



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA**

**Department of Signal Theory and Communications**

**Ph.D. Dissertation**

**Hierarchical Region Based Processing of  
Images and Video Sequences:  
Application to Filtering, Segmentation and  
Information Retrieval.**

**Author: Luis Garrido Ostermann**

**Advisor: Prof. Philippe Salembier Clairon**

Barcelona, April 2002



# Abstract

This work discusses the usefulness of hierarchical region based representations for image and video processing. Region based representations offer a way to perform a first level of abstraction and reduce the number of elements to process with respect to the classical pixel based representation. In this work the two representations that have demonstrated to be useful for region based processing are reviewed, namely region adjacency graphs and trees, and it is discussed why *tree based representations* are better suited for our purpose. In fact, trees allow representing the image in a hierarchical way and efficient and complex processing techniques can be applied on it. Two major issues are discussed in this work: how the hierarchical representation may be created from a given image and how the tree may be manipulated or processed.

Two tree based representations have been developed: the *Max-Tree*, and the *Binary Partition Tree*. The Max-Tree structures in a compact way the connected components that arise from all possible level sets from a gray-level image. It is suitable for the implementation of anti-extensive connected operators, ranging from classical ones (for instance, area filter) to new ones (such as the motion filter developed in this work). The Binary Partition Tree structures the set of regions that are obtained during the execution of a region merging algorithm. Developed to overcome some of the drawbacks imposed by the Max-Tree – in particular the lack of flexibility of the tree creation and the self-duality of the tree representation –, it has demonstrated to be a representation useful for a rather large range of applications, as it is shown in this work.

Processing strategies are focused 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. Pruning techniques applied on the Max-Tree lead to anti-extensive operators, whereas *self-dual operators* are obtained on the Binary Partition Tree, if the tree is created in a self-dual manner. The pruning techniques that have been developed in this work are directed to the following applications: *filtering*, *segmentation* and *content based image retrieval*.

The filtering (in the context of connected operators) and segmentation applications are based on the same principle: the nodes of the tree are analyzed according to a fixed criterion, and the decision to remove or preserve a node usually relies on a threshold applied on the

former measured criterion. Pruning is then performed according to the previous decision. As a result, the image associated to the pruned tree represents a filtered or segmented version of the original image according to the selected criterion. Some of the criteria that are discussed in this work are based, for instance, on area, motion, marker & propagation or a rate-distortion strategy. The problem of the lack of robustness of classical decision approaches of *non-increasing criteria* is discussed and solved by means of an optimization strategy based on the Viterbi algorithm.

Content based image retrieval is the third application we have focused on in this work. Hierarchical region based representations are particularly well suited for this purpose since they allow to represent the image at different scales of resolution, and thus the regions of the image can be described at different scales of resolution. In this work we focus on an image retrieval system which supports low level queries based on visual descriptors and spatial relationships. For that purpose, region descriptors are attached to the nodes of the tree. Two types of queries are discussed: *single region query*, in which the query is made up of one region and, *multiple region query*, in which the query is made up of a set of regions. In the former visual descriptors are used to perform the retrieval whereas visual descriptors and spatial relationships are used in the latter case. Moreover, a relevance feedback approach is presented to avoid the need of manually setting the weights associated to each descriptor.

An important aspect that has been taken into account throughout this work is the *efficient implementation* of the algorithms that have been developed for both creation and processing of the tree. In the case of the tree creation, efficiency has been obtained mainly due to the use of hierarchical queues, whereas in the processing step analysis algorithms based on recursive strategies are used to get efficient algorithms.

# Resumen

Este trabajo estudia la utilidad de representaciones jerárquicas basadas en regiones para el procesamiento de imagen y de secuencias de vídeo. Las representaciones basadas en regiones ofrecen una forma de realizar un primer nivel de abstracción y reducir el número de elementos a procesar con respecto a la representación clásica basada en el píxel. En este trabajo se revisan las dos representaciones que han demostrado ser de utilidad para el procesamiento basado en regiones, a saber el grafo de regiones adyacentes y el árbol, y se discute por qué las *representaciones basadas en árboles* son más adecuadas para nuestro propósito. De hecho, los árboles permiten la representación de la imagen de forma jerárquica y pueden ser aplicadas sobre éstas técnicas eficientes y complejas. En este trabajo se discuten dos cuestiones principales: cómo puede ser creada la representación jerárquica a partir de una imagen determinada y cómo se puede manipular o procesar el árbol.

Se han desarrollado dos representaciones basadas en árboles: el *Árbol de Máximos*, y el *Árbol de Particiones Binario*. El *Árbol de Máximos* estructura de forma compacta las componentes conexas que surgen de todos los posibles conjuntos de niveles de una imagen de nivel de gris. Es una representación adecuada para la implementación de operadores conexas antiextensivos, desde operadores clásicos (por ejemplo, filtro de área) hasta operadores nuevos (como el filtro de movimiento desarrollado en este trabajo). El *Árbol de Particiones Binario* estructura el conjunto de regiones que se obtiene durante la ejecución de un algoritmo de fusión basado en regiones. Desarrollado para superar alguno de los inconvenientes impuestos por el *Árbol de Máximos* – en particular la falta de flexibilidad de la creación del árbol y la auto-dualidad de la representación del árbol –, ha demostrado ser una representación apta para un gran número de aplicaciones, tal y como se muestra en este trabajo.

Las estrategias de procesamiento se basan en técnicas de *poda*. Las técnicas de poda eliminan algunas ramas del árbol basándose en un algoritmo de análisis aplicado a los nodos del árbol. Las técnicas de poda aplicadas al *Árbol de Máximos* permiten obtener operadores antiextensivos, mientras que para el caso del *Árbol de Particiones Binario* se obtienen *operadores auto-duales* si éste ha sido creado de forma auto-dual. Las técnicas de poda desarrolladas en este trabajo están dirigidas hacia las siguientes aplicaciones: *filtrado*, *segmentación* y *recuperación de datos basada en el contenido*.

Las aplicaciones de filtrado (en el contexto de los operadores conexos) y segmentación están basados en el mismo principio: los nodos del árbol son analizados de acuerdo a un criterio determinado, y la decisión de eliminar o preservar un nodo se basa normalmente en un umbral aplicado sobre la anterior medida del criterio. La poda se realiza entonces de acuerdo con la ésta decisión. Como resultado, la imagen asociada al árbol podado representa una versión filtrada o segmentada de la imagen original de acuerdo con el criterio seleccionado. Alguno de los criterios discutidos en este trabajo están basados, por ejemplo, en área, movimiento, marcador & propagación o una estrategia de tasa-distorsión. El problema de la falta de robustez de las estrategias clásicas para *criterios no crecientes* es estudiado y solucionado gracias a un algoritmo de optimización basado en el algoritmo de Viterbi.

La recuperación de imágenes basada en el contenido es la tercera aplicación en la que nos hemos centrado en este trabajo. Las representaciones jerárquicas basadas en regiones son particularmente adecuadas para este propósito ya que permiten representar la imagen a diferentes escalas de resolución, y por lo tanto las regiones asociadas a una imagen pueden ser descritas a diferentes escalas de resolución. En este trabajo nos centramos en un sistema de recuperación de imágenes que soporta preguntas de bajo nivel basadas en descriptores visuales y relaciones espaciales. Para ello, se adjuntan descriptores de región a los nodos del árbol. Se discuten dos tipos de preguntas: *pregunta basada en una región*, en el que la pregunta esta formada por una región, y *pregunta basada en múltiples regiones*, en el que la pregunta esta formada por un conjunto de regiones. En el primero la recuperación se realiza utilizando descriptores visuales, mientras que en el segundo se utilizan descriptores visuales y relaciones espaciales. Además, se presenta una estrategia de realimentación por relevancia para eludir la necesidad de establecer manualmente el peso asociado a cada uno de los descriptores.

Un aspecto importante que se ha tenido en cuenta a lo largo de este trabajo es la *implementación eficiente* de los algoritmos desarrollados tanto para la creación como el procesado del árbol. En el caso de la creación del árbol, la eficiencia se obtiene principalmente gracias al uso de colas jerárquicas, mientras que en el procesado se utilizan algoritmos basados en estrategias recursivas para obtener algoritmos eficientes.

# Agradecimientos

El trabajo presentado en este documento se inscribe dentro del marco del proyecto *Consultation Thématique Informelle* (CTI96-ME22, Octubre 1996 a Diciembre 1999): Caracterización automática del contenido semántico de secuencias de vídeo aplicado a la indexación y la búsqueda de información. Ha sido financiado por France Telecom CNET (actualmente France Telecom R&D) y en el han participado el Centro de Morfología Matemática (Fontainebleau), la Universidad Politécnica de Cataluña (Barcelona) e INRIA (Rocquencourt).

Este documento es fruto de la investigación realizada durante estos últimos cinco años. Ha sido un largo recorrido en el que he podido formarme tanto profesionalmente como personalmente. Quisiera por ello agradecer aquí a toda aquella gente que ha contribuido, directa o indirectamente, a la elaboración de este trabajo. En particular quisiera agradecer a las siguientes personas su implicación en esta tesis.

A Philippe Salembier por confiar en mi al encargarme este trabajo. Por su dedicación e interés durante todo este tiempo. Es una persona excelente, con la que he podido aprender mucho.

A todo el Grupo de Imagen del departamento por el buen ambiente dentro del éste, y por tener la puerta de sus despachos siempre abierta para cualquier problema que tuviera.

A Henri Sanson y a los participantes del proyecto CTI por su dedicación y buenos consejos para llevar a buen término este trabajo.

Qué sería una tesis sin compañeros de doctorado. Con ellos he estado trabajando estos últimos años. Ellos han animado el día a día y de ellos tendré muchos y buenos recuerdos.

Finalmente, quisiera agradecer también a mis amigos y mi familia todo el apoyo recibido durante éstos años.





# Notation

The most frequently used notation is listed below. Other notation appearing in this document is valid only in the section it is used.

$\bigvee$	supremum
$\bigwedge$	infimum
$p = (p_x, p_y)$	pixel position $(p_x, p_y)$
$\mathcal{N}_E(p)$	neighbor pixels of $p$ in grid $E$
$X, Y$	binary set
$CC_p(X)$	connected component of $X$ containing $p$
$f, g$	function or image
$NG$	maximum possible gray-level value (usually $NG = 255$ )
$f(p)$	function value at pixel position $p$
$X_h(f)$	upper level set with parameter $h$
$X^h(f)$	lower level set with parameter $h$
$UpCC_h^k$	$k$ 'th connected component of $X_h(f)$
$LowCC_k^h$	$k$ 'th connected component of $X^h(f)$
$\mathcal{P}$	partition of the image
$\mathcal{P}^{Fz}$	partition of flat zones of the image
$N_{\mathcal{P}}$	number of regions of partition $\mathcal{P}$
$\psi(X)$	binary operator applied on set $X$
$\Psi(f)$	operator applied on function $f$
$\Psi^*(f)$	dual operator of $\Psi(f)$
$R_k$	region with index $k$
$N_k$	node with index $k$
$N_k \rightarrow N_j$	node $N_k$ is parent of node $N_j$

$A_R$	area in pixels of region $R$
$\partial R$	perimeter of region $R$
$O(R_1, R_2)$	merging order between $R_1$ and $R_2$
$C(R_1, R_2)$	merging criterion between $R_1$ and $R_2$
$M_R$	region model associated to region $R$
$M_R(p)$	region model evaluated at pixel position $p$
$\mathcal{M}(R)$	criterion value of region $R$
$\lambda$	threshold on criterion value $\mathcal{M}(R)$
$Q$	query region
$T$	target region
$\mathcal{Q} = \{Q_i\}$	multiple query region
$\mathcal{T} = \{T_i\}$	multiple target region
$NR_{\mathcal{Q}}$	number of regions of $\mathcal{Q}$
$D_{desc}(R)$	“desc” descriptor associated to region $R$

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.1.1	Region Adjacency Graph based processing . . . . .	2
1.1.2	Tree based processing . . . . .	3
1.1.3	Tree vs. Region Adjacency Graph . . . . .	4
1.2	General objectives . . . . .	6
1.3	Thesis organization . . . . .	7
<b>2</b>	<b>General Framework</b>	<b>9</b>
2.1	Basic definitions . . . . .	9
2.2	Hierarchical region based processing using trees . . . . .	12
2.2.1	Base terminology . . . . .	13
2.2.2	Notation and properties . . . . .	14
2.2.3	Pruning . . . . .	15
2.3	State of the Art . . . . .	17
2.3.1	Quadtree . . . . .	17
2.3.2	Partition Trees . . . . .	18
2.3.3	Critical Lake Tree . . . . .	19
2.3.4	Area Tree . . . . .	22
2.3.5	Inclusion Tree . . . . .	24
2.3.6	Discussion . . . . .	27
2.4	Objectives and contribution of the thesis . . . . .	28
2.5	General framework . . . . .	29
2.5.1	Tree construction (Part I) . . . . .	29
2.5.2	Tree processing (Part II) . . . . .	30

<b>I</b>	<b>Tree construction</b>	<b>33</b>
<b>3</b>	<b>Max-Tree</b>	<b>35</b>
3.1	Definition . . . . .	35
3.2	Min-Tree . . . . .	41
3.3	Discussion . . . . .	41
3.4	Efficient implementation . . . . .	42
3.4.1	Algorithm . . . . .	42
3.4.2	Tree complexity . . . . .	44
3.4.3	Performance . . . . .	46
<b>4</b>	<b>Binary Partition Tree</b>	<b>51</b>
4.1	Definition . . . . .	51
4.2	General Merging Algorithm . . . . .	52
4.2.1	Definition . . . . .	52
4.2.2	Merging Algorithm . . . . .	54
4.3	BPT construction with the General Merging Algorithm . . . . .	55
4.3.1	Gray-level homogeneity . . . . .	55
4.3.2	Color homogeneity . . . . .	57
4.3.3	Motion homogeneity . . . . .	57
4.3.4	Forcing support of nodes . . . . .	61
4.4	Discussion . . . . .	62
4.5	Efficient implementation . . . . .	63
4.5.1	Graph structure . . . . .	63
4.5.2	Recursivity . . . . .	64
4.5.3	Hierarchical Queues and Binary Search Trees . . . . .	65
4.5.4	Performance . . . . .	68
<b>II</b>	<b>Tree processing &amp; Applications</b>	<b>71</b>
<b>5</b>	<b>Filtering &amp; Segmentation</b>	<b>73</b>
5.1	Objectives . . . . .	73
5.1.1	Segmentation Algorithms . . . . .	73
5.1.2	Connected operators . . . . .	74
5.1.3	Discussion . . . . .	78

---

5.2	Tree processing strategy . . . . .	79
5.2.1	Pruning . . . . .	79
5.2.2	Reconstruction . . . . .	80
5.3	Pruning strategy . . . . .	82
5.4	Increasing and Non-Increasing Criteria . . . . .	83
5.5	Optimization for Non-Increasing Criteria . . . . .	87
5.6	Discussion . . . . .	95
5.7	Examples . . . . .	96
5.7.1	Marker & Propagation Segmentation . . . . .	96
5.7.2	Area Filtering . . . . .	105
5.7.3	Contrast Filtering . . . . .	108
5.7.4	Complexity Filtering . . . . .	112
5.7.5	Motion filtering . . . . .	115
5.7.6	Rate & Distortion Browsing . . . . .	129
5.8	Performance . . . . .	133
5.9	Conclusions . . . . .	135
<b>6</b>	<b>Information Retrieval</b> . . . . .	<b>137</b>
6.1	Objectives . . . . .	137
6.2	Supported Queries . . . . .	139
6.3	Tree processing strategy . . . . .	139
6.4	Region Descriptors . . . . .	142
6.4.1	Color . . . . .	142
6.4.2	Geometry . . . . .	143
6.5	Single region query . . . . .	148
6.5.1	Query strategy . . . . .	148
6.5.2	Region similarity . . . . .	148
6.5.3	Overall similarity assessment between two regions . . . . .	154
6.5.4	Distance normalization . . . . .	154
6.5.5	Results . . . . .	155
6.6	Multiple region query . . . . .	162
6.6.1	Query strategy . . . . .	162
6.6.2	Query object normalization . . . . .	162
6.6.3	Structural similarity . . . . .	164

---

6.6.4 Overall similarity . . . . .	166
6.6.5 Discussion . . . . .	167
6.6.6 Search algorithm . . . . .	168
6.6.7 Results . . . . .	172
6.7 Relevance feedback . . . . .	180
6.7.1 Motivation and objective . . . . .	180
6.7.2 Update of weights . . . . .	181
6.7.3 Results . . . . .	184
6.8 Conclusions . . . . .	190
<b>7 Conclusions</b>	<b>193</b>
7.1 Contribution . . . . .	193
7.2 Future research . . . . .	194
<b>Bibliography</b>	<b>205</b>