Chapter 1

Introduction

1.1 Motivation

In the last years, image processing has become an increasingly active field of development. The situation is motivated by the day to day increasing multimedia applications and by the increasing utilization of the World Wide Web. In most image processing based applications, an image is usually viewed as a set of pixels placed on a rectangular grid. Most of the developed work has been focused on the basis of this point of view. One of the main difficulties that arise when working directly with a pixel based representation is the large number of pixels composing the image or video sequence. Thus only rather simple algorithms may be applied on them. For instance, the classical DPCM coding strategy is an example of such type of processing. Moreover, in the JPEG [98] and MPEG-1 [30] standards, also devoted to coding purposes, the image or frame to code is divided into blocks of constant size without taking into account the spatial organization of the image. In the recent years, there is an increasing interest of interpreting the image as composed of a set of arbitrarily shaped regions or objects. The image is not understood anymore as a set of pixels, but as a set of regions composing the image. This has taken, for instance, to the development of the MPEG-4 [13] or MPEG-7 [83] standards. Both standards interpret the image or video sequences as a set of audiovisual objects. Thus new representations of the image – not pixel based – should be developed in which the notion of region is implicitly present. In other words, an abstraction from pixel to region should be done. Moreover, appropriate techniques that are able to deal with these representations should be developed.

We are going next to review the two structures that have demonstrated to be useful in the field of region based image processing: region adjacency graph and tree based structures. Both are graph representations in which nodes are used to represent regions – the smallest information unit – and the links are used to represent certain relationships between regions. We now discuss the way such structures are used in the field of region based processing as

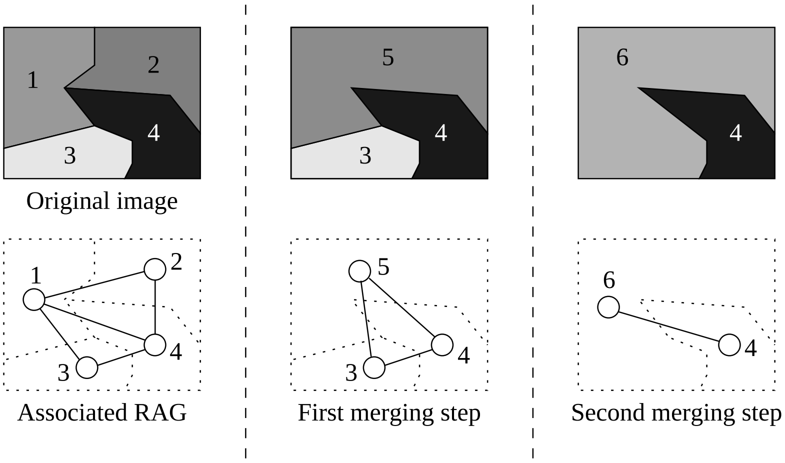

Figure 1.1: Region merging technique on Region Adjacency Graph.

well as their advantages and drawbacks. Both structures are compared in Sec. 1.1.3 in order to reason why tree based representations are better suited for our purpose.

1.1.1 Region Adjacency Graph based processing

Increasing attention is being paid to graph based structures. The region adjacency graph is the most well known region oriented structure. The *Region Adjacency Graph* is a graph which is constituted by a set of nodes representing regions of the space and a set of links connecting two spatially neighboring nodes. The region adjacency graph is usually used to represent a partition of the image. Note that a node of the graph can represent a region, a flat zone (a connected component of the space where the signal is constant, see Sec. 2.1 for a formal definition of flat zone) or even a single pixel. Processing techniques relying on region adjacency graphs have mainly focussed on merging techniques. The graph is constructed based on an initial partition: each region of the partition image is associated to a node in the graph, and two nodes are connected if their associated regions are neighbors in the partition image. A merging algorithm on such a graph is simply an iterative process that removes some of the links and merges the corresponding nodes. The merging order, that is, the order in which the links are processed is usually based on a similarity criterion between regions. Such homogeneity criterion may be based on color, motion, depth, etc. Each time a link is processed its associated nodes (i.e. regions) are merged together. After each merging the algorithm has to look for the links whose distance has to be recomputed. The merging ends once a termination criterion is reached. Most commonly used termination criterion is the number of nodes associated to the graph.

Fig. 1.1 shows an example of region based processing using region adjacency graphs.

The original image and its associated region adjacency graph are shown on the left. The algorithm begins by measuring the homogeneity value between each pair of linked nodes, that is, between each pair of neighboring regions. Assume, for instance, that regions 1 are 2 are the most similar according the measured homogeneity criterion. The merging algorithm decides to merge regions 1 and 2 which is represented with a new node labelled 5. The homogeneity distance has then to be recomputed for the links associated to regions 5 and 3, and regions 5 and 4. The merging algorithm then decides to merge regions 3 and 5. The merging process may stop here if the termination criterion states that merging should stop when the region adjacency graph has two nodes. The resulting image is shown on the top-right corner of Fig. 1.1. Note that the merging techniques applied to region adjacency graphs are flexible in the sense that at each step of the merging process the algorithm may decide which type of homogeneity criterion may be applied. For example, the merging algorithm may begin merging the regions using a homogeneity criterion based on color. At a certain point of the merging process, one may change to a motion based criterion.

1.1.2 Tree based processing

Another suitable structure for representing the images using regions relies on *trees*. A tree is an acyclic graph. It has one root node, and from each node several nodes may emerge. These nodes are called children, whereas the node to which they are linked is called the parent node. Nodes from which no children emerge are called leaf nodes. Tree representations are well suited for representing data in a hierarchical way. The links of the tree indicate “containment” or “is composed of”. Tree representations are usually used to provide access to information or data in an easy and fast way.

One may take, for example, the table of contents of a book. The table of contents is usually divided in chapters, sections, subsections and so on. Assume we want to represent the table of contents using a tree structure. The title of the book can be considered as being associated to the root node of such tree. The chapters are associated to the children of the root node, whereas the sections are the children of the associated chapters nodes. Other possible examples of tree structures of every day use are, for example, the directory structures that are used to organize files on a hard disc.

In a similar way, tree structures may be used in image analysis to represent the structure of the image in a hierarchical and compact way. Fig. 1.2 illustrates an example of image representation using trees. Assume that the images shown on top correspond to a decomposition of the original image into a set of regions. It should be noted that such decomposition does not correspond to a partition hierarchy. In fact, the image shown on the top-right corner of Fig. 1.2 is not a partition, since each pixel of the image is not classified into a particular class (see Sec. 2.1 for a formal definition of partition). The decomposition is interpreted as follows: the whole image is represented by region R_1, which in turn is composed of regions R_2 and

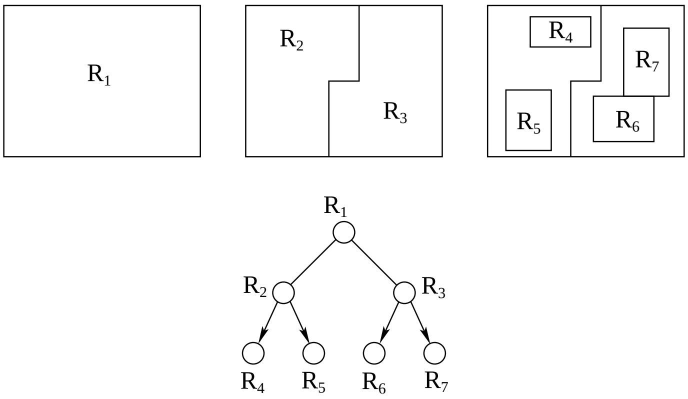

Figure 1.2: Example of tree structure. On top, a possible decomposition of an image is shown. The associated tree representation is shown on bottom.

R_3. Regions R_4 and R_5 are included in region R_2, whereas regions R_6 and R_7 are included in region R_3. This structural relationship can be represented easily with a tree, as shown on bottom of Fig. 1.2.

Processing techniques on trees are mainly based on pruning techniques. Pruning techniques remove some of the branches of the tree based on an analysis algorithm applied on the nodes of the tree. One may, for instance, assess a certain criterion on each node of the tree and then select by means of a threshold which nodes have to be removed and which should be kept in the final tree.

Fig. 1.3 shows an example of tree pruning. On top, the example tree of Fig. 1.2 is shown. Assume that the analysis algorithm decides to prune regions R_4 and R_5 which is equivalent (in this particular case) to remove its associated nodes[1]. We have thus removed the detail associated to region R_2.

1.1.3 Tree vs. Region Adjacency Graph

Note that the region adjacency graph structure is oriented towards coding the neighborhood relationships between regions, whereas a tree structure is oriented towards the coding of inclusion relationship. As such, the tree is not able to code neighborhood relationship. Note, for example, that one cannot infer directly from the tree structure the neighborhood relationship between regions R_4 and R_5 or regions R_6 and R_7 of Fig. 1.2.

On the other hand, the tree structure is inherently hierarchical. By analyzing the structure of Fig. 1.2 one may say that its associated image is composed of two regions, R_2 and R_3. Depending on the application, this may give us enough level of detail of the image. We may

[1] We will see in Chap. 2 that the interpretation of "pruning" depends on what the tree represents itself.

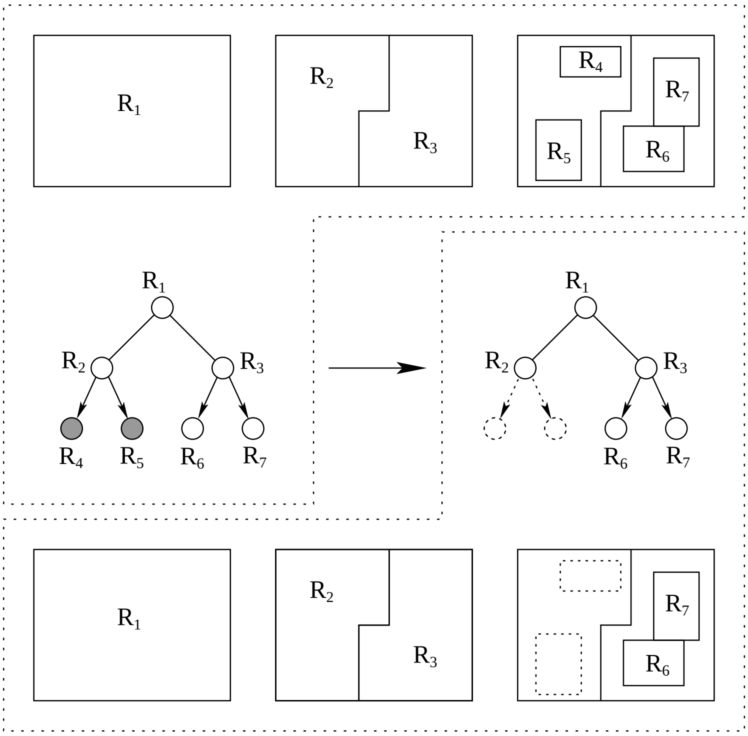

Figure 1.3: Example of tree pruning. On the left, the tree of Fig. 1.2 is shown. Nodes to be removed are marked in gray. On the right, the pruned tree with its associated image decomposition is shown.

say also, for example, that the image is composed of regions R_2 and R_3 in which regions R_4, R_5 and R_6, R_7 are included. The level of detail we are considering now is higher than before. The maximum level of detail is reached if we consider the leafs of the tree.

Tree based image processing offer several advantages over region adjacency graph based processing:

- **Efficiency**: one of the main differences between tree and region adjacency graph based processing is that on trees the merging order of the nodes is *fixed a priori*, whereas on region adjacency graph the merging order has to be computed during the merging process. On a region adjacency graph, each time two nodes are merged together the homogeneity values associated to the resulting merged node and its neighbors have to be recomputed. On the other hand, the pruning steps that are performed on the tree structure do not modify the processing order. This key difference allows to implement fast algorithms on a tree.

- **Hierarchy**: trees represent the image in a hierarchical way. Region adjacency graph based approaches are only able to represent the spatial relationships between regions for a certain resolution level. By contrast, the tree based representation gives information of how each region is composed. However, in the tree structured representation neighborhood relationships may have been lost, only the most important and representative relationships for the particular application are represented. The hierarchical property of the tree implies that the latter may represent the image in a multi-resolution way.

- **Processing techniques**: the tree structured representation allows to deal with complex techniques due to the fixed merging order. On a region adjacency graph, it is difficult to undo the mergings that have been already done. By contrast, for a tree based representation, due to its fixed structure, it is possible to "see" what would happen if some nodes are pruned without the need of really pruning them. This fact gives to the tree based structure a much wider range of applications and possibilities.

- **Compromise**: The tree structured representation is able to deal with complex processing techniques but is not flexible. On the other hand, region adjacency graphs have a higher flexibility but do not allow to handle with complex processing techniques. There is therefore a compromise between flexibility and the ability of dealing with complex techniques.

The work developed in this thesis discusses the usefulness of tree representations for region based image processing purposes. Its usefulness will be demonstrated in particular through the development of efficient algorithms.

1.2 General objectives

Our purpose in this work is to discuss the usefulness of tree based structures for hierarchical region based processing. For that purpose, two major issues have to be discussed.

- **Hierarchical representation**: a tree describes an image in a hierarchical way. It is thus necessary to show how it may be created from an image. As the reader may have noticed, an image does not have a unique tree representation. In fact, the tree representation should be defined according to the type of applications at which it is targeted.

- **Hierarchical processing**: processing deals with the way the tree is manipulated. In this work we focus only on pruning techniques. That is, techniques in which some subtrees of the original tree are removed according to a criterion assessed on the nodes of the tree. The pruning techniques that have been developed in this work lead to the following applications: filtering, segmentation and content based image retrieval algorithms.

An important issue in this work is the computation efficiency. The developed algorithms (tree creation and processing) should be fast enough to allow easy use of the presented techniques. Therefore efficient algorithms are presented in this work.

1.3 Thesis organization

This section briefly details the organization of the present document. In this thesis we have proceeded in three steps. First, in Chap. 2, some mathematical notions are presented, the state of the art of tree based processing is reviewed and the scheme used to process images using hierarchical structures is discussed. Then, the tree construction, in particular the one of the Max-Tree and the Binary Partition Tree, is studied in Chap. 3 and Chap. 4 respectively. Finally, Chap. 5 and Chap. 6 develop several processing techniques that may be applied on the tree structures. Chap. 5 focuses on filtering and segmentation techniques, whereas Chap. 6 presents a content based region retrieval system.