n-dimensional space through a recursive subdivision using arbitrarily oriented hyperplanes. In the 2-D 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. The **Figure 2** shows an example of a partition and the corresponding BSP Tree:

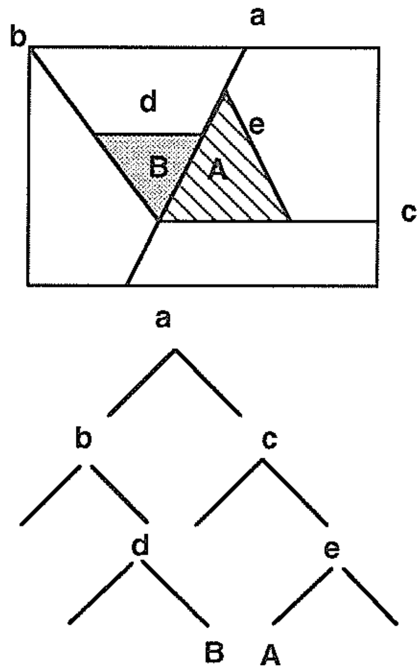

Figure 2. An example of an image which is decomposed into a BSP tree.

The root node represents the whole region. A binary partition of the plane is formed by a line resulting into two half-planes. These two half-planes are represented by the left and right children of the root, respectively. A recursive binary partition of each of these two half-planes may be performed. The (inner) nodes of the tree contain the equation of the partitioning lines whereas the leaf nodes, called cells, represent unpartitioned convex regions. For any inner node of the tree, the corresponding region is defined by the union of the two convex region represented by its left and right child.

2.3 Complete segmentation procedure

To avoid shadowing artefacts, the input image is first filtered to extract the contours of the image. The axis of symmetry is calculated on the basis of such edge information, while the coefficient of symmetry is estimated using the luminance function of the original image. The equation of the detected axis is stored in the current node of the tree as the partitioning line of that node.

If the axis splits the current region into two sub-regions with a sufficiently high symmetry characteristic, one tries to predict one region with respect to its symmetric counterpart [4]. The energy of this prediction error (the residue) is then calculated. If this energy is small enough only the bigger region is further decomposed, as the other can be reconstructed with the "symmetrical prediction" strategy. Otherwise both regions are decomposed using the same strategy.

If instead the absolute value of the coefficient of symmetry is not close enough to "1", a bounding axis *n* is used to bind the bigger region. *n* is parallel to the axis of symmetry *d*, and maximizes the magnitude of the coefficient of symmetry defined between the smaller region generated by *d* and the bounded region (the one which is bounded by *d* and *n*). The smaller symmetric region is then labelled as symmetric.

If no symmetry is found the algorithm divides recursively the two sub-regions separated by *d*, while regions labelled as symmetric are not further processed, but are predicted from their symmetric counterpart.

The algorithm ends when

- a uniform or almost uniform luminance function has been reached in the corresponding region;
- the region is too small;
- the last N (N = 5) parents of a given region have not lead to any symmetric axis.

As an example we consider the **Figure 3** representing an image and the corresponding BSP tree.

The first axis of symmetry, labelled as 1 divides this image into two almost symmetric regions. However, due to the presence of other objects the coefficient of symmetry is not close enough to 1, so in the bigger region (the left one) an axis parallel to the axis labelled as 1 is detected to bind the symmetric region to a non symmetric one. This new axis is labelled as 2 and it is calculated considering a series of parallel axis to 1, which are contained in the region located left to 1. It is selected at the location which leads to the highest coefficient of symmetry with respect to 1. At this point the smaller symmetric region is the one defined between the axes labelled as 2 and 1. It will be predicted from the right hand region with respect to the axis labelled as 1. The segmentation procedure continues by looking for future symmetries in the regions that are located to the right of the axis labelled as 1 and to the left of the axis labelled as 2.

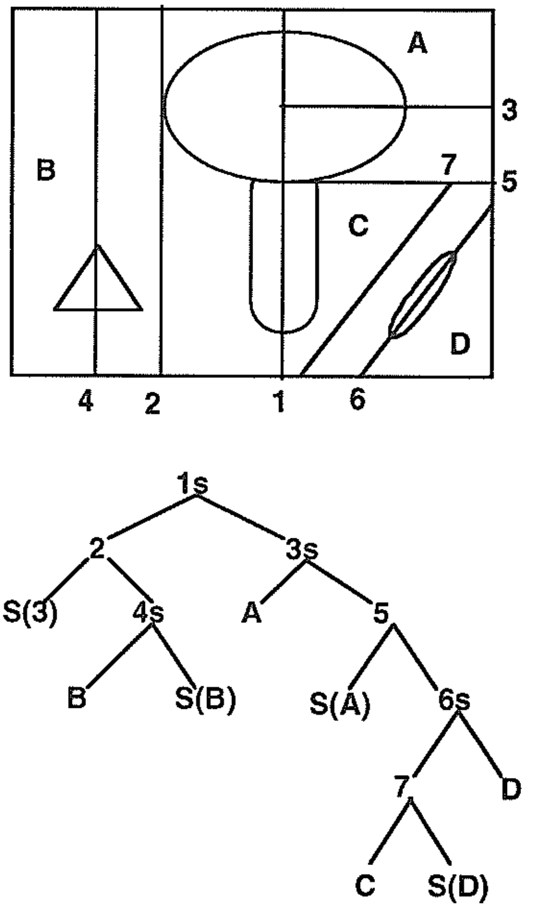

Figure 3. An image segmented according to the procedure described previously, and its corresponding BSP tree. Leaf nodes labelled as S(X) can be predicted symmetrically with respect to the regions labelled as X.

Figure 4 indicates the result of the segmentation procedure on a real image.

3. Coding strategy

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

1) Data structure coding: that is, the binary tree structure defining the segmentation graph. It involves 1 bit/tree node.

2) The partitioning line equation coding. 1 bit is needed to identify a symmetric line from a non symmetric one. The line equation is coded by quantizing its parametric representation. The limited size of the region that it splits binds its parameter range, so that an efficient quantization process can be developped [5].

3) The leaf node coding. If a region has a uniform or almost uniform luminance function, only the mean value is coded. Non uniform regions will be coded as suggested in Radha et al. [5]. A boundary based decomposition is in this case performed on these regions to obtain a BSP sub-tree that represents them.

Compression performance are difficult to estimate at this point but they can be expected to be of the same order of magnitude of the one obtained using a simple boundary based BSP tree representation of an image.

4. Decoding strategy

From a coded BSP Tree representation a simple decoding procedure can be developped. This one analyzes the tree with a breadth-first navigation.

The algorithm begins reading the root node. If the root node is not a leaf the stored line detects two sub-regions. Two pointers to these two regions are stored into a FIFO stack. The first pointer stored on the stack is popped. If the node is a leaf, it can represent a symmetric or a non symmetric region. The non symmetric regions are decoded first and reconstructed.

The regions labelled as symmetric are reconstructed by prediction of their symmetric counterpart. This region (the counterpart) is the other child of the same parent, if the parent holds a symmetric axis. If the parent is not labelled as symmetric, the symmetric region is the other child of the parent of its parent. As an example see the decomposition shown in **Figure 3**.

If we want to reconstruct a predicted region, but its symmetric is still not coded, the pointer to the predicted one pushes it back into the stack to be processed later.

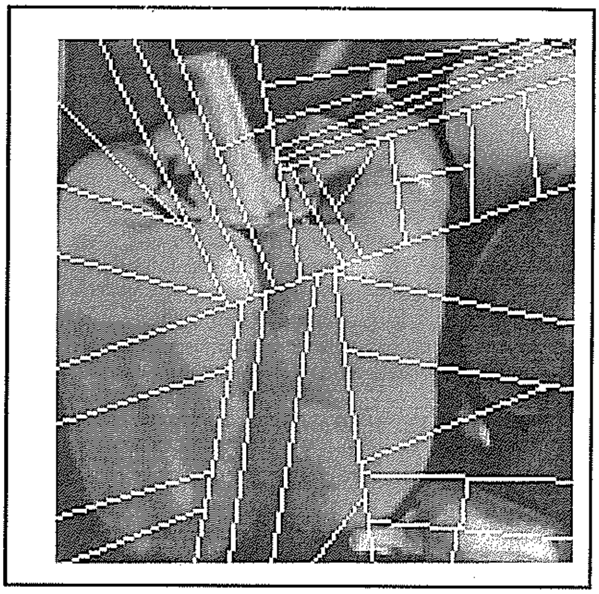

Figure 4. Symmetry based BSP representation of a real image.

5. Concluding remarks and future directions

This paper introduces a new approach to segmentation based coding, where segmentation is obtained by identifying symmetry in a image. Unfortunately, the first idea to use the PAI to detect symmetries is not always efficient as a wrong axis creates a bias in the extraction of any local symmetry. Next step will be to improve the detection of symmetry using other mathematical tools. A complete coding/decoding algorithm will be implemented.

6. References

[1] P. Cicconi, R. Leonardi & M. Kunt: "Symmetry-based image coding", *Proc. of the VCIP'92 Conf.*, Boston, SPIE-1818(3): 1312-1323, Nov. 1992.

[2] C. Saraceno & R. Leonardi: "Symmetrical Segmentation-Based Image Coding", *Proc. of the 1993 SPIE Visual Communications and Image Processing Conf.*, Boston, SPIE-2094(I):31-36, 1993.

[3] G. Marola: "On the detection of axis of symmetry of symmetric and almost symmetric planar images", *IEEE Trans. on Pattern Analysis & Machine Intel.ligence*, PAMI-11(1): 104-108, Jan.1989.

[4] C. Saraceno & R. Leonardi: "Symmetrical Linear Prediction", *Proc. of the IMDSP'93 workshop*, Cannes, 28-29, 1993.

[5] H. Radha, M. Vetterli & R. Leonardi: "Coding images using BSP trees", to be submitted to *IEEE Trans. on Image Processing*.