Roberto Medeiros de Souza

“Maximal Max-Tree Simplification”

“*Simplificação Maximal da Árvore Máxima*”

CAMPINAS
2014

University of Campinas
School of Electrical and Computer Engineering

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Roberto Medeiros de Souza

"Maximal Max-Tree Simplification"

"Simplificação Maximal da Árvore Máxima"

Supervisor/Orientador
Prof. Dr. Roberto de Alencar Lotufo
Co-supervisor/Co-orientador
Profª. Drª. Letícia Rittner

MSc Dissertation presented to the Post Graduate Program of the School of Electrical and Computer Engineering of the University of Campinas to obtain a MSc degree in Electrical Engineering.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Mestrado em Engenharia Elétrica.

This volume corresponds to the final version of the Dissertation defended by Roberto Medeiros de Souza, under the supervision of Prof. Dr. Roberto de Alencar Lotufo.

Este exemplar corresponde à versão final da Dissertação defendida por Roberto Medeiros de Souza, sob orientação de Prof. Dr. Roberto de Alencar Lotufo.

Supervisor's signature

Assinatura do Orientador(a)

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

	Medeiros, Roberto de Souza, 1989-
M467m	Maximal max-tree simplification / Roberto Medeiros de Souza. – Campinas, SP : [s.n.], 2014.
	Orientador: Roberto de Alencar Lotufo. Coorientador: Letícia Rittner. Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.
	1. Árvores (teoria dos grafos). 2. Filtros digitais (Matemática). I. Lotufo, Roberto de Alencar,1955-. II. Rittner, Letícia. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

<u>Informações para Biblioteca Digital</u>

Título em outro idioma: Simplificação maximal da árvore máxima
Palavras-chave em inglês:
Trees (graph theory)
Digital filters (Mathematics)
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Roberto de Alencar Lotufo [Orientador]
Luciano de Fontoura Costa
Neucimar Jerônimo Leite
Data de defesa: 28-03-2014
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Roberto Medeiros de Souza

Data da Defesa: 28 de março de 2014

Título da Tese: "Simplificação Maximal da Arvore Máxima"

Prof. Dr. Roberto de Alencar Lotufo (Presidente): ______________________

Prof. Dr. Luciano da Fontoura Costa: ______________________

Prof. Dr. Neucimar Jeronimo Leite: ______________________

Abstract

The Component Tree is a data structure that represents an image through the hierarchical relationship of its connected components. It is an adequate structure to implement connected filters, and it has been successfully used in many applications. The Max-Tree is a compact structure for the Component Tree representation.

The main contribution of this work is the proposal of the Maximal Max-Tree Simplification (MMS) filter with two possible criteria to compute the filter: a normalized threshold criterion (MMS-T) and a Maximally Stable Extremal Regions (MSER) criterion (MMS-MSER). A methodology to apply the MMS filter in association to the Extinction filter, which is formally defined in this work, is presented. It is shown that after applying our simplification methodology, which sets the number of relevant maxima in the image to be kept, the number of nodes in the simplified Max-Tree is at most twice this number. In order to define the MMS filter, new concepts, such as composite node and sub-branch are introduced. These concepts are important to define many Max-Tree algorithms, and they have interesting interpretations in terms of image processing. Possible applications of the methodology proposed, such as text location, object recognition, and image simplification/segmentation are illustrated to demonstrate the potential of this methodology. Also, exploratory studies, such as detection of distinguished regions in the image, and analysis of the robustness of the Max-Tree topology are presented.

Keywords: Max-Tree; Component Tree; Composite node, Sub-branch; Extinction Filter; MSER; Maximal Max-Tree Simplification.

Resumo

A Árvore de Componentes é uma estrutura de dados que representa uma imagem através da relação de hierarquia de seus componentes conexos. Ela é uma estrutura adequada para a implementação de filtros conexos e que foi utilizada com sucesso em muitas aplicações. A Árvore Máxima é uma estrutura compacta para a representação da Árvore de Componentes.

A principal contribuição deste trabalho é a proposta do filtro de Simplificação Maximal da Árvore Máxima (MMS) com dois possíveis critérios para efetuar o seu cálculo: um critério de limiarização normalizada (MMS-T) e um critério de Regiões Extremais Maximamente Estáveis (MMS-MSER). Uma metodologia para aplicar o filtro MMS em associação com o filtro de Extinção, que é formalmente definido nesse trabalho, é apresentada. É mostrado que após a aplicação da metodologia de simplificação, a qual escolhe o número de máximos relevantes a serem mantidos na imagem, o número de nós da Árvore Máxima simplificada é no máximo duas vezes o número de máximos mantidos. Para definir o filtro MMS, novos conceitos, como nó composto e sub-ramo são apresentados. Esses conceitos são importantes para definir muitos algoritmos da Árvore Máxima, e eles possuem interpretações interessantes em termos de processamento de imagem. Possíveis aplicações da metodologia proposta, tais como localização de texto, simplificação/segmentação de imagens e reconhecimento de objetos são ilustrados para mostrar o potencial da metodologia. Também, estudos explortatórios de detecção de regiões salientes em imagens e análise da robustez da topologia da Árvore Máxima são apresentados.

Palavras-chave: Árvore Máxima; Árvore de Componentes; Nó composto; Sub-ramo; Filtro de extinção; MSER; Simplificação Maximal da Árvore Máxima.

Contents

Abstract **vii**

Resumo **ix**

Dedication **xiii**

Acknowledgements **xv**

1 Introduction **1**

1.1 Motivation 2
1.2 Objectives 3
1.3 Contributions 3
1.4 Organization of the Thesis 4

2 Theoretical Background **5**

2.1 Graph Theory basic Definitions 5
2.2 Image Processing Definitions 7
2.3 Conclusions 11

3 Structures for Image Representation: Component Tree and Max-Tree **12**

3.1 Component Tree 12
3.1.1 Component Tree Representation 13
3.1.2 Component Tree Attribute Signature 14
3.2 Max-Tree 16
3.2.1 Max-Tree Representation 16
3.2.2 Max-Tree Filtering 17
3.3 Sub-branch 21
3.4 Conclusions 24

4 Extinction Filter **25**
4.1 Computing Extinction Values from the Max-Tree 25
4.2 Extinction Filter 27
4.3 Relationship between Extinction Filters and Attribute Filters 33
4.4 Conclusions 36

5 Maximal Max-Tree Simplification **37**
5.1 MMS Filter 37
5.1.1 MMS-T 38
5.1.2 MMS-MSER 38
5.2 Analysis of the MMS Filter 39
5.3 Methodology 39
5.4 Conclusions 43

6 Applications and Exploratory Studies **45**
6.1 Applications 45
6.1.1 Text Location and Recognition in Natural Scenes 45
6.1.2 Object Recognition 51
6.1.3 Segmentation 51
6.2 Exploratory Studies 55
6.2.1 Analysis of the Max-Tree Topology Robustness 55
6.2.2 Detection of Distinguished Regions 58
6.3 Conclusions 63

7 Conclusions **64**
7.1 Conclusions 64
7.2 Future works 65
7.3 Publications 65

A Repeatability Results **69**

B Processing Time **76**

I dedicate this work to my family, friends, and all the loved ones that are no longer here.

Acknowledgements

I would like to thank my family Jorge (father), Suzana (mother), Gustavo (brother), Yara (aunt), Neusa (grandmother), and Ilze (sister-in-law) for all the support throughout these years. They have always been there for me when I needed. Jonathas, Ramon, Renan and Antonio, old friends from Belém. Mariana, Suelen, Jomara, Wallace, Augusto, Paul, Rafael, Rodrigo, André and Victor, friends that I made in Campinas. Professor Lotufo and Professor Letícia, my advisors, they helped me a lot during these two years. Also, I would like to thank CAPES for the financial support and FEEC/UNICAMP for the good infra-structure provided.

Chapter 1

Introduction

The Max-Tree is a data structure proposed by Salembier et al. (1998). It represents an image through the hierarchical relationship of its connected components. In fact, each node of the Max-Tree stores a set of flat zones of the image, but its corresponding connected component can be recovered by aggregating the pixels the node stores with the pixels its descendants store. There are algorithms to build the Max-Tree in quasi-linear time (Najman and Couprie, 2004, 2006; Carlinet and Géraud, 2012), and filtering the image consists simply in contracting some of the Max-Tree nodes, which is equivalent to merging sets of flat zones. Salembier et al. (1998) proved that contracting nodes of the Max-Tree is a connected filter, i.e. a filter that simplifies the image while preserving its contour information (Salembier and Serra, 1995), which for many applications is a desirable property.

Building and filtering the Max-Tree are fast operations suitable for real time applications. In fact, many morphological filters can be more efficiently implemented in the Max-Tree data structure. For instance, Fabrizio and Marcotegui (2009) proposed an algorithm to perform the ultimate opening using the Max-Tree, which runs much faster than the usual implementation that uses a sequence of openings with increasing structuring elements. The Max-Tree is also a very adequate structure for computing Maximally Stable Extremal Regions (MSER), which are a set of distinguished regions proposed by Matas et al. (2002, 2004). MSER regions are used in many applications, such as stereo matching (Matas et al., 2002, 2004), tracking of objects (Bischof, 2006), computation of shape descriptors for object recognition (Forssen and Lowe, 2007), among others.

Vachier (1995a) proposed the concept of Extinction values, which can be seen as an extension of the concept of dynamics (Grimaud, 1992). Extinction values are a powerful tool to measure the persistence of a crescent attribute, and are useful to discern relevant from irrelevant extrema, usually noise. The most common extinction values attributes used are the height, area and volume, and they can be efficiently computed in the Max-

Tree structure. Silva and Lotufo (2008) proposed new crescent attributes that can have their extinction values computed in the Max-Tree. These attributes are the number of descendants, the sub-tree height, and the height and width of the connected component bounding box. These new attributes were used to count objects in images and perform quality inspection.

Other applications that the Max-Tree was successfully used were detection, tracking and recognition of license plates (Donoser et al., 2007), recognition of text in natural scenes (Merino-Gracia et al., 2012), 3D Segmentation (Donoser and Bischof, 2006), interactive filtering and visualization of 3D medical images (Westenberg et al., 2007).

The Max-Tree is many times treated as a synonym of the Component Tree (Jones, 1999), but in fact they are distinct structures. The Max-Tree is a compact structure for the Component Tree representation, and because of that, the implementation of the same filters in these structures are different, which may cause confusion for unsuspecting readers.

The literature presents two groups of strategies for filtering the Max-Tree: pruning strategies and non-pruning strategies. In the pruning strategy, if a Max-Tree node is to be contracted, all its descendants will also be contracted. Filters based on increasing criteria, such as the *hmax* (Salembier and Oliveras, 1996) and area opening and closings (Vincent, 1994) can be efficiently implemented through pruning strategies on the Max-Tree.

In the non-pruning strategy, if a Max-Tree node is to be contracted, its descendants do not necessarily have to be contracted. Filters based on non-increasing criteria usually lead to non-pruning strategies. This kind of filter strategy is not much explored in the literature, since many authors do not consider it robust (Salembier et al., 1998). Therefore, most authors try to circumvent this problem by transforming the non-pruning strategy into a pruning strategy filter. In order to do that, Salembier et al. (1998) proposed the *min*, the *max*, and the *Viterbi* rules.

Non-pruning strategies are implemented through the direct rule for node contraction (Salembier et al., 1998) or the subtractive rule for node contraction (Urbach and Wilkinson, 2002). To the best of our knowledge, applications of filters based on non-pruning strategies are not explored by the scientific community, because, as mentioned before, many authors do not consider them robust. The Maximal Max-Tree Simplification filter proposed in this work is based on a non-pruning strategy.

1.1 Motivation

The motivation for this work is the fact that we believe that the Max-Tree is a very powerful data structure, which has the potential for solving many practical problems, such as image segmentation, detection of distinguished regions, iterative visualization, object recognition, tracking and many others. Although its great potential, the Max-Tree is still not much explored by the scientific community. Our motivation is to investigate further the Max-Tree, and introduce new concepts, filters, and explore possible applications, where it can achieve competitive results, when compared to other methods in the literature.

1.2 Objectives

The main objective of this work is to define the Maximal Max-Tree Simplification (MMS) filter, which is based on a non-pruning strategy, with two possible criteria to compute it: a normalized threshold criterion (MMS-T) and a Maximally Stable Extremal Regions (MSER) criterion (MMS-MSER). Other objectives are the formalization of the Extinction filter, which is a simplification filter based on a pruning strategy that keeps only the persistent extrema of the image, and preserves their height, and the development of a simplification methodology that applies the MMS filter in association to the the Extinction filter. It is shown that after applying the simplification methodology, which sets the number of most relevant maxima to be preserved, the number of nodes in the simplified Max-Tree is at most twice this number. One last objective is to show the potential of the methodology proposed to solve real problems, such as object recognition, image segmentation/simplification, text location, and detection of robust image distinguished regions. In order to define the MMS filter, it is presented the Max-Tree and its relation to the Component Tree with a clear notation. New concepts, such as composite node, and sub-branches are introduced. They are important to establish the relationship between these data structures, and are useful to define many Max-Tree algorithms.

1.3 Contributions

The contributions of this dissertation are listed below:

- Formal definition of the Extinction filter, and analysis of its effects on the image and the Max-Tree;

- Proposal of the MMS filter with a normalized threshold criterion (MMS-T) and a Maximally Stable Extremal Regions (MSER) criterion (MMS-MSER), along with a methodology to apply it;

- Proof that the number of nodes in the Max-Tree after applying the methodology proposed is at most twice the number of leaves;

- Illustration of the potential of the methodology proposed by indication of possible applications, such as text location, object recognition and image segmentation;
- Exploratory studies concerning the robustness of the Max-Tree topology and detection of image distinguished regions using the methodology proposed.

1.4 Organization of the Thesis

This thesis is organized as follows: Chapter 2 presents a minimal theoretical background about graphs and image processing necessary to understand the Max-Tree, the Component Tree and the tools that will be developed in this work. Chapter 3 presents the Max-Tree, and the Component Tree with a consistent notation, and their differences and similarities are analyzed. Also, the composite node and sub-branch concepts are introduced. Chapter 4 formalizes the Extinction filter, and compares it with the usual simplification filters. Chapter 5 defines the MMS filter, and presents a methodology to apply it. Chapter 6 illustrates applications and exploratory studies of the methodology proposed. Chapter 7 states our conclusions, and future works.

Chapter 2

Theoretical Background

In this chapter, it is presented a theoretical background about graph theory and basic image processing concepts necessary to explain the Max-Tree, the Component Tree, and the tools that will be developed in the subsequent chapters.

2.1 Graph Theory basic Definitions

Definition 2.1 ***Undirected Graph:*** *An undirected graph can be seen as a set of nodes, V, and a set of edges, E, represented by non-ordered pairs of nodes. Usually a graph G is denoted by $G(V, E)$.*

Definition 2.2 ***Path:*** *A path between nodes i and j is a sequence of edges connecting these two nodes.*

Definition 2.3 ***Simple path:*** *A path with no repeated nodes is called a simple path.*

Definition 2.4 ***Cycle:*** *A cycle is defined as being a path that starts and ends at the same node.*

Definition 2.5 ***Connected Component:*** *A connected component of an undirected graph is a maximal set of nodes in which any two nodes are connected to each other by paths.*

Definition 2.6 ***Connected Graph:*** *A graph is said connected if there is a path from any node to any other node in the graph.*

Definition 2.7 ***Tree:*** *A tree is an undirected graph G that is connected and has no cycles.*

An undirected graph G with $V = \{0, 1, 2, 3, 4, 5\}$ and $E = \{(1, 2), (1, 3), (2, 3), (3, 4), (3, 5)\}$ is illustrated in Figure 2.1. The graph is not connected, since node 0 is not connected to anyone, and the path joining nodes 1, 2, and 3 form a cycle. A tree is illustrated in Figure 2.2.

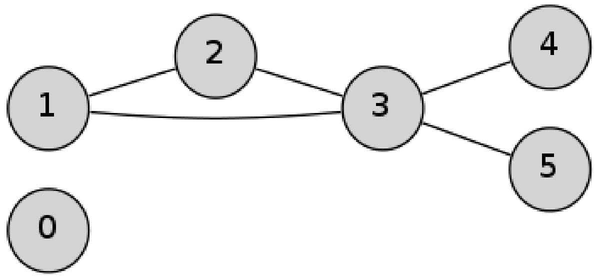


Figure 2.1: Illustration of a graph G with $V = \{0, 1, 2, 3, 4, 5\}$ and $E = \{(1, 2), (1, 3), (2, 3), (3, 4), (3, 5)\}$.

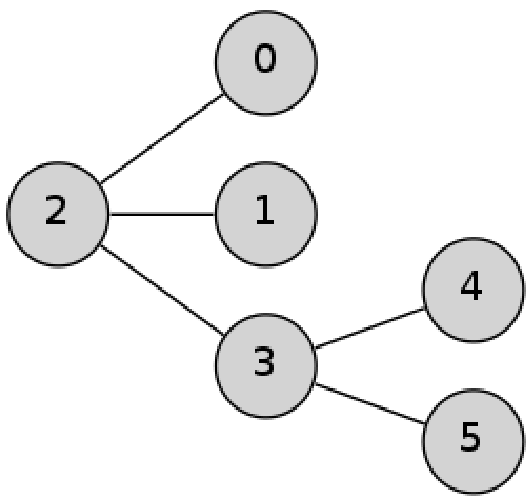


Figure 2.2: Illustration of a tree.

Definition 2.8 ***Rooted Tree:*** *A rooted tree, $\mathcal{T}$, is a tree in which one of its nodes has been designated the root, therefore its edges have a natural orientation, i.e. it is a directed graph. It has a partial ordering of its nodes. We say that $i \leq j$, if and only if the path from the root to i passes through j.*

Definition 2.9 ***Parent:*** *The parent of a node is the node connected to it on the path to the root.*

Definition 2.10 ***Child:*** *A child of a node i is a node of which i is the parent.*

Definition 2.11 ***Root:*** *It is the only node in the rooted tree that does not have a parent.*

Definition 2.12 ***Leaf:*** *It is a node in the tree that does not have children.*

Definition 2.13 ***Ramification:*** *It is a node with more than one child.*

Definition 2.14 ***Internal:*** *It is any node that is neither a leaf or the root of the tree.*

Definition 2.15 ***Sibling:*** *Two nodes are said to be siblings if they have the same parent.*

Definition 2.16 ***Descendant:*** *A descendant node i of a node j is a node such that j is in the path from i to the root of the tree.*

Definition 2.17 ***Ancestor:*** *An ancestor node of i is any node in the path from i to the root of the tree.*

Definition 2.18 ***Branch:*** *A branch is a set of nodes that join a leaf to the root node.*

A rooted tree is illustrated in Figure 2.3. The arrows point towards the parents. Node 0 is the root. Nodes 6 and 7 are leaves, nodes 3 and 4 are siblings. Node 2 is a ramification. The sequence $0-1-2-3-5-6$ constitutes a branch of the tree.

Definition 2.19 ***Binary Tree:*** *A binary tree is a rooted tree in which each of its nodes has at most two children. A binary tree is said to be full if each of its nodes is either a leaf or possesses exactly two child nodes. A full binary tree has $2n-1$ nodes, where n is the number of leaves in the tree.*

A full binary tree is illustrated in Figure 2.4.

2.2 Image Processing Definitions

Definition 2.20 ***Grey-scale image:*** *It is a function $I(z) : E \rightarrow k$, $E \in \mathcal{N}^2$ and $k \in \mathcal{Z}$. p is an ordered pair (z_{lin}, z_{col}) that represents a pixel of the image, and $I(z)$ is its luminosity intensity, usually $I(z) \in [0, L-1]$, $L > 1$. If $L = 2$, the image is said to be a binary image, the pixels with intensity equal 1 compose the foreground and the pixels with intensity equal 0 compose the background of the binary image.*

Definition 2.21 ***Negative of an image:***

$$Neg(I)(z) = (L-1) - I(z), \quad \forall z \in E \tag{2.1}$$

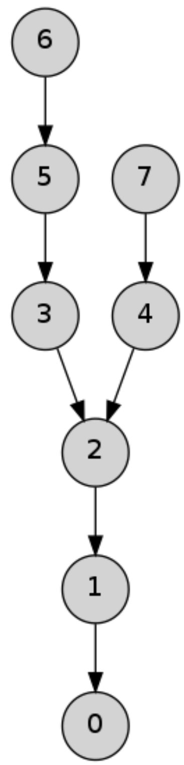


Figure 2.3: Illustration of a rooted tree. The arrows point towards the parents.

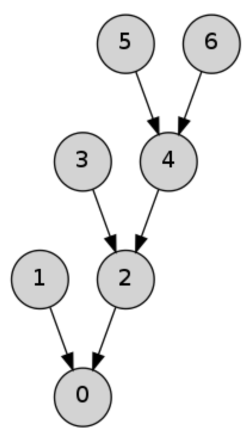
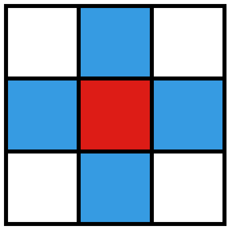
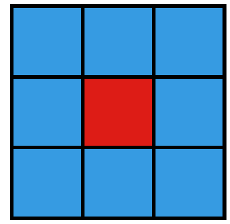


Figure 2.4: Illustration of a full binary tree. It has 4 leaves, therefore the tree has 7 nodes.

Definition 2.22 ***Image Thresholding:*** *The image thresholding is a procedure that given a threshold h, it transforms an image into a binary image. There are two cases the upper and the lower thresholding. They are expressed, respectively by the following*



(a) $C4$



(b) $C8$

Figure 2.5: Illustration of the neighborhoods $C4$ (a) and $C8$ (b) of a pixel. The blue pixels are the neighbors of the red pixels.

equations:

$$\mathcal{X}_h^{\geq}(z) = \begin{cases} 1 & \textit{if } I(z) \geq h \\ 0 & \textit{otherwise} \end{cases}, \tag{2.2}$$

$$\mathcal{X}_h^{\leq}(z) = \begin{cases} 1 & \textit{if } I(z) \leq h \\ 0 & \textit{otherwise} \end{cases}. \tag{2.3}$$

Definition 2.23 ***Neighborhood of a pixel:*** *The neighborhood of a pixel z corresponds to a set of coordinates $C(p) \subset E$ defined in relation to z in the image domain E. The most common pixel neighborhoods used are the neighborhoods $C4$ and $C8$ defined below:*

$$C4(z) = \{z + (-1, 0), z + (0, -1), z + (0, 1), z + (1, 0)\} \cap E \tag{2.4}$$

$$C8(z) = C4(z) \cup \{z + (-1, -1), z + (-1, 1), z + (1, -1), z + (1, 1)\} \cap E \tag{2.5}$$

The neighborhoods $C4$ and $C8$ of a pixel are illustrated in Figure 2.5.

Definition 2.24 ***Connectivity:*** *An image can be seen as a graph, where the pixels correspond to the nodes. Two pixels are connected, i.e. there is an edge joining them, if they are neighbors, according to the neighborhood used, that share a common property that defines a component. The property may be color, brightness, range of brightness values, or anything else of interest.*

Definition 2.25 ***Adjacency:*** *Two pixels z_1 and z_2 are adjacent, if they are connected.*

Definition 2.26 ***Upper threshold set:*** *The upper threshold set is the set of connected components $\{C_{h,1}, C_{h,2}, ..., C_{h,n_{CC}}\}$ that compose the foreground of the binary image resulting of the upper threshold $\mathcal{X}_h^{\geq}(I)$. The similarity criterion to define the connectivity is that the pixels have the same grey-level.*

An important property of level sets is that for each connected component $C_{h+1,m}$ resulting of the upper threshold $\mathcal{X}_{h+1}^{\geq}(I)$ there is a connected component $C_{h,n}$ resulting of the upper threshold $\mathcal{X}_{h}^{\geq}(I)$, such that $C_{h+1,m} \subseteq C_{h,n}$.

Definition 2.27 ***Flat zone:*** *Flat zones are connected components in which the similarity criteria used to define connectivity is that the pixels have the same grey-level.*

Definition 2.28 ***Regional Maximum:*** *A regional maximum of an image I is a flat zone M, if $I(z) > I(q)$, $\forall z \in M$ and q is any neighboring pixel of M.*

Definition 2.29 ***Partition*** *A partition $\mathcal{P}$ of a set X is a set of nonempty subsets of X such that every element x in X is in exactly one of these subsets.*

$$\mathcal{P}(X) = \{X_0, X_1, ...X_{n-1}\}, \quad X_i \cap X_j = \emptyset, \quad for \quad i \neq j \tag{2.6}$$

$$X = X_0 \cup X_1 ... \cup X_{n-1} \tag{2.7}$$

Definition 2.30 ***Grey-scale Image Ordering:*** *A grey-scale image f is said to be contained in a grey-scale image I of the same dimensions, $f \leq I$, if and only if:*

$$f[z] \leq I[z], \quad \forall z. \tag{2.8}$$

Definition 2.31 ***Anti-extensive Filter:*** *An anti-extensive filter ψ is a filter such that:*

$$\psi(I) \leq I, \quad \forall I. \tag{2.9}$$

Definition 2.32 ***Connected Filter:*** *is a filter in which the partition composed of the input image flat zones is always finer than the partition of the filtered image flat zones. The mathematical definition of a connected filter ψ is given by:*

$$\mathcal{P}_I \subseteq \mathcal{P}_{\psi(I)}, \quad \forall I. \tag{2.10}$$

Connected filters do not create new contours in the image and they do not modify the position of the existing contours.

Definition 2.33 ***Extinction value:*** *Consider M a regional maximum of an image I, and $\Psi = (\psi_\lambda)_\lambda$ is a family of decreasing connected anti-extensive transformations. The extinction value corresponding to M with respect to Ψ and denoted by $\varepsilon_\Psi(M)$ is the maximal λ value, such that M still is a regional maxima of $\psi_\lambda(I)$. This definition can be expressed through the following equation:*

$$\varepsilon_\Psi(M) = sup\{\lambda \geq 0 | \forall \mu \leq \lambda, M \subset Max(\psi_\mu(I))\}, \tag{2.11}$$

where $Max(\psi_\mu(I))$ is the set containing all the regional maxima of $\psi_\mu(I)$. Extinction values of regional minima can be defined similarly.

Definition 2.34 ***Image Region:*** *A region Q is a set of pixels, where each pixel has at least one neighbor pixel that belongs to the region.*

Definition 2.35 ***Boundary of a Region:*** *The boundary ∂Q_i of a region Q_i is the set of pixels that are neighbors to at least one element of the region, but do not belong to the region.*

Definition 2.36 ***Extremal Region:*** *A extremal region $Q \subset E$ is a region such that either for all $z \in Q$, $q \in \partial Q$: $I(z) > I(q)$ (maximum intensity region) or for all $z \in Q$, $q \in \partial Q$: $I(z) < I(q)$ (minimum intensity region).*

Definition 2.37 ***Maximally Stable Extremal Region:*** *Let Q_1,..,Q_{i-1},Q_i be a sequence of nested extremal regions, $Q_i \subset Q_{i+1}$. Extremal region Q_j is maximally stable if and only if j is a minimum of the following expression:*

$$q(j) = \frac{\mid Q_{j+\Delta} \setminus Q_{j-\Delta} \mid}{\mid Q_j \mid}. \tag{2.12}$$

MSER regions corresponding to a sequence of upper thresholds and lower thresholds are denominated $MSER+$ and $MSER-$, respectively. According to Matas et al. (2002, 2004), MSER regions are invariant to affine intensity transformations, they preserve the adjacency when submitted to continuous transformations, and they perform multi-scale detection, since the detected regions may be either large or small.

Definition 2.38 ***Structural Similarity index (SSIM) :*** *The SSIM index proposed by Wang et al. (2011) measures the structural similarity between images. It is defined by the following expression:*

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{2.13}$$

where x and y are images of the same dimensions μ_x, μ_y, σ_x^2, σ_y^2, and σ_{xy} represent the means, the variances and the covariance of images x and y. C_1 and C_2 are constants and their default values are 0.01 and 0.03, respectively.

2.3 Conclusions

This chapter presented an overview of basic image processing and graph concepts that will be used as a basis to explain the tools that will be developed in the remaining of this work.

Chapter 3

Structures for Image Representation: Component Tree and Max-Tree

The Component Tree was proposed by Jones (1999). It is a data structure that represents an image through the hierarchical relationship of its connected components. It separates the image filtering procedure on three steps: tree construction, filtering, and image restitution. The Max-Tree was proposed by Salembier et al. (1998). It is a compact structure for the Component Tree representation. Most filters are designed thinking in terms of the Component Tree, but they are usually more efficiently implemented in the Max-Tree.

This chapter starts by presenting the Component Tree data structure, and the notion of attribute signature. Then, the Max-tree is introduced, along with the concept of composite node, which helps defining Max-Tree based algorithms, and establishing the relationship between the Max-Tree and the Component Tree. The filtering procedure on the Max-Tree is explained, and its meaning on the Component Tree is analyzed. The sub-branch concept is presented. The chapter concludes by summarizing the differences between the Max-Tree and the Component Tree.

3.1 Component Tree

The Component Tree was proposed as a structure for image representation that represents all connected components resulting of all possible thresholds of the image, and that provides an attribute signature as means of discriminating features in the image. It is efficient to implement connected anti-extensive filters, and by duality extensive filters.

3.1.1 Component Tree Representation

The Component Tree, $\mathcal{T}_{CT}$, is a rooted tree in which each node stores a set of attributes. It is completely defined by its set of nodes $\mathcal{N}(\mathcal{T}_{CT})$:

$$\mathcal{N}(\mathcal{T}_{CT}) = \{0, 1, .., m-1\}, \tag{3.1}$$

and parents $\mathcal{P}(\mathcal{T}_{CT})$:

$$\mathcal{P}(\mathcal{T}_{CT}) = \{p_i | i \in (\mathcal{N} \setminus 0)\}, \tag{3.2}$$

where $\setminus$ is the set minus operand, and p_i is the parent of node i. Our convention is that node 0 is the root.

Suppose that the grey-levels of an image I are bounded between I_{min} and I_{max}. The upper threshold sets $\{C_{h,1}, C_{h,2}, ..., C_{h,n_{CC}}\}$ resulting of every upper threshold $\mathcal{X}_h^{\geq}(I)$ for h varying between I_{min} and I_{max} can be computed, and we know that threshold sets have the following property:

$$\forall C_{h+1,m}, \quad \exists C_{h,n} : C_{h+1,m} \subseteq C_{h,n}. \tag{3.3}$$

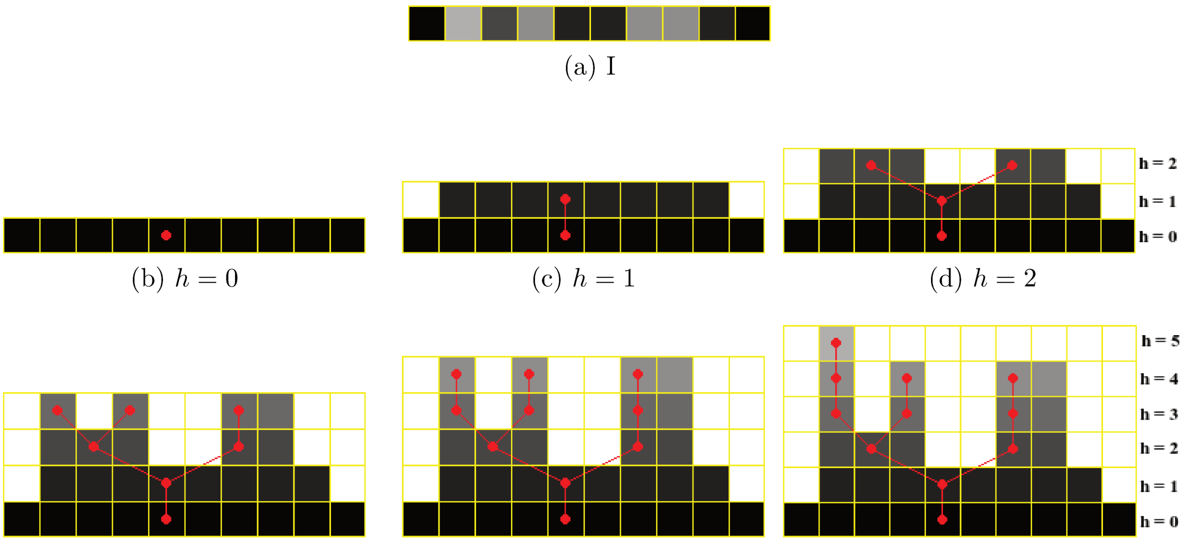
This hierarchy property allows a tree representation. For each connected component $C_{h,n}$ resulting of all threshold sets, there is a Component Tree node i with attributes $h_i = h$ and $C_i = C_{h,n}$. h_i and C_i are the basic attributes stored in the Component Tree nodes necessary to recover the image from the tree. Other attributes, such as height, area and volume can be computed and stored in the nodes during the tree construction or they can be computed just when required.

Node i is a parent of node j if C_i contains C_j, and $h_i = h_j - 1$. Note that the Component Tree may represent a pixel in more than one node, therefore it has some redundancy. The leaves of the Component Tree correspond to regional maxima in the image and the root represents the whole domain of the image.

The restitution of the image corresponding to a given Component Tree, is given by the following equation:

$$I(z) = max\{h_i | z \in C_i\}. \tag{3.4}$$

The construction of the Component Tree corresponding to the 1D image $I = [0, 5, 2, 4, 1, 1, 4, 4, 1, 0]$ is illustrated in Figure 3.1. This illustration is meant just to explain the Component Tree structure and construction, it does not correspond to the most efficient way to build it, for that, there are algorithms that run in quasi-linear time (Najman and Couprie, 2004, 2006).



(a) I

(b) $h = 0$ (c) $h = 1$ (d) $h = 2$

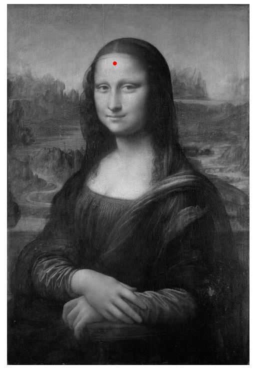
(e) $h = 3$ (f) $h = 4$ (g) $h = 5$

Figure 3.1: Component Tree construction of the 1D image $I = [0, 5, 2, 4, 1, 1, 4, 4, 1, 0]$ (a). Bottom-up construction (b)-(g). Resulting Component Tree (g)

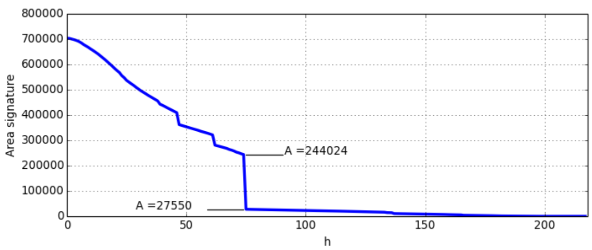
3.1.2 Component Tree Attribute Signature

The notion of attribute signature in the Component Tree is very important, it conveys information concerning the variation in shape or size of a connected component. The attribute signature uses the linking information between connected components at sequential grey-levels in the image to help the decision making process. The attribute signature of any two nodes i and j connected by a path in the Component Tree is simply the sequence of node attributes in this path.

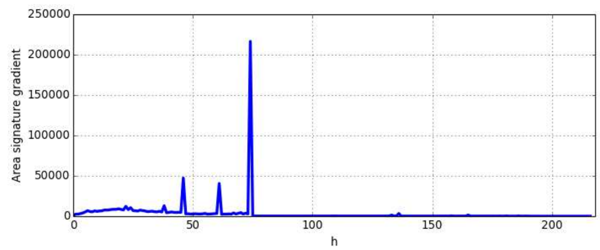
For instance, the attribute signature may be used to choose a threshold value to segment an object in an image. The Mona Lisa painting is depicted in Figure 3.2(a). The red dot in the image corresponds to a regional maxima, i.e. a Max-Tree leaf. The area signature, and its gradient starting at this leaf and ending at the root of the tree are shown in Figure 3.2(b) and Figure 3.2(c), respectively. The area signature presents some discontinuities, i.e. spikes in the gradient curve. The highest spike happens in the transition between $h = 75$ to $h = 76$. At $h = 75$, the area of the connected component is of 244024, in the following grey-level, $h = 76$, the area drops to 27550. The reconstruction of the connected components at levels 75 and 76 are depicted in Figure 3.2(d) and 3.2(e), respectively. It is clear that the connected component at level 75 merges Mona Lisa's face with a considerable part of the background, while the connected component at level 76 provides a better segmentation of Mona Lisa's face.



(a) Mona Lisa Painting

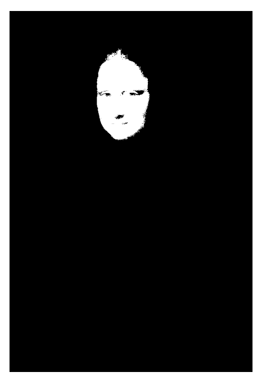


(b) Area signature



(c) Gradient of the area signature

(d) Reconstruction at $h = 75$



(e) Reconstruction at $h = 76$

Figure 3.2: Mona Lisa painting, the red dot corresponds to a regional maxima (a). Area signature (b), and its gradient (c). Reconstruction of the connected component at level 75 (d) and 76 (e).

3.2 Max-Tree

The Max-Tree can be seen as a compact structure for the Component Tree representation. Every operation that can be done in the Component Tree can be done in the Max-Tree and vice-versa.

3.2.1 Max-Tree Representation

The Max-Tree, $\mathcal{T}_{MT}$, is a rooted tree in which each node stores a set of attributes. It is completely defined by its set of nodes $\mathcal{N}(\mathcal{T}_{MT})$:

$$\mathcal{N}(\mathcal{T}_{MT}) = \{0, 1, .., m' - 1\}, \tag{3.5}$$

and parents $\mathcal{P}(\mathcal{T}_{MT})$:

$$\mathcal{P}(\mathcal{T}_{MT}) = \{p_i | i \in (\mathcal{N} \setminus 0)\}. \tag{3.6}$$

$\widehat{C_{h,n}}$ is defined by:

$$\widehat{C_{h,n}} = \{z \in C_{h,n} | I(z) = h\}, \tag{3.7}$$

it represents a set of flat zones (Serra and Salembier, 1993). For each non-empty $\widehat{C_{h,n}}$, there is a Max-Tree node i with $h_i = h$ and $\widehat{C_i} = \widehat{C_{h,n}}$. The pixels represented by $\widehat{C_i}$ and their level h_i are the minimum set of attributes that a node i has to store in order to be able to recover the image from the Max-Tree. Note that storing $\widehat{C_i}$ is much more memory efficient than storing C_i, there is no redundancy in the pixels stored. The connected component C_i can be recovered by:

$$C_i = \widehat{C_i} \cup \bigcup_{\forall j \in D(i)} \widehat{C_j}, \tag{3.8}$$

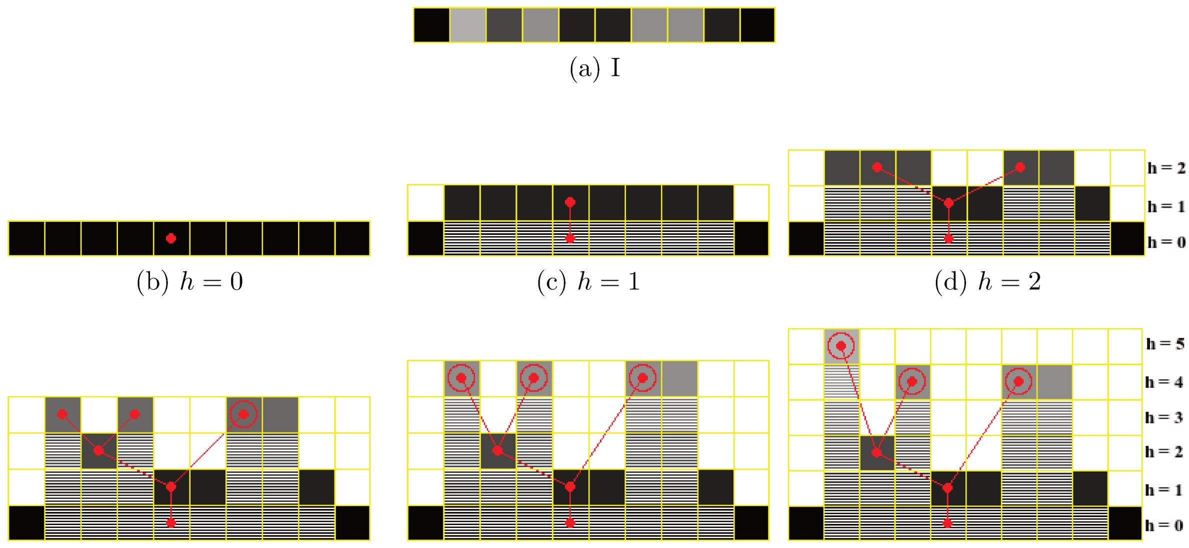
where $D(i)$ is the set containing all the descendants of i. Node i is a parent of node j if C_i is the smallest connected component that contains C_j.

The number of connected components *nlevels* that a Max-tree node i represents is given by:

$$nlevels(i) = h_i - h_{p_i}. \tag{3.9}$$

If $nlevels(i) > 1$, this node is a **composite node**, i.e. a node that represents a connected component that remained unchanged for a sequence of threshold values. This notion of composite node is useful to implement the Max-Tree based algorithms.

The restitution of the image corresponding to a given Max-Tree, is an inverse mapping of the pixels stored in the nodes to the image coordinates system, and is expressed by the



(a) I

(b) $h = 0$ (c) $h = 1$ (d) $h = 2$

(e) $h = 3$ (f) $h = 4$ (g) $h = 5$

Figure 3.3: Max-Tree construction of the 1D image $I = [0, 5, 2, 4, 1, 1, 4, 4, 1, 0]$ (a). Bottom-up construction (b)-(g). Resulting Max-Tree (g).

following equation:

$$I(z) = \{h_i | z \in \widehat{C_i}\}. \tag{3.10}$$

The construction of the Max-Tree corresponding to the 1D image $I = [0, 5, 2, 4, 1, 1, 4, 4, 1, 0]$ is illustrated in Figure 3.3. The pixels stored in each node is obtained by the set difference between the pixels obtained at level h minus the pixels at level $h + 1$, and the empty nodes are removed from the final representation. The corresponding leaves of the Max-Tree and the Component Tree store the same set of pixels. This illustration is meant just to explain the Max-Tree structure and construction, it does not correspond to the most efficient form to build it, for that, there are algorithms that run in quasi-linear time (Salembier et al., 1998; Carlinet and Géraud, 2012). The composite nodes are represented by double circles. The filled pixels in each connected component represent the pixels each Max-Tree node stores ($\widehat{C_i}$), and its union with the dashed pixels in the same connected component (C_i) is the result of applying Equation 3.8 to this node.

3.2.2 Max-Tree Filtering

Filtering the Max-Tree consists on contracting its nodes. The most usual rule used to perform this contraction is the direct rule (Salembier et al., 1998). There are two possible cases: a full node contraction and a composite node partial contraction. The full contraction of a sequence of nodes that constitute a sub-tree rooted in a given node is called pruning. These operations are explained in the next subsections.

Node Contraction by Direct Rule

In the full node contraction by direct rule, the node selected contracts with its parent, and its pixels are also merged with it. The children of the node being contracted become children of its parent. The full contraction by direct rule procedure of a node i with its parent p_i is given by:

$$p_j = p_i, \quad \forall j \in children(i), \tag{3.11}$$

$$\widehat{C_{p_i}} = \widehat{C_{p_i}} \cup \widehat{C_i}, \tag{3.12}$$

$$\mathcal{N} = \mathcal{N} \setminus i, \tag{3.13}$$

where $children(i)$ is the set containing all the children of node i. The full node contraction decreases the number of Max-Tree nodes by 1.

In the partial node contraction by direct rule, the composite node is contracted by decreasing its $nlevels$ attribute. The composite node partial contraction by direct rule of a node i by x is given by:

$$h_i = h_i - x, \quad 1 \leq x \leq nlevels(i) - 1. \tag{3.14}$$

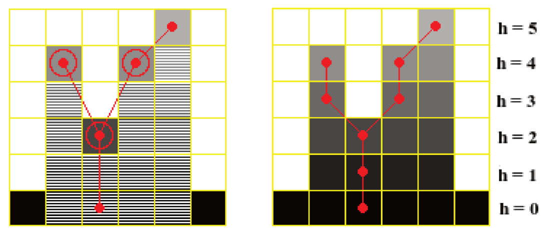
The parameter x defines how many levels the composite node will contract. The partial node contraction does not alter the number of Max-Tree nodes.

Note that by contracting a node j in the Max-tree, the attribute $nlevels$ of its children is being increased. In the full contraction, it is increased by $nlevels(j)$, and in the partial contraction it is increased by x.

It is important to mention that contracting a node of the Max-Tree is a connected filter, and after choosing which nodes will be fully contracted, partially contracted and the parameter x, the order of the contractions is irrelevant, the result will always be the same.

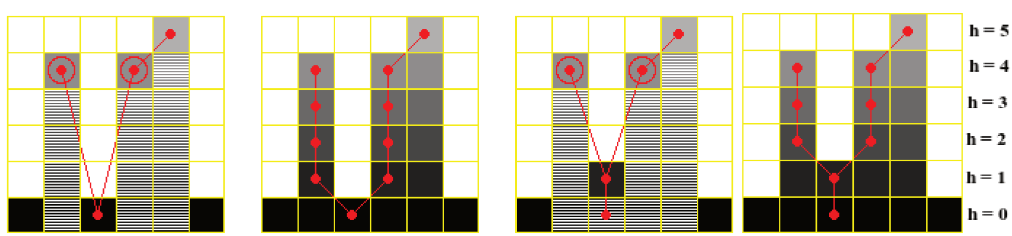
The Max-Tree node being contracted may be a leaf, a ramification, or a node with one child, and depending on the node it has a different effect on the equivalent Component Tree. These three cases are analyzed.

The Max-Tree of the image $I = [0, 4, 2, 4, 5, 0]$, and its corresponding Component Tree are illustrated in Figure 3.4. The result of a full contraction and a partial contraction with $x = 1$ of the ramification node in Figure 3.4(a), which is a composite node with $nlevels =$ 2, and its meaning on the Component Tree is illustrated in Figure 3.5. The full contraction operation reduced the number of Max-Tree nodes by 1, and increased the number of nodes in the Component Tree from 8 to 10. The full contraction of the ramification node in the Max-Tree is equivalent to replacing the Component Tree connected components at levels $h = 1$ and $h = 2$, by the two connected components of their descendants at level $h = 3$. The partial contraction did not change the number of Max-Tree nodes as expected. The effect on the Component Tree was replacing the connected component at level $h = 2$ by the two connected components of their descendants at level $h = 3$, therefore the number of Component Tree nodes increased from 8 to 9. The conclusion is that contracting Max-Tree ramification nodes increases the number of nodes of its equivalent Component Tree.



(a) Max-Tree (b) Component Tree

Figure 3.4: Max-Tree (a) and Component Tree (b) corresponding to the image $I = [0, 4, 2, 4, 5, 0]$.



(a) (b) (c) (d)

Figure 3.5: Max-Tree (a) and Component Tree (b) after full contraction, and Max-Tree (c) and Component Tree (d) after partial contraction by direct rule with $x = 1$ of the ramification node in Figure 3.4(a).

The result of a full contraction and a partial contraction with $x = 1$ of the leaf on the top left of the Max-Tree in Figure 3.4(a), which is a composite node with $nlevels = 2$, and its meaning on the Component Tree is illustrated in Figure 3.6. The full contraction operation reduced the number of Max-Tree nodes by 1, and decreased the number of nodes in the Component Tree by 2. The partial contraction did not change the number of Max-Tree nodes, and reduced the number of Component Tree nodes by 1. The conclusion is that contracting Max-Tree leaves reduces the number of nodes of its equivalent Component Tree.

The result of a full contraction and a partial contraction with $x = 1$ of the node with one child on the top right of the Max-Tree in Figure 3.4(a), which is a composite

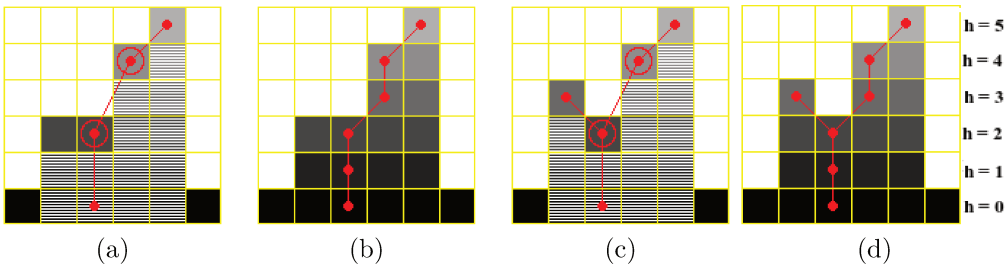


Figure 3.6: Max-Tree (a) and Component Tree (b) after full contraction, and Max-Tree (c) and Component Tree (d) after partial contraction by direct rule with $x = 1$ of the leaf on the top left of the Max-Tree in Figure 3.4(a).

node with $nlevels = 2$, and its meaning on the Component Tree is illustrated in Figure 3.7. The full contraction operation reduced the number of Max-Tree nodes by 1, and did not alter the number of nodes in the Component Tree. The partial contraction did not change neither the number of Max-Tree nodes, neither the number of Component Tree nodes. The conclusion is that contracting Max-Tree nodes with one child does not alter the number of nodes of its equivalent Component Tree.

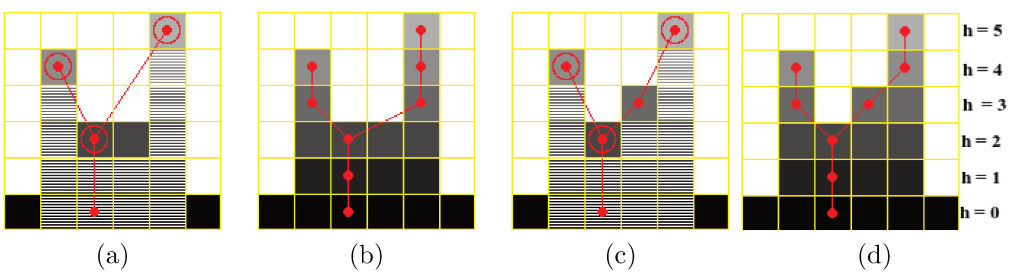


Figure 3.7: Max-Tree (a) and Component Tree (b) after full contraction, and Max-Tree (c) and Component Tree (d) after partial contraction by direct rule with $x = 1$ of the node with one child on the top right of the Max-Tree in Figure 3.4(a).

Pruning

The pruning consists on performing full contractions of the nodes of a sub-tree. It is useful to implement attribute filters with increasing criteria, such as the area opening (Vincent, 1994). For instance, the Max-Tree, and its corresponding Component Tree of the image $I = [0, 2, 5, 3, 4, 5, 2, 0]$ are illustrated in Figure 3.8. If we decide to apply an area opening filter that removes all the components with area less than or equal to 4, it is necessary to prune the sub-tree rooted in the ramification node of the Max-Tree. The result of the pruning is also illustrated in Figure 3.8. Note that the number of Max-Tree nodes reduced from 6 to 2, and the number of Component Tree nodes reduced from 8 to 3, therefore the pruning reduces the number of nodes in both the Component Tree and the Max-Tree. The Max-Tree filtering operations and their effect on the Component Tree are summarized in Table 3.1.

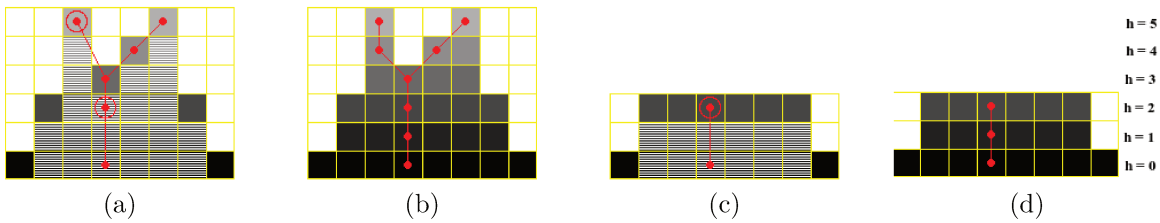


Figure 3.8: Max-Tree (a) and Component Tree (b) corresponding to the image $I = [0, 2, 5, 3, 4, 5, 2, 0]$. Max-Tree (c) and its corresponding Component Tree (d) after the pruning of the sub-tree rooted in the ramification node.

Max-Tree Operation	**Effect on the Component Tree**
Contraction of a leaf	reduces the number of nodes
Contraction of a ramification	increases the number of nodes
Contraction of a node with one child	does not alter the number of nodes
Pruning	reduces the number of nodes

Table 3.1: Summary of Max-Tree filtering operations and their effects on the Component Tree.

3.3 Sub-branch

Definition 3.1 *A sub-branch SB_j of a rooted tree $\mathcal{T}$ is defined as a set of nodes that joins a ramification or a leaf until the nearest ramification child or root child. The root is defined as being a sub-branch with a single node. Sub-branches that have only one node are called trivial sub-branches.*

According to this definition, each node of the rooted tree belongs to one and only one sub-branch, therefore the set composed of all sub-branches of a rooted tree is disjoint and contains all its nodes.

$$SB_j \cap SB_k = \emptyset, \quad if \quad j \neq k, \tag{3.15}$$

$$\mathcal{N}(\mathcal{T}) = SB_0 \cup SB_1 ... \cup SB_m. \tag{3.16}$$

A rooted tree and its sub-branches are illustrated in Figure 3.9. We can think of each sub-branch as being a single node, and therefore we can build a sub-branch tree, as illustrated in Figure 3.10. The sub-branch tree is the minimal tree that preserves the same ramifications with the same degrees and ordering as the ramifications of the original tree.

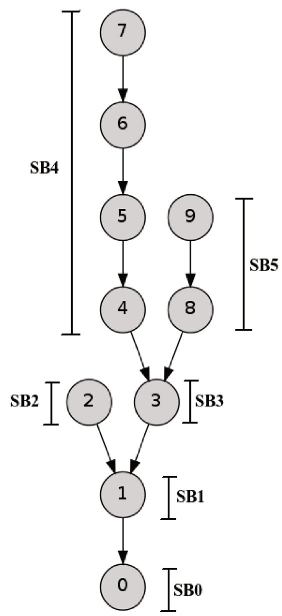


Figure 3.9: Illustration of sub-branches.

In the Max-Tree, each set of nodes SB_j that constitutes a sub-branch represents the variation of a connected component for different threshold values, before it splits in two or more components or before it completely disappears. The number of threshold levels before the connected component represented by the sub-branch splits or completely disappears will be denoted by *ntlevels*, and is given by the following equation:

$$ntlevels(SB_j) = \sum_{\forall k \in SB_j} nlevels(k). \tag{3.17}$$

The number of nodes in the sub-branch tree is equal to the number of sub-branches in the original tree and is given by the number of leaves n plus the number of ramifications

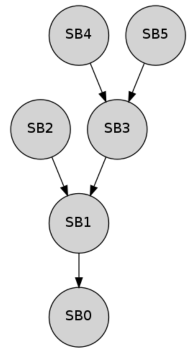
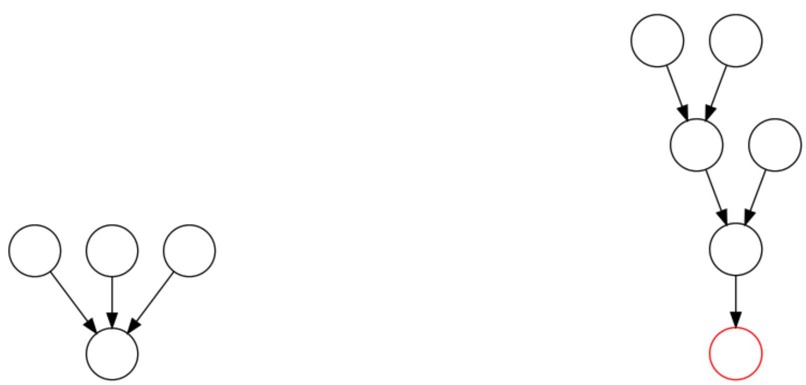


Figure 3.10: Illustration of the sub-branch tree corresponding to the tree in Figure 3.9.

plus one, if the root is not a ramification. Its minimum value is equal $n+1$. It happens when all leaves are connected directly to the root. Its maximum value happens when the sub-branch tree is almost a full binary tree, except for the root that has only one child. From the theory, the number of nodes in a full binary tree is $2n-1$, since in the sub-branch tree, the root may have only one child, the maximum number of nodes of the sub-branch tree is $2n$. Therefore,

$$n+1 \leq \text{Number of sub-branch tree nodes} \leq 2n. \tag{3.18}$$

The case where the number of nodes in the sub-branch tree is equal to the lower bound $n+1$ is illustrated in Figure 3.11(a), and the case where the number of nodes in the tree is equal to the upper bound $2n$ is illustrated in Figure 3.11(b) .



(a) $n+1$

(b) $2n$

Figure 3.11: Illustration of the cases where the number of nodes in the sub-branch tree are equal to $n+1$ (a) and $2n$ (b).

The concept of sub-branch will be useful to define the MMS filter in Chapter 5.

3.4 Conclusions

In this chapter, the Component Tree and the Max-Tree structures were reviewed. The composite node and the sub-branch concepts were introduced, along with their interpretation on the Max-Tree. The filtering procedure on the Max-Tree and its meaning on the Component Tree were analyzed. The main differences between the Component Tree and the Max-Tree are summarized below:

- The Max-Tree represents a connected component that remained unchanged for a sequence of threshold values in a single node called composite node, therefore the number of Max-Tree nodes is always less than or equal to the number of Component Tree nodes. The equality happens when the Max-Tree does not have any composite nodes.

- The number of Component Tree nodes is equal to the sum of the attribute *nlevels* of each Max-Tree node;

- The Max-Tree nodes store $\widehat{C_i}$ while the Component Tree nodes store C_i;

- After filtering the Max-Tree, the number of nodes is always less than or equal to the number of nodes of the original Max-Tree, while the filtering procedure of the Component Tree may increase, decrease or not alter the number of Component Tree nodes, see Table 3.1 for a summary.

The main advantage of the Max-Tree in relation to the Component Tree is its compactness, which is one of the goals of the MMS filter that will be defined in Chapter 5. For that reason, from now on, all operations will be defined in terms of the Max-Tree, even though sometimes we will go back to the Component Tree attribute signature for aid in some decision making steps.

Chapter 4

Extinction Filter

In this chapter, it is defined the Extinction Filter (EF), which is a connected filter that preserves the relevant extrema of the image. Extinction filters are closely related to dynamic openings (Salembier and Wilkinson, 2009), and unlike the usual contrast and size filters, the heights of the remaining maxima in the image are completely preserved. It is shown that applying the EF reduces the number of Max-Tree nodes and sub-branches, and that the average length of the remaining sub-branches in the tree increases. The relationship between Extinction filters and the *hmax* (Salembier and Oliveras, 1996), *vmax* (Vachier, 1995b) and *area opening* (Vincent, 1994) attribute filters is analyzed.

This chapter starts by presenting an algorithm to compute height, area, and volume extinction values from the Max-Tree. This algorithm can be easily extended to compute extinction values of any crescent attribute. Then, the EF is defined and analysed. After that, the EF is compared to the *hmax*, *vmax* and *area opening* attribute filters. Finally, the conclusions about this chapter are presented.

4.1 Computing Extinction Values from the Max-Tree

Extinction values are a measure of persistence from the attribute being analyzed proposed by Vachier (1995a). They are a powerful tool to measure the persistence of a crescent attribute, and are useful to discern relevant from irrelevant extrema, usually noise. The most usual extinction attributes are height, area, and volume. They can be efficiently computed from the Max-Tree structure. The height, area, and volume corresponding to the level component C_i of node i are given, respectively, by the following equations:

$$\mu_h(C_i) = \max_{\forall k \in descendants(i)} \{h_k - h_i\}, \tag{4.1}$$

$$\mu_a(C_i) = \sum_{\forall z \in C_i} 1, \tag{4.2}$$

$$\mu_v(C_i) = \mu_a(C_i) + \sum_{k \in descendants(i)} nlevels(k)\mu_a(C_k), \tag{4.3}$$

where $descendants(i)$ is a set containing all the descendants of node i.

An algorithm to compute height, area, and volume extinction values from the Max-Tree is illustrated in the Code Fragment 4.1. The algorithm is similar to the one presented in Silva and Lotufo (2008), but with a modification (lines 6 to 8) to account for the composite nodes. The algorithm is described using a Python-NumPy syntax, which has high level commands similar to a pseudo-code, and are easy to understand. The function receives 5 arrays and a string. The string corresponds to the attribute being analyzed "height", "area", or "volume". $nlevels$, *area*, *parents*, *nchild*, *attr* are arrays storing the $nlevels$, the area, the parent ID, the number of children, and the attribute being analyzed, respectively. They are indexed by the nodes IDs. Line 4 generates a boolean mask (*cn*) set with true for the composite nodes. Lines 6 through 8 update the attribute depending of the option received in the string. This has to be done since for the height and volume attributes of composite nodes, we need to analyze the attribute value, which corresponds to the level component hidden in the lower connected component of the composite node. Lines 10 through 16 store for each node the highest attribute value among their children in the array $achmax$, and the corresponding node ID in the array $ichmax$.

Lines 18 to 29 compute the extinction values of the regional maxima, i.e. the leaves of the tree. Initially, all extinction values are set as 0. Then, a loop is performed. For each tree leaf, a path towards the root is initiated. When a node with more than one child appears in the path, a verification is done on each children of this node. If the node in the leaf path being analysed has a sibling with higher attribute, then the attribute of this node in the leaf path is defined as the extinction value of the leaf being analyzed and another iteration of the loop is performed, else we continue the path until the next node with more than one child. Finally, if the path reaches the root, then its extinction value is set as the attribute value of the first node in the path before arriving at the root,which is the maximum image extinction value. Line 30 returns an array with all the extinction values. Nodes that are not leaves have their extinction value set to 0

Code Fragment 4.1: Python-NumPy code to compute height, area, and volume extinction values from the Max-Tree.

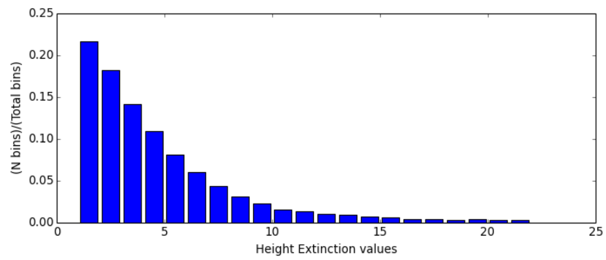
```python
def getExtinctionValues(nlevels,area,parents,nchild,attr, opt = "area"):
  N = len(area)
  root = 0
  cn = nlevels > 1

  if opt == "area":pass
  elif opt == "height": attr[cn] = attr[cn] + (nlevels[cn] - 1)
  elif opt == "volume": attr[cn] = attr[cn] + (nlevels[cn] - 1)*area[cn]

  ichmax = zeros(N)
  achmax = zeros(N)
  for node in arange(1,N)[::-1]:
    nparent = parents[node]
    if attr[node] > achmax[nparent]:
      achmax[nparent] = attr[node]
      ichmax[nparent] = node

  ext_values = zeros(N)
  for nd in nonzero(nchild == 0)[0]:
    node = nd
    while parents[node] != root:
      vv  = attr[node]
      big = ichmax[parents[node]]
      if node != big:
        ext_values[nd] = vv
        break
      node = parents[node]
    if parents[node] == root:
      ext_values[nd] = achmax[node]
  return ext_values
```

4.2 Extinction Filter

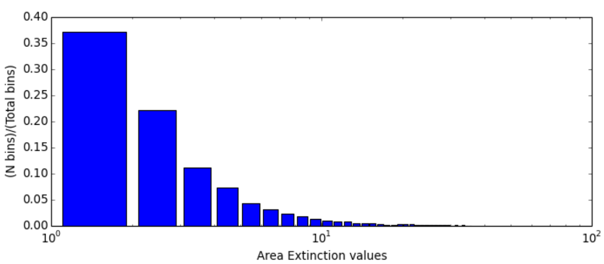
The Extinction Filter (EF) is a connected filter that preserves the relevant maxima of the image, while reducing the tree complexity, i.e. the number of tree nodes. Most natural images are contaminated by noise, therefore they probably contain many irrelevant extrema with low extinction values. For instance, the famous Lena image with dimensions 554×507 pixels is shown in Figure 4.1(a). Its corresponding Max-Tree using connectivity

$C8$ [1] has 37689 nodes of which 14780 are leaves, i.e. maxima. The height, area and volume normalized extinction histograms, respectively are shown in Figure 4.1(b)-(d). Only the lower extinction values are displayed for better visualization. It is clear that the histograms are highly concentrated on the lower extinction values, and probably most of these maxima in the image correspond to noise or irrelevant artifacts.

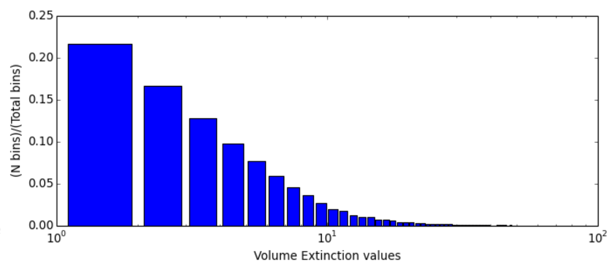
(a) Lena



(b) Height extinction histogram



(c) Area extinction histogram

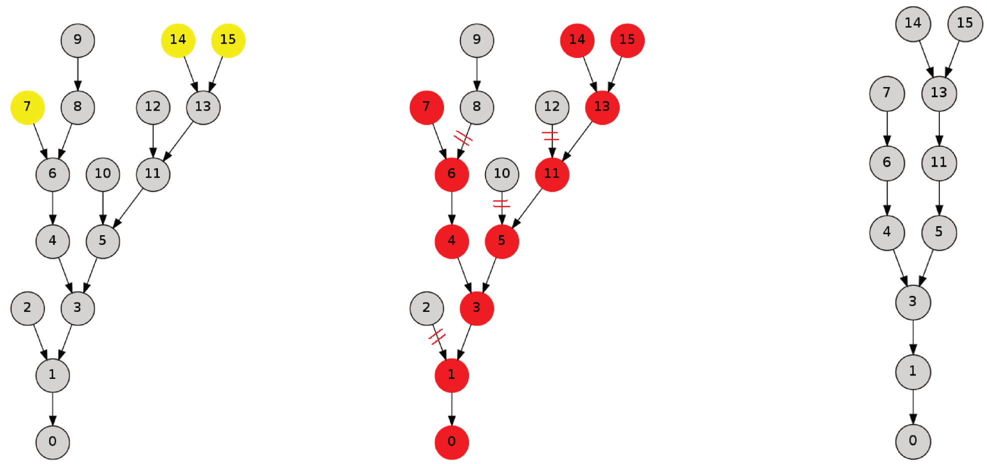


(d) Volume extinction histogram

Figure 4.1: Lena image (a), and its height (b), area (c), and volume (d) normalized extinction histograms.

The EF operation is very simple. The n leaves with highest extinction values concerning the crescent attribute being analyzed are chosen. The nodes in the paths from these leaves to the the root are marked as to be kept. All the other nodes are fully contracted. Since the contraction of Max-Tree nodes is a connected filter, the EF is also a connected filter. The EF procedure is illustrated in Figure 4.2. Suppose that $n = 3$ and nodes 7, 14 and 15 (the yellow nodes) of Figure 4.2(a) are the leaves with highest extinction value according to the attribute being analyzed. The nodes in the paths from these leaves to the the root are marked in red, Figure 4.2(b), the remaining nodes are pruned. The resulting tree is illustrated in Figure 4.2(c).

[1] From now on we assume that all Max-Trees are built considering a connectivity $C8$, unless explicitly said otherwise.



(a) Original Max-Tree (b) Nodes to be kept marked in red (c) Resulting Max-Tree

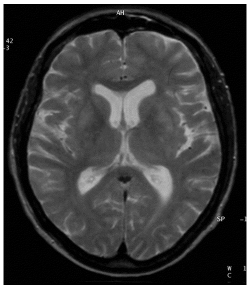
Figure 4.2: Original Max-Tree (a), the yellow nodes are the three nodes with highest extinction values. Nodes in the path from the three leaves with highest extinction values to the root are marked in red (b). The result of the pruning of the nodes not marked in red (c).

In order to illustrate, and analyze the EF, it was chosen 5 sample images with different characteristics: a brain MRI, a lung CT, a natural image with text in it, a cameraman picture, and an image with many objects in it. They are 8 bits images with their pixel intensities varying between 0 and 255, their information is summarized in Table 4.1, and the images are displayed in Figure 4.3.

Image Name	**Dimensions**	**Nb. of Max-Tree nodes**	**Nb. of Sub-branches**
MRI	1280×1105	197203	79784
CT	512×512	44829	27572
Cameraman	256×256	11123	5883
Objects	406×409	12627	4442
Text	375×500	25827	12699

Table 4.1: Summary of the information concerning the sample images.

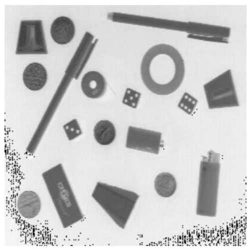
The results of applying the Area EF to the sample images are illustrated in Figure 4.4. The parameter n was set as being 1% of the number of maxima in the original images. The average reduction in the number of Max-Tree nodes after the filtering was around a factor of 10, and all filtered images have an SSIM index higher than 0.94 in relation to their corresponding original image. It is hard to visually notice differences between



(a) MRI

(b) CT

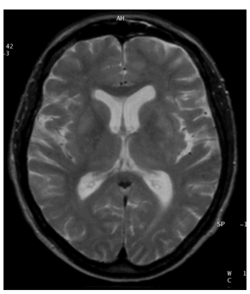
(c) Cameraman



(d) Objects

(e) Text

Figure 4.3: Sample images.



(a) 13287 nodes, $SSIM = 0.998$

(b) 3313 nodes, $SSIM = 0.998$

(c) 2218 nodes, $SSIM = 0.995$

(d) 2946 nodes, $SSIM = 0.943$

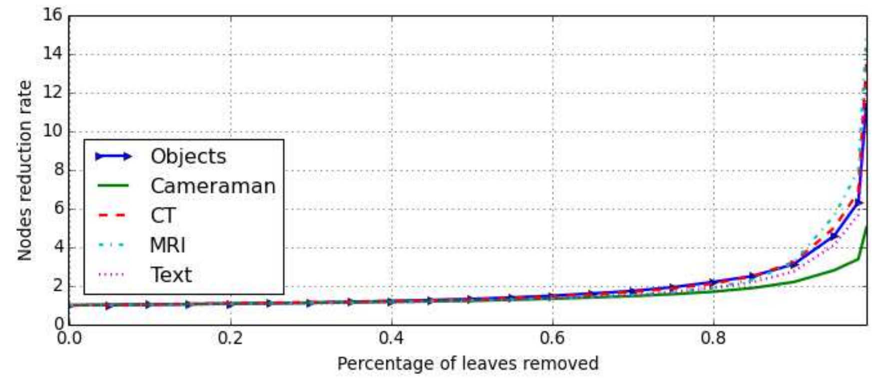
(e) 2067 nodes, $SSIM = 0.997$

Figure 4.4: Illustration of the Area EF applied to the sample images. The parameter n was equal to 1% of the number of maxima in the corresponding original image.

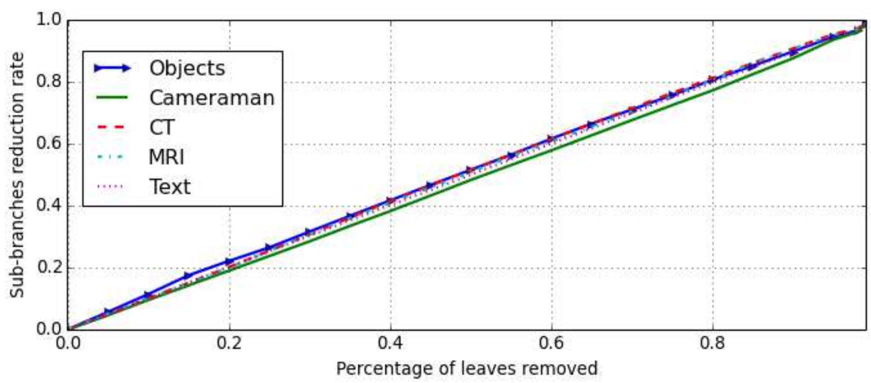
the original images and the filtered images, except for the Objects image, in which a considerable part of the black texture in the bottom of the original image disappeared in the filtered image. These results support the hypothesis that in general most of the extrema in natural images correspond to noise.

The variation of the number of nodes, sub-branches and the average sub-branch length according to the parameter n used in the Area EF are illustrated in Figure 4.5. The abscissas of the plots represent the percentage of maxima with lower extinction values that were chosen to be removed, i.e. $1 - \frac{n}{\text{nb. of leaves}}$. The ordinate of Figure 4.5(a) represents the ratio between the number of nodes of the original Max-Tree and the number of nodes after the EF. The ordinate of Figure 4.5(b) represents one minus the ratio between the number of sub-branches in the filtered Max-Tree and the number of sub-branches of the original Max-Tree. Finally, the ordinate of Figure 4.5(c) represents the average sub-branch length.

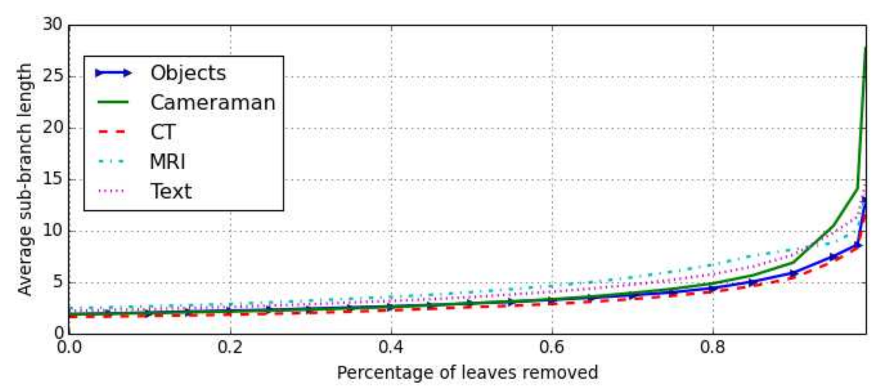
The nodes reduction rate in Figure 4.5(a) is smaller than 2 until the percentage of leaves removed is equal 0.7, probably because most leaves correspond to irrelevant maxima, and they are contained in sub-branches of small length. The sub-branches reduction rate has practically a linear behavior for all sample images. Also, as expected the average sub-branch length becomes greater with the increase in the percentage of maxima being filtered.



(a) Nodes reduction rate



(b) Sub-branches reduction rate



(c) Average sub-branch length

Figure 4.5: Illustration of the variation of the number of nodes (a), sub-branches (b) and the average sub-branch length (c) according to the parameter n used in the Area EF.

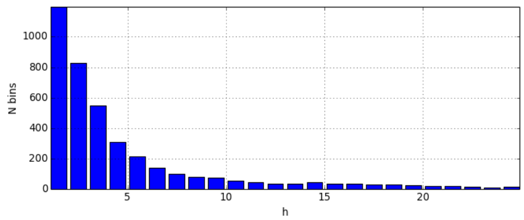
4.3 Relationship between Extinction Filters and Attribute Filters

Attribute filters, such as the *hmax*, *vmax*, and the *area opening* may also be used to set the number of extrema to be preserved in an image. The procedure is simple. First, the extinction values of the crescent attribute being analyzed have to be calculated. Then, the extinction histogram is computed. The relationship between the number of extrema in the image and the parameter of the attribute filter is given by the curve attribute value versus the number of maxima minus the cumulative distribution of the extinction histogram. This curve may have discontinuities, since there may be leaves with the same extinction value, therefore when using attribute filters often it is not possible to set the exact number of extrema to be preserved. This case is illustrated in Figure 4.6. The height extinction histogram of the Cameraman sample image is shown in Figure 4.6(a). The curve height versus number of extrema is shown in Figure 4.6(b). The zoom in the plot highlights the discontinuities in this curve. For instance, if the image is filtered with the *hmax* filter with $h = 50$ the number of maxima in the resulting image will be 101, but if it is filtered with $h = 51$ the number of maxima in the resulting image will be 99, it is not possible to obtain exactly 100 maxima using the *hmax* filter, while with the Height EF it is possible. The EF can use a second criteria to choose between the maxima that have a extinction tie. The results of the Height EF with $n = 100$ and the *hmax* filter with $h = 50$ are illustrated in Figure 4.6(c) and Figure 4.6(d), respectively. In order to solve the extinction tie problem, the maxima to be preserved were randomly chosen between the maxima with the same height extinction values.

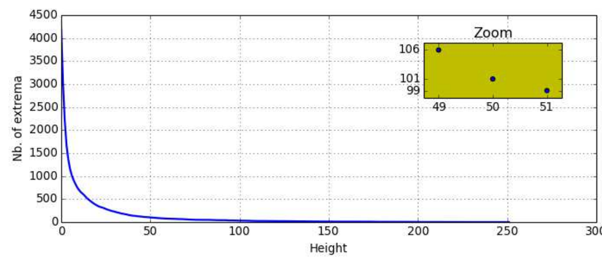
The fact that the EF can set exactly the number of maxima to be preserved in the image may be an advantage, since in many segmentation problems we are looking for a specific number of regions in the image, and many segmentation techniques use the maxima in the image as seeds to the segmentation process.

The result of applying the Volume EF and the *vmax* filter are illustrated in Figure 4.7. The parameter n was set as being 0.25% of the number of maxima in the original images, and the parameter v was set in order to preserve the same number of maxima or at least the closest value possible. The images obtained using the Volume EF obtained a SSIM index slightly higher than the SSIM index obtained using the *vmax* filter, but this difference is practically negligible. The *vmax* filter obtained better simplification results in terms of flat zones, and number of Max-Tree nodes than the Volume EF, these results are summarized in Table 4.2.

Attribute filters usually achieve better simplification results, this happens due the fact that this kind of filter only preserves the extrema, but not their original height. The EF preserves perfectly the heights of the extrema chosen to be kept. When set to preserve the



(a)



(b)

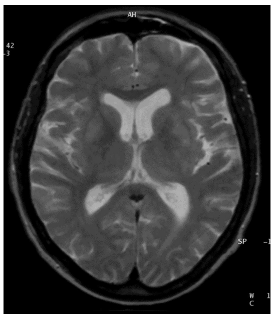
(c)

(d)

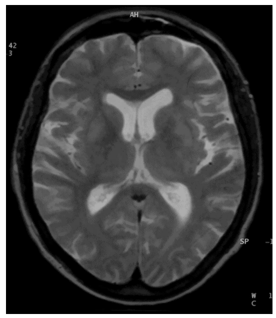
Figure 4.6: Comparison of the height EF and the *hmax* filter. The number of maxima in image (c) is 100, while the number of maxima in image (d) is 101. This may happen when there are ties between extrema extinction values.

	Original Image		**Volume EF**		**vmax**	
	Nodes	**Flat zones**	**Nodes**	**Flat zones**	**Nodes**	**Flat zones**
MRI	197203	761080	7328	471797	3688	449234
CT	44829	169318	1900	108433	990	106092
Cameraman	11123	43268	1118	26010	527	22184
Objects	8398	55834	563	43412	371	41913
Text	28132	133907	988	83490	391	77248

Table 4.2: Summary of the results obtained by the Volume EF and the *vmax* filter.

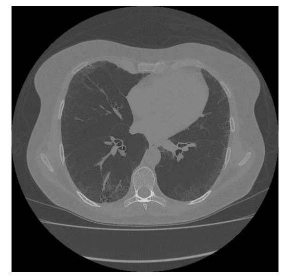


(a) SSIM = 0.998



(b) SSIM = 0.997

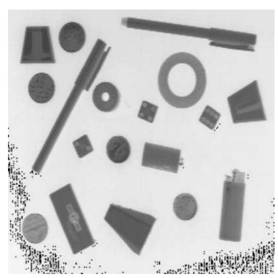
(c) SSIM = 0.997



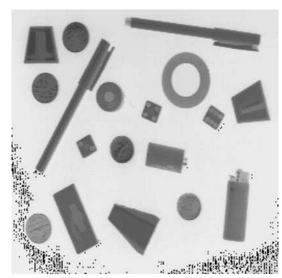
(d) SSIM = 0.996

(e) SSIM = 0.984

(f) SSIM = 0.974



(g) SSIM = 0.998



(h) SSIM = 0.996

(i) SSIM = 0.983

(j) SSIM = 0.973

Figure 4.7: Comparison of the results obtained by the volume EF and the *vmax* filter.

same number of extrema, the resulting Max-Trees of the EF filter and the attribute filter have the same number of sub-branches, except possibly in cases where there are extinction ties. Their main difference is that the sub-branches that contain the leaves preserved by the EF are usually longer than the sub-branches that contain the leaves preserved by the attribute filter.

4.4 Conclusions

In this chapter, Extinction filters were defined in terms of the Max-Tree. It was seen that the smaller the parameter n of the EF, the larger is the nodes reduction rate, and the average sub-branch length increases. Also, the illustrations indicated an almost linear relationship between the parameter n and the reduction rate in the number of sub-branches. The relationship between Extinction filters and attribute filters was analyzed, and the main differences are:

- Extinction filters perfectly preserve the height of the extrema to be kept, while attribute filters erode the height of the extrema. This means that attribute filters usually have a greater simplification power. One or other property may be more desirable depending of the problem.

- It is not always possible to set an exact number of extrema in the image when using attribute filters, this happens due to extinction ties. This problem does not occur with Extinction filters, since they can use a second criterion to decide among the extrema with extinction ties.

Although attribute filters usually achieve greater Max-Tree simplification results than Extinction filters, we will see in Chapter 5 that this brings no advantage for the methodology we propose, since the number of sub-branches in the tree obtained by the attribute filter and the EF are the same, and the methodology proposed preserves only one Max-Tree node per sub-branch, therefore the number of nodes after the MMS filter will be the same either using attribute filters or the EF. For that reason, we argue that the EF filter is more adequate, to our methodology, because it perfectly preserves the height of the extrema kept in the image.

Chapter 5

Maximal Max-Tree Simplification

This chapter presents the Maximal Max-Tree Simplification (MMS) filter, which is a filter that maximally reduces the number of Max-Tree nodes, while preserving the sub-branches structure of the tree. Two possible criteria to compute the MMS filter are proposed. The first is a normalized threshold criterion (MMS-T) and the second is a Maximally Stable Extremal Regions (MSER) (Matas et al., 2002, 2004) criterion (MMS-MSER). The filter is analyzed along with its sensible cases, and a methodology to apply it in association with Extinction filters is proposed.

5.1 MMS Filter

Definition 5.1 *The Maximal Max-Tree Simplification (MMS) is a filter that selects one node in each sub-branch of the Max-Tree, and fully contracts all other nodes, i.e. all sub-branches become trivial, therefore the resulting tree is the minimal tree that preserves the same ramifications with the same degrees and ordering as the sub-branches tree.*

The MMS filter is a connected filter, since its operation consists in contracting Max-Tree nodes. It greatly reduces the number of Max-Tree nodes. Since every sub-branch is now a trivial sub-branch, the number of Max-Tree nodes is bounded between $n + 1$ and $2n$, where n is the number of Max-Tree leaves, as explained in Section 3.3.

Notice that the MMS filter removes intermediary nodes, i.e., it is not a pruning, which many authors consider an operation that is not robust (Salembier et al., 1998). We believe that with the right criterion it provides robust results. We propose two criteria to choose the sub-branch node to be preserved, the first is a normalized threshold criterion and will be called MMS-T and the second uses the MSER stability criterion and will be called MMS-MSER.

5.1.1 MMS-T

The MMS-T uses a normalized threshold criterion to select the node to be preserved. A threshold value t between 0 and 1 is chosen, and this threshold value is mapped into a node in the sub-branch. The procedure is the following: suppose we are analyzing SB_j, the grey-level h of the node to be preserved in this sub-branch is given by the following equation:

$$h = h_{sn} - nlevels(sn) + 1 + t[ntlevels(SB_j) - 1], \tag{5.1}$$

where h_{sn} is the grey-level of the sub-branch node closer to the root. The node to be preserved is:

$$i = \{i \in SB_j | h_i - nlevels(i) < h \leq h_i\}. \tag{5.2}$$

After choosing the node to be preserved in each sub-branch, all other nodes are fully contracted. For each node i to be preserved, if $h_i - h \geq 1$, the node is partially contracted by $x = h_i - h$. Setting $t = 0$ corresponds to choose the sub-branch node closest to the root, which is the connected component with highest area in the sub-branch. Setting $t = 1$ corresponds to choose the sub-branch node further away from the root, which is the connected component with smallest area in the sub-branch.

5.1.2 MMS-MSER

The MMS-MSER chooses to preserve the node in the sub-branch with the highest stability measure given by a slightly modified formulation of the MSER stability criterion given by:

$$\Psi^{\Delta}(i) = \frac{A(C_i) - A(C_j)}{A(C_j)}, \tag{5.3}$$

where $A()$ represents the area of the connected component in the argument, and j is the first ancestor of i, where $h_j \leq h_i - \Delta$. The region C_i represented by node i is said a MSER region if $\Psi^{\Delta}(i)$ is a local maximum.

The stability measures of the connected components hidden in the composite nodes do not have to be considered, since the connected component in the highest level of the composite node will always have the highest stability measure according to this slightly modified MSER formulation. This happens because its area remains constant for $nlevels$, and the components hidden below it will vary sooner.

The node k to be preserved in the sub-branch SB_j using the MMS-MSER is given by the following equation:

$$k = \underset{\forall i \in SB_j}{argmax}\{\Psi^{\Delta}(i)\}. \tag{5.4}$$

After choosing the node to be preserved in each sub-branch, all other nodes are fully contracted.

5.2 Analysis of the MMS Filter

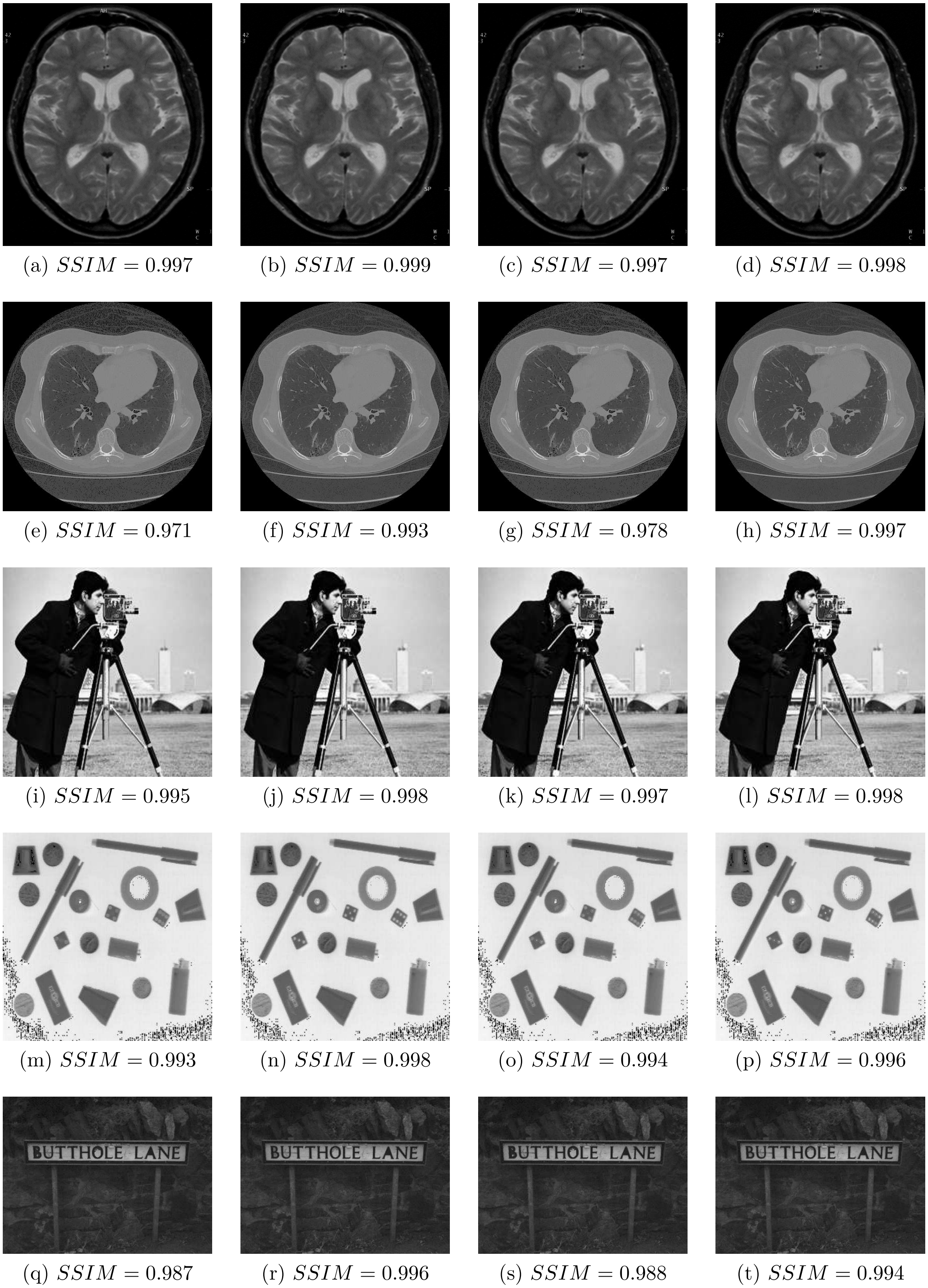
It was seen in Section 4.2 that the average sub-branch length of a natural image is around 2, therefore applying the MMS filter with no pre-processing will reduce the number of Max-Tree nodes on average by a factor of 2. The direct application of the MMS-T filter with $t = 0$, $t = 0.5$, $t = 1$, and the MMS-MSER filter to the sample images presented in Figure 4.3 are illustrated in Figure 5.1. Since, the average sub-branch length of the original images is around 2, basically the MMS filter is choosing one among two nodes to preserve in the Max-Tree, and regardless of the criterion used the filtering results are expected to be similar. The smallest SSIM index obtained was of 0.971 at the CT image with the MMS-T filter and $t = 0$. We will see that the methodology proposed in Section 5.3 simplifies much more the Max-Tree structure, while trying to preserve the necessary information to solve image analysis and recognition problems.

A sensible case of the MMS filter occurs when the sub-branches in the image are long and represent a great variation in size and/or shape of a connected component. When this happens, the filter may lose important image information. For instance, a synthetic image that contains a rectangle inside a circle, which is inside a triangle, each shape darker than the object that it contains, is illustrated in Figure 5.2(a). These shapes constitute a sub-branch (the only sub-branch) of this image. The results of the direct application of the MMS-T filter, with $t = 0.5$, 0.75, and 1 are illustrated in Figure 5.2(b)-(d) , respectively. In each filtered image only one of the shapes in the original image is preserved, if more than one shape was important to the problem the method would fail, fortunately this case does not occur in many problems.

5.3 Methodology

The filtering methodology proposed consists of applying the EF, the attribute used by the EF vary according to the problem, followed by the MMS filter. The EF sets the number n of leaves in the Max-Tree, and the MMS filter transforms each sub-branch in a trivial sub-branch, therefore the number of nodes in the resulting tree will be bounded between $n + 1$ and $2n$.

The criteria to be used depends on the problem. If we are interested in detecting distinguished regions, the MSER criterion is best, but if we are just looking to simplify the image, the normalized threshold criterion with $t = 0.5$ usually yields good results and it is faster than the MSER criterion.



(a) $SSIM = 0.997$ (b) $SSIM = 0.999$ (c) $SSIM = 0.997$ (d) $SSIM = 0.998$

(e) $SSIM = 0.971$ (f) $SSIM = 0.993$ (g) $SSIM = 0.978$ (h) $SSIM = 0.997$

(i) $SSIM = 0.995$ (j) $SSIM = 0.998$ (k) $SSIM = 0.997$ (l) $SSIM = 0.998$

(m) $SSIM = 0.993$ (n) $SSIM = 0.998$ (o) $SSIM = 0.994$ (p) $SSIM = 0.996$

(q) $SSIM = 0.987$ (r) $SSIM = 0.996$ (s) $SSIM = 0.988$ (t) $SSIM = 0.994$

Figure 5.1: Illustration of the direct application of the MMS-T filter with $t = 0$, (first column), $t = 0.5$ (second column), $t = 1$ (third column), and the MMS-MSER filter (fourth column).

(a) Original Image (b) $t = 0.5$ (c) $t = 0.75$ (d) $t = 1$

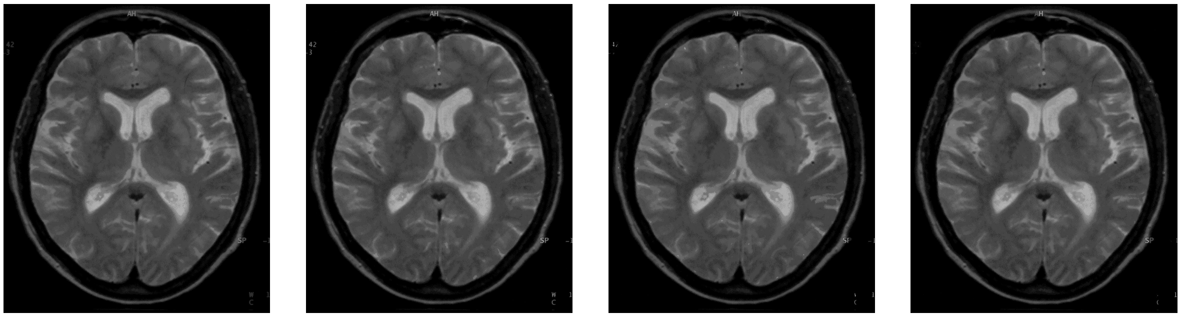
Figure 5.2: Shapes image (a), and the result of applying the MMS-T filter with $t = 0.5$ (b), $t = 0.75$ (c), and $t = 1$ (d).

The image resulting of the methodology proposed loses information when compared to the original image, but we believe that with the right criterion, the information necessary to solve the problem at hand is preserved. The application of the methodology proposed to the set of sample images is illustrated in Figure 5.3. The parameter n was set as 1% of the total number of leaves of the corresponding Max-Trees. It is clear that the filtered images suffer a considerable amount of degradation, but most of the information is still there and the reduction in the number of Max-Tree nodes was greater than 100 in all cases, as summarized in Table 5.1.

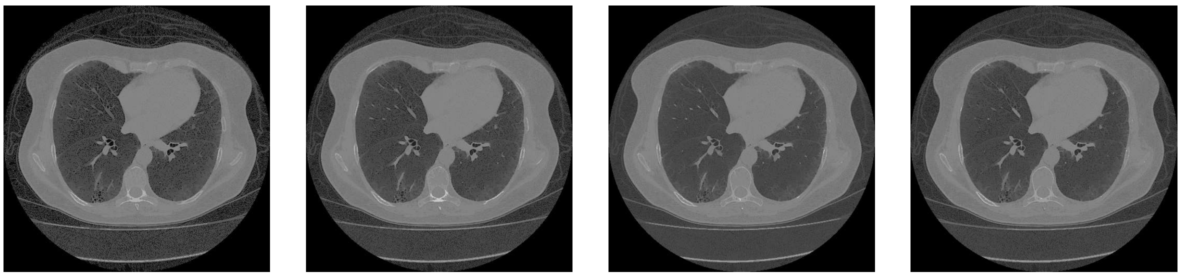
Image Name	**Nodes Reduction rate**
MRI	201
CT	160
Cameraman	139
Objects	233
Text	164

Table 5.1: Summary of the nodes reduction rate after applying the EF followed by the MMS filter to the sample images.

A small portion of a license plate image with dimensions of 30×50 pixels is depicted in Figure 5.4(a). The image is composed of three dark characters on a bright background, therefore the expexted Max-Tree of the negative of the image would have only four nodes: the root that would correspond to the background and three leaves corresponding to the three characters, but this does not occur in practice. The Max-Tree of the negative of the image has in practice 322 nodes, of which 54 are leaves. The Max-Tree filtered with the Volume EF with $n = 3$ has 174 nodes. The filtered image is shown in Figure 5.4(b). The posterior processing of the Max-Tree using the MMS-T filter with $t = 0.5$ has only 6 nodes, and its corresponding image is illustrated in 5.4(c). The Max-Tree graph before and after the filtering are presented in 5.4(d)-(e). This illustration shows the potential of

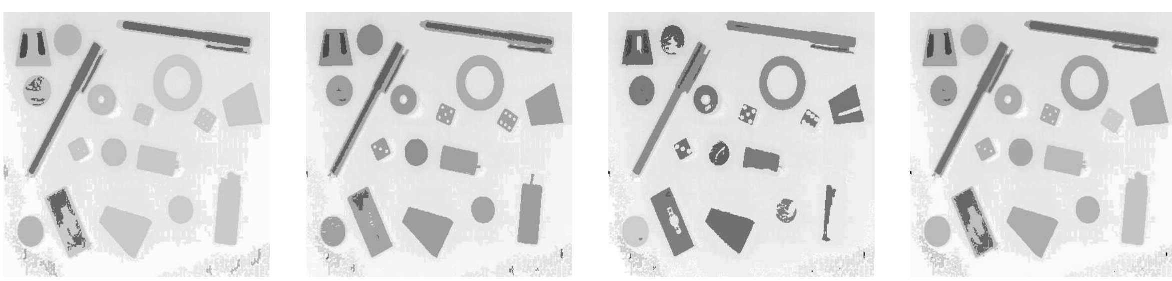


(a) $SSIM = 0.992$ (b) $SSIM = 0.994$ (c) $SSIM = 0.990$ (d) $SSIM = 0.991$



(e) $SSIM = 0.944$ (f) $SSIM = 0.976$ (g) $SSIM = 0.990$ (h) $SSIM = 0.984$

(i) $SSIM = 0.967$ (j) $SSIM = 0.979$ (k) $SSIM = 0.978$ (l) $SSIM = 0.972$



(m) $SSIM = 0.665$ (n) $SSIM = 0.823$ (o) $SSIM = 0.808$ (p) $SSIM = 0.775$

(q) $SSIM = 0.864$ (r) $SSIM = 0.939$ (s) $SSIM = 0.911$ (t) $SSIM = 0.836$

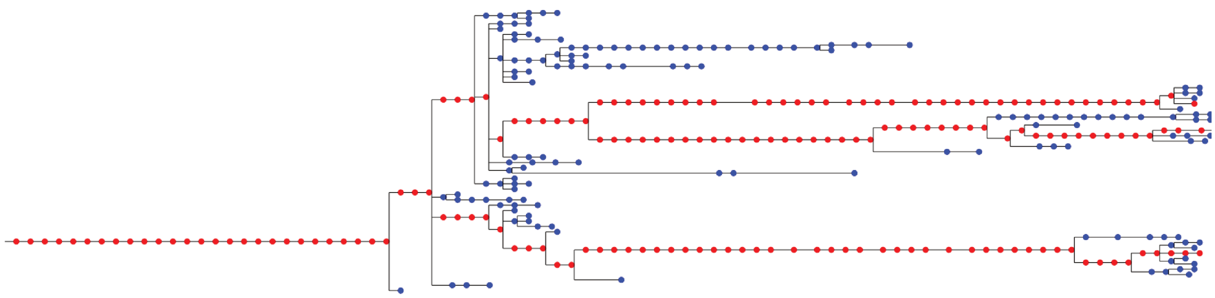
Figure 5.3: Illustration of the Area EF followed by the MMS filter with $t = 0.3$, (first column), $t = 0.5$ (second column), $t = 0.8$ (third column), and MSER (fourth column) applied to the sample images.

the methodology for simplifying the Max-Tree structure without losing the information necessary to solve image analysis and recognition problems.

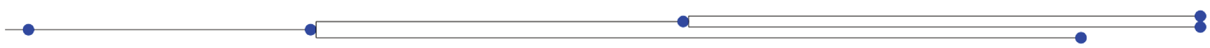
(a) Original Image

(b) Volume EF

(c) MMS-T



(d) Max-Tree



(e) Max-Tree after EF + MMS

Figure 5.4: Original image (a), after Volume EF (b), and after the MMS-T filter (c). Max-Tree graph (d), the red nodes are preserved by the EF. Max-Tree graph after EF and MMS filter.

The curve percentage of leaves filtered versus the nodes reduction rate after applying the methodology proposed is illustrated in Figure 5.5. All curves have a reduction rate higher than 10 when the abscissa axis reaches 0.8, and after that point the reduction rate starts to increase much faster. Our experience tells us that keeping between 1% and 5% of the relevant maxima of natural images usually yields a high nodes reduction rate, and preserves most of the image information.

5.4 Conclusions

In this chapter, it was presented the MMS filter along with a normalized threshold criterion (MMS-T) and a MSER criterion (MMS-MSER) to compute it. It was seen that a sensible case of the MMS filter happens, when there is a sub-branch that represents a great variation in size and/or shape of a connected component, in that case the MMS filter preserves the information of only one of the shapes. Also, the direct application of the MMS filter does not have much utility, since on average it reduces the number of Max-Tree nodes by a factor of 2, and it does not filter noise.

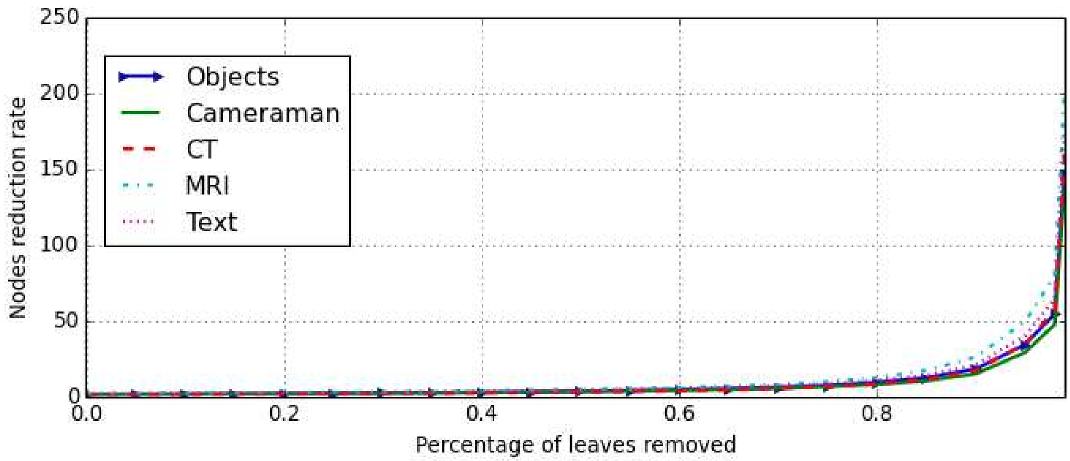


Figure 5.5: Curve percentage of relevant maxima preserved versus the nodes reduction rate after applying the methodology proposed.

The methodology presented that associates the EF to the the MMS filter has great potential to simplify the Max-Tree structure. First, it removes the irrelevant extrema, making the sub-branches average length higher, then it selects one node per sub-branch to be kept. The number of nodes in the resulting Max-Tree is bounded between $n + 1$ and $2n$, therefore by setting the parameter n of the EF, we have an estimate of the number of nodes of the resulting Max-Tree after applying the methodology proposed. The considerable reduction of Max-Tree nodes makes viable the extraction of invariant descriptors, such as the one proposed in (Forssen and Lowe, 2007), to be used by a classifier, in order to detect and recognize structures of interest in the image.

Chapter 6

Applications and Exploratory Studies

This chapter presents applications and exploratory studies of the methodology proposed in Section 5.3. The applications described are meant to illustrate the potential of the methodology to solve real problems, a discussion in depth is not intended. Three applications are illustrated: text location and recognition in natural scenes, object recognition, and image segmentation. The exploratory studies refer to the detection of image distinguished regions, and analysis of the robustness of the Max-Tree topology.

6.1 Applications

6.1.1 Text Location and Recognition in Natural Scenes

In this section, it is illustrated the procedure to perform text location and recognition in natural scenes using the EF associated to the MMS filter methodology. Two examples are analyzed. The first example requires a dual processing, because the text is either dark (minima) or bright (maxima). The second example is a license plate location and recognition example, which is a sub-problem of text location.

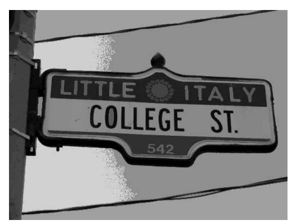
Text Location and Recognition: Dual Processing

A natural image which contains text written both in black and white is depicted in Figure 6.1(a). In order to locate all the text, it is necessary to perform the dual Max-Tree processing, since we are interested in both maxima and minima. The image is 384×512 pixels, and its corresponding Max-Tree has 13054 nodes of which 8249 are leaves. The characters in the image are expected to have high volume attribute values, therefore the first step is to apply the Volume EF with n set to 0.5% of the number of Max-Tree leaves. The Max-Tree resulting of the filtering has 2419 nodes and a SSIM index of 0.999. The filtered image is depicted in Figure 6.1(b), and it is clear that all white characters were preserved. Then, the MMS-T filter with $t = 0.5$ is applied, the resulting Max-Tree has 76 nodes, that is a reduction factor of around 172, when compared to the number of nodes of the initial Max-Tree, and the SSIM index is of 0.921. The image resulting of the MMS-T filter is shown in Figure 6.1(c).

The next step is to perform the dual processing with the negative of the original image, which is illustrated in Figure 6.1(d). The Max-Tree corresponding to the negative of the image has 13144 nodes of which 8416 are leaves. The Volume EF is applied. The Max-Tree resulting of the filtering has 1923 nodes and a SSIM index of 0.999. The corresponding image is depicted in Figure 6.1(e), and it is clear that all black characters were preserved. Finally, the MMS-T filter with $t = 0.5$ is applied, and the resulting Max-Tree has 75 nodes, that is a reduction factor of around 175, when compared to the number of nodes of the initial Max-Tree. The SSIM index of the filtered image is of 0.940. The image resulting of the MMS-T filter is illustrated Figure 6.1(f).

(a) Original Image

(b) Volume EF



(c) MMS-T

(d) Negative of the Original Image

(e) Volume EF

(f) MMS-T

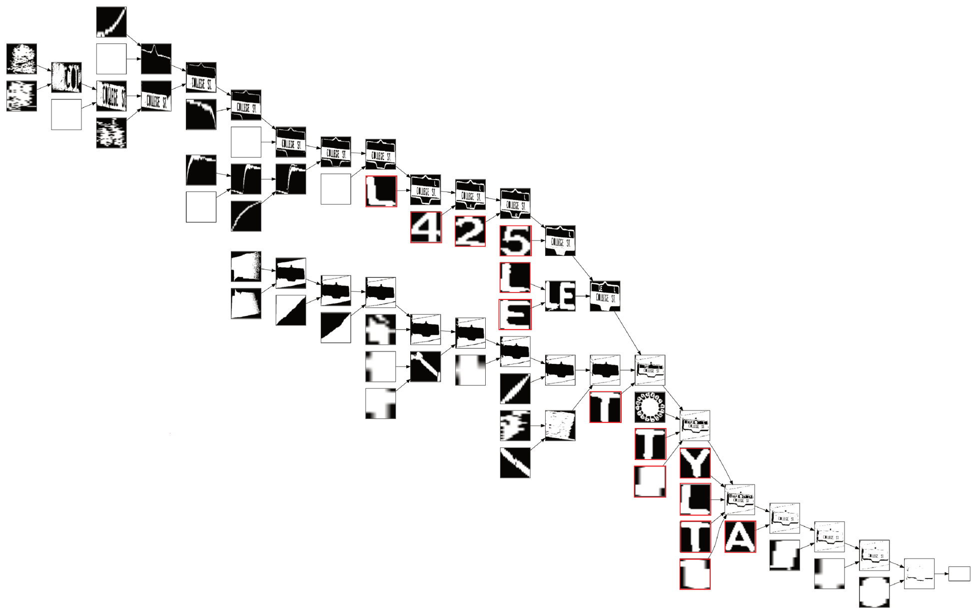
Figure 6.1: Illustration of the procedure to locate text in natural scenes. Original image (a), result of the Volume EF (b), result of the MMS-T filter (c). Negative of the original image (d), result of the Volume EF (e), result of the MMS-T filter (f).

The graphs corresponding to the Max-Tree resulting of the processing, and the dual processing are illustrated in Figure 6.2. Notice that all characters are present in the reconstruction of the Max-Tree nodes. The great reduction in the number of Max-Tree nodes allows the extraction of descriptors of the Max-Tree nodes that can be used as input to a classifier to locate and recognize the characters in the image. Usually it is not necessary to extract descriptors of all the remaining nodes, for instance the nodes that have an area value too high, such as the root, or an area value too low, such as some of the remaining leaves, can be filtered by some threshold criteria before extracting their descriptors. The reconstruction of the nodes that contain the text in the image is illustrated in Figure 6.3. The text in red was obtained from the Max-Tree processing of the image, and the text in green was obtained from the dual processing.

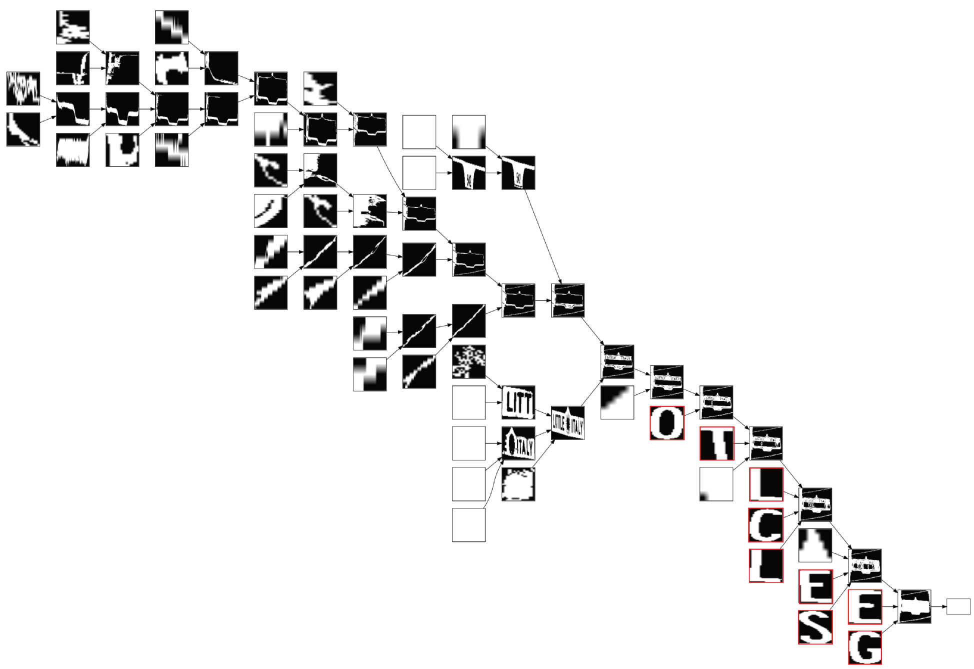
License Plate Location and Recognition

The same procedure used to locate and recognize text can be used to locate and recognize license plates, but with one advantage, most countries have plates with its characters either black or white, therefore the dual processing is not necessary. For instance, most Brazilian license plates contains 7 black characters. Therefore, we can build the Max-Tree of the negative of the image, since we want to analyze minima, and apply the Extinction filter. The plate characters in the negated image are expected to have high volume attribute values, but they are not necessarily the 7 extrema with highest extinction values in the negated image. For this reason, we choose a value for n greater than 7. We applied the Volume EF with $n = 15$, followed by the MMS-T filter with $t = 0.5$. We could have used a higher n value to ensure the characters information will not be lost. There is a trade-off, since we know the number of nodes in the resulting tree will be bounded between $n + 1$ and $2n$, the value of n should be chosen in a manner that the necessary information will be preserved, and the processing time will attend the application needs.

The results of the filtering are depicted in Figure 6.4. The Max-Tree of the original image has 24310 nodes of which 6031 are leaves. After the EF, it has 1202 nodes and a SSIM index of 0.9923. After the MMS-T filter, the number of nodes reduces to 28, a reduction rate of around 868, and the SSIM index of the image is of 0.970. The volume signature of the branches that contain the plate characters are illustrated in Figure 6.5. Note that there is a range of values where the volume values are very close and they do not vary much. The Max-Tree reconstruction of the portion corresponding to this range where the volumes are all close and vary little is illustrated in Figure 6.6. Note that the characters are recognizable in most of the nodes. The graph after the MMS-T filter is illustrated in Figure 6.7. The seven plate characters were preserved and they can be described and applied to a classifier to locate and recognize the license plate.



(a)



(b)

Figure 6.2: Max-Tree graphs after the MMS-T filter corresponding to the original image processing (a), and its dual processing (b). The connected components were interpolated for better visualization, and they are not in scale. The characters are marked in red.

Figure 6.3: Reconstruction of the Max-Tree nodes corresponding to the characters. The red characters come from the original image processing, and the green characters come from the dual processing.

(a) Original Image (b) Volume EF (c) MMS-T

Figure 6.4: Original image (a), result of the Volume EF (b), result of the MMS-T filter (c).

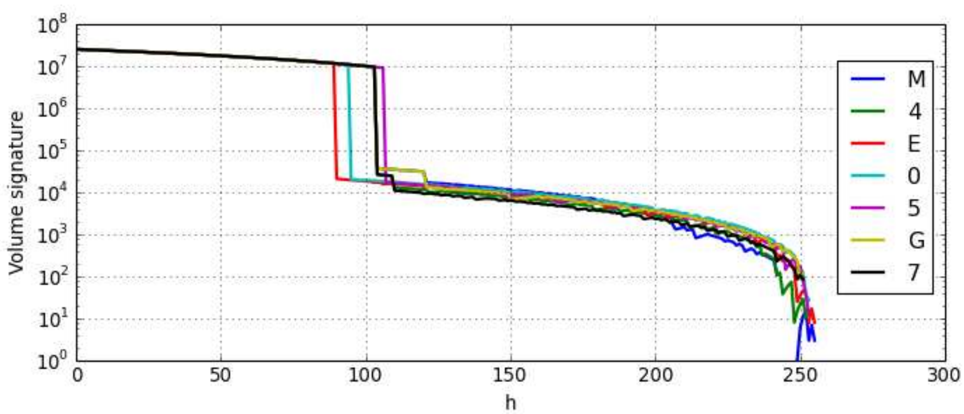


Figure 6.5: Volume signatures of the branches that contain the plate characters.

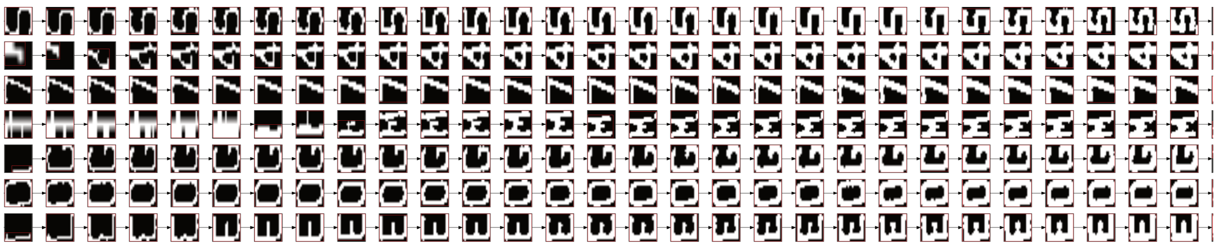


Figure 6.6: Illustration of the portion of the graph corresponding to the Max-Tree after the EF that contains the plate characters. The connected components were interpolated for better visualization, and they are not in scale.

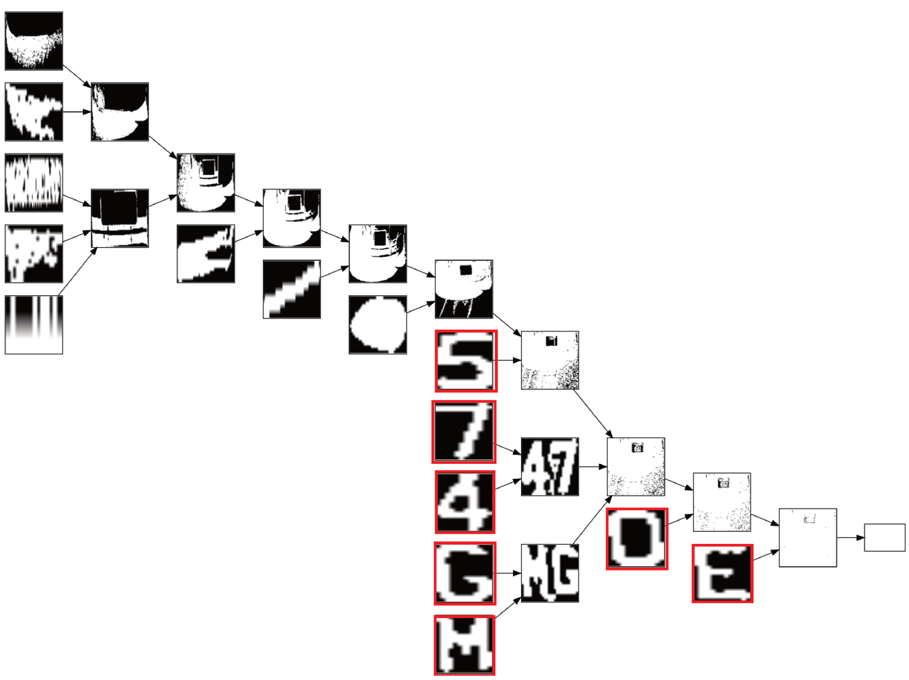
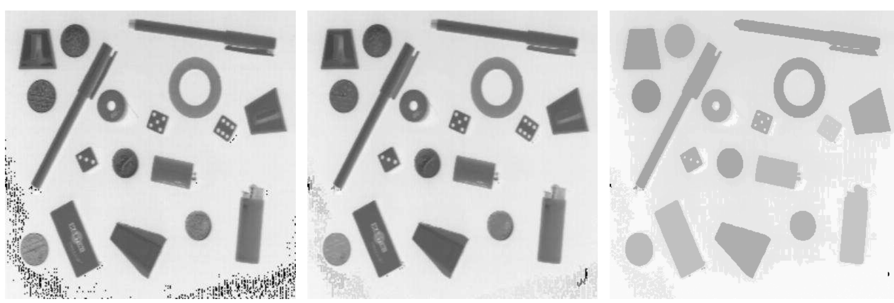


Figure 6.7: Graph corresponding to the Max-Tree after the EF and the MMS filter. The connected components were interpolated for better visualization, and they are not in scale. The characters are marked in red.

6.1.2 Object Recognition

The EF associated to the MMS filter methodology may also be used for object recognition. An image with many objects is depicted in Figure 6.8(a). We have to process the Max-Tree of the negative of the image, since the objects are dark. The Max-Tree has 12627 nodes, of which 2949 are leaves. The resulting image after applying the Volume Extinction filter with $n = 20$ is shown in Figure 6.8(b). After the EF, the number of Max-Tree nodes reduced to 2614. After the MMS-MSER filter, the number of Max-Tree nodes is 36, and the resulting image is displayed in Figure 6.8(c). We can see that the shapes of all relevant objects were preserved, therefore it is possible to extract descriptors from the connected components of the Max-Tree nodes, to recognize the objects in the image. We believe that the association of the Extinction filter to the MMS-MSER detects more robust distinguished regions, since the regions detected attend the MSER stability criterion and they are also on the path of the most persistent extrema in the image.

The Max-Tree graph after the EF, and the MMS-MSER filter is illustrated in Figure 6.9, and the graph with its edges length proportional to the attribute *nlevels* of the nodes is shown in Figure 6.10. The shape of all objects of the image were preserved in the Max-Tree nodes. The reconstruction of the nodes that represent the interest objects of the original image is shown in Figure 6.11.



(a) (b) (c)

Figure 6.8: Illustration of the object recognition procedure. Original image (a), result of the Volume EF (b), result of the MMS-T filter (c).

6.1.3 Segmentation

The methodology proposed may also be used to perform image segmentation. A lung CT image is shown in Figure 6.12(a). Since we are interested in segmenting the lungs, which represent dark structures in the image, we build the Max-Tree corresponding to the negative of the original image. The original Max-Tree has 43585 nodes of which 20975

Figure 6.9: Graph corresponding to the Max-Tree after the EF and the MMS filter. The connected components were interpolated for better visualization, they are not in scale.The objects are marked in red.

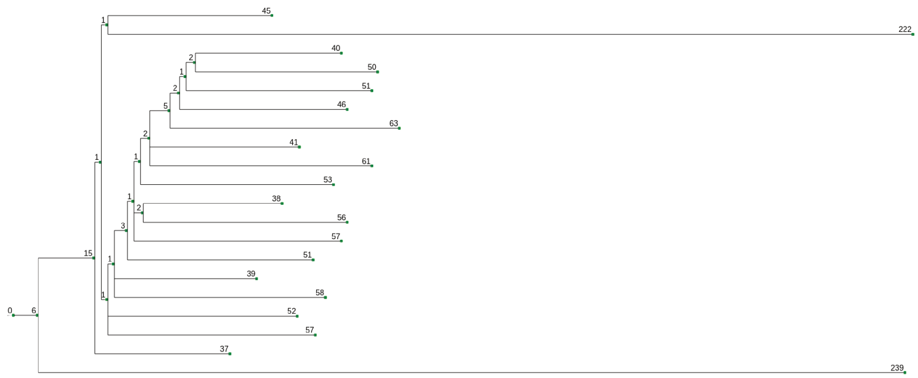


Figure 6.10: Graph corresponding to the Max-Tree after the EF and the MMS filter. The edges length are proportional to the attribute *nlevels* of the nodes

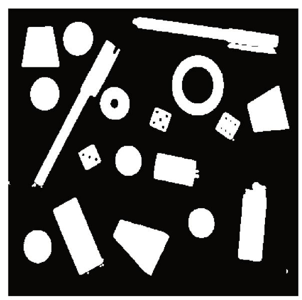


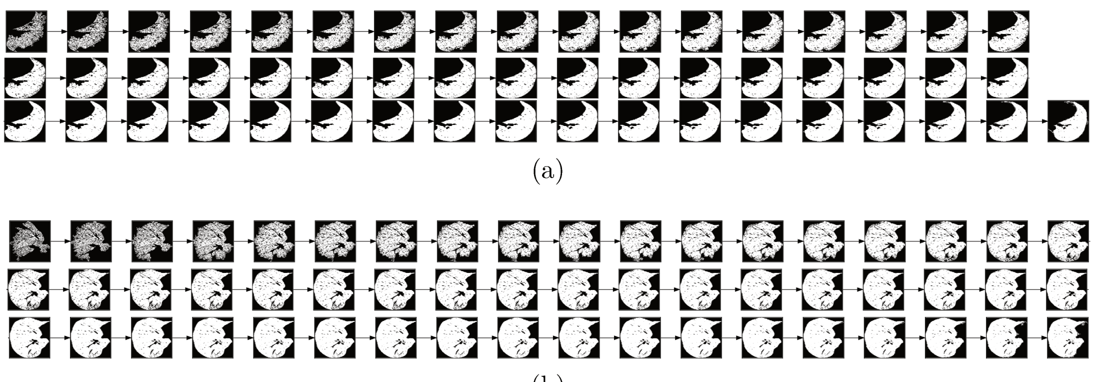
Figure 6.11: Reconstruction of the Max-Tree nodes that correspond to the objects in Figure 6.8(a).

are leaves. The next step is to apply the Volume EF with $n = 10$. In this case of lung CT segmentation the parameter n can be easily set manually, since we have an estimate of the number of relevant extrema. The relevant extrema are the two lungs and the background, we super-estimated the n parameter to 10. The Max-Tree resulting of the EF has 419 nodes, and a SSIM index of 0.998. The reconstruction of the sub-branches that contain the left and the right lungs is illustrated in Figure 6.13. Before the Volume EF, the Max-Tree nodes that contained possible lung segmentations would be spread around multiple sub-branches. Now the segmentation procedure is much easier, since the lungs are contained in only one sub-branch each. All we have to do is to use a criterion to choose two Max-Tree nodes that are good segmentations of the left and right lungs.

The result of applying the MMS-T filter with $t = 0.5$ is illustrated in Figure 6.12(c), the number of nodes now is 19, and the SSIM index of the resulting image is of 0.924. The reconstruction of the nodes that contain the lungs is shown in Figure 6.14, and we can see visually that it corresponds to a good segmentation of the lungs. Notice that this segmentation also delineated the major blood vessels inside the lungs.

(a) Original Image (b) Volume EF (c) MMS-T

Figure 6.12: Illustration of the procedure to segment the lungs. Original image (a), result of the Volume EF (b), result of the MMS-T filter (c).



(a)

(b)

Figure 6.13: Reconstruction of the sub-branches containing the right (a) and left (b) lungs. The connected components increase from left to right and top to bottom.

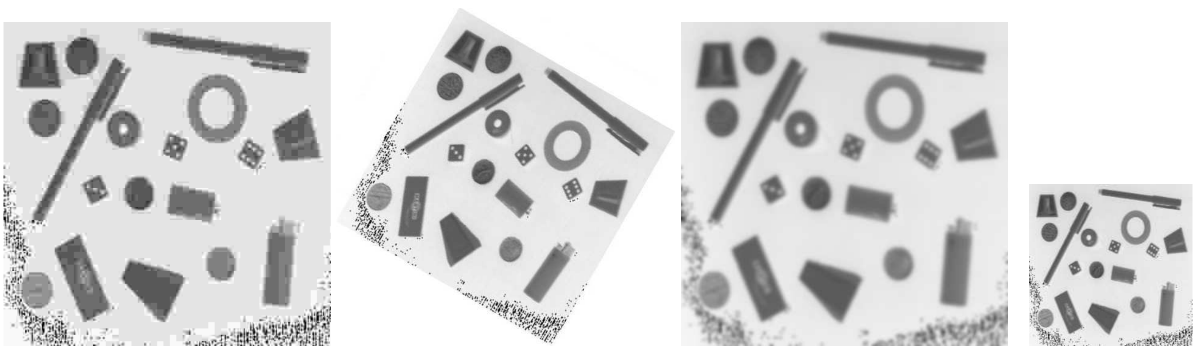
Figure 6.14: Reconstruction of the Max-Tree nodes that correspond to the lungs in Figure 6.12(a).

6.2 Exploratory Studies

In this section, it is presented some exploratory studies concerning the robustness of the Max-Tree topology for an image that has suffered different types of transformations. Also, we compare the performance of the methodology proposed and EF as a pre-processing step of the MSER method against the usual MSER to detect image distinguished regions.

6.2.1 Analysis of the Max-Tree Topology Robustness

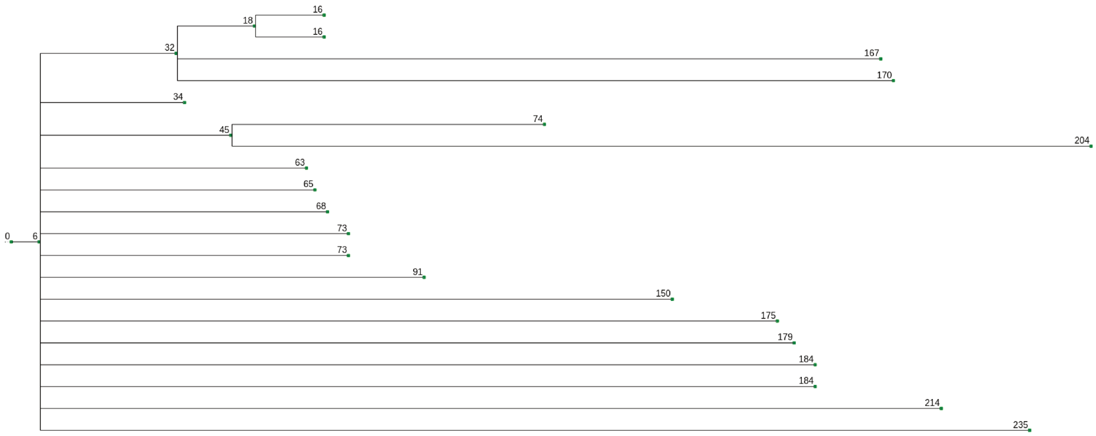
In this section, it is investigated the robustness of the Max-Tree topology through an example. We will use the Objects image, Figure 6.8(a). The same image JPEG compressed, rotated, scaled, and blurred with a Gaussian filter are shown in Figure 6.15. The Volume EF with $n = 20$ followed by the MMS-MSER is applied to the negative of all images.



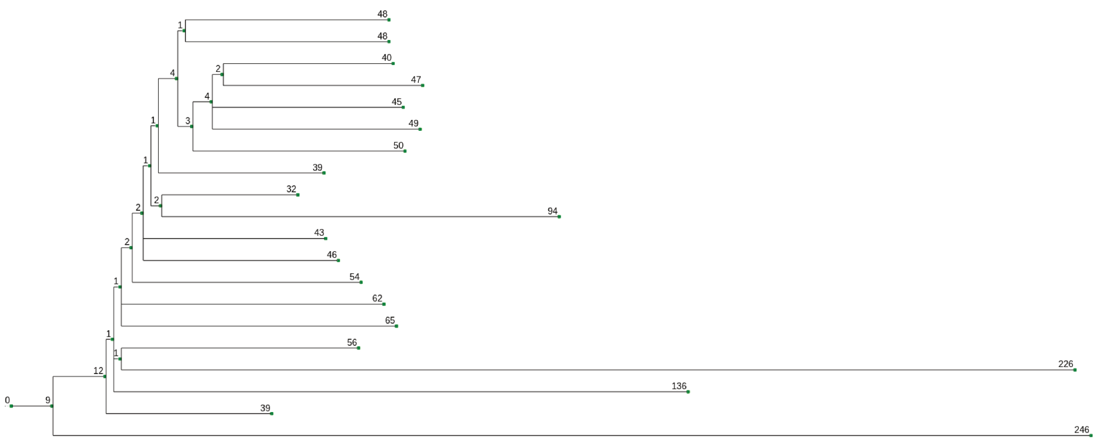
(a) JPEG Compressed (b) Rotated (c) Blurred (d) Scaled

Figure 6.15: Objects image after several different types of transformations.

The Max-Tree graph of the original image after filtering is illustrated in Figure 6.10. The Max-Tree graphs corresponding to the degraded images after filtering are illustrated in Figure 6.16 and Figure 6.17. We can see that the topology of the Max-Trees changed in all cases. The most extreme case was the JPEG compressed image case, where most leaves of the resulting Max-Tree are connected directly to the first ramification node, whcih in this example is the result closer to the expected topology since tha image is constituted of many disjoint objects on a background. Although the topology varied, we can see that the main differences of the topology are due to the internal nodes, but the graphs shows that these internal nodes have a small persistence, i.e. low value of the *nlevels* attribute, therefore an alternative is to filter these nodes, in order to achieve a tree with a topology more similar to the topology we would expect. Also, the hierarchy between the objects is always preserved and all objects are still detectable in all cases. This simple example indicates that the Max-Tree topology is not robust, but perhaps a posterior filtering step can make it more robust.

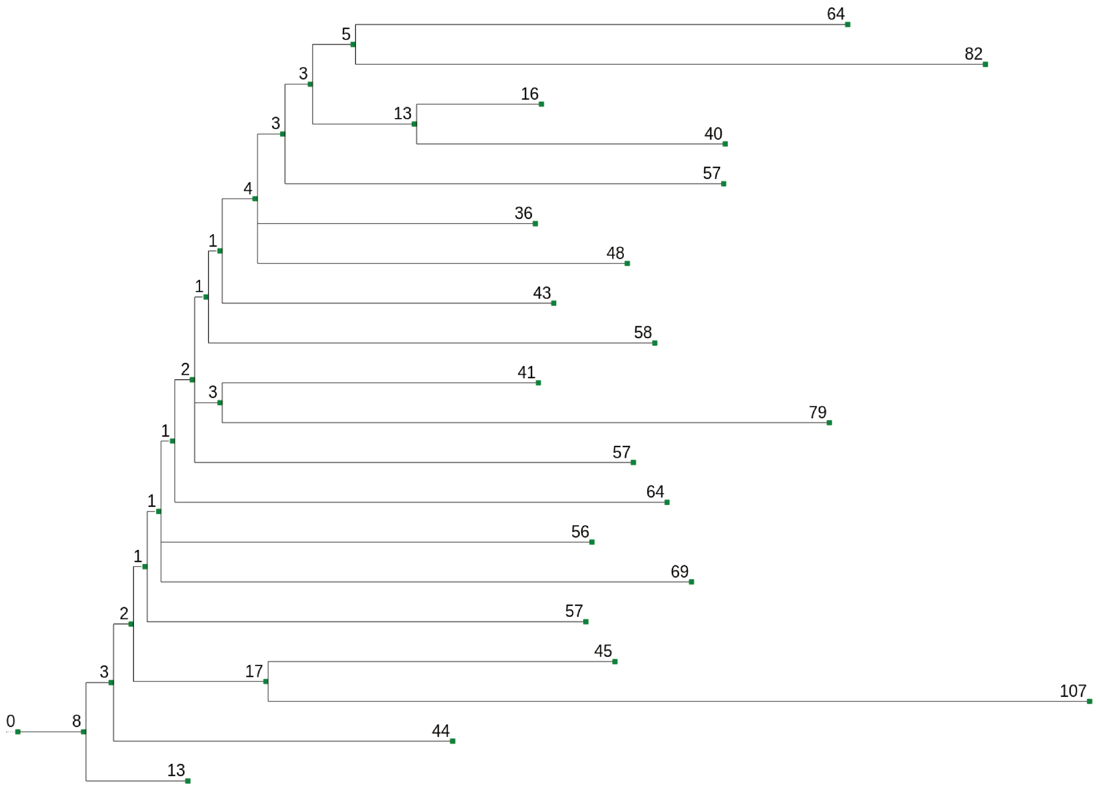


(a) JPEG Compressed

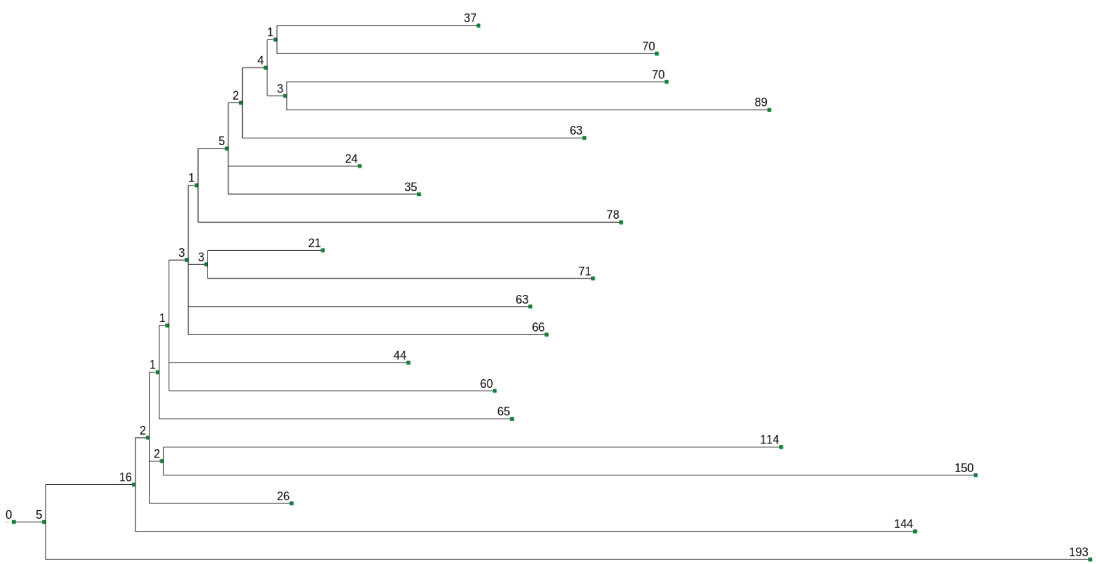


(b) Rotated

Figure 6.16: Graphs of the resulting Max-Trees part 1. The edges length are proportional to the *nlevels* attribute of the nodes.



(a) Blurred



(b) Scaled

Figure 6.17: Graphs of the resulting Max-Trees part 2. The edges length are proportional to the *nlevels* attribute of the nodes.

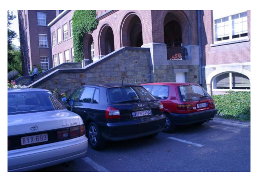
6.2.2 Detection of Distinguished Regions

In this section, it is evaluated the performance of the methodology proposed applied to the detection of distinguished regions. The methodology proposed is compared to the MSER method, which is one of the detectors with highest repeatability rates (Mikolajczyk et al., 2005). Also, we analyze the use of EF as a pre-processing step before the extraction of MSER regions. This gives us an insight on how much information is lost when using EF with different values of n.

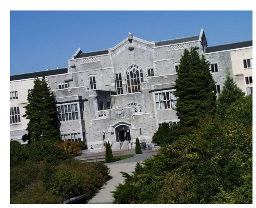
In the simulations, it was used the benchmark dataset and the protocol intended to evaluate affine region detectors proposed by Mikolajczyk et al. (2005). The dataset is composed of structured and textured scenes divided in eight groups, where each group suffered a different type of image transformation. One sample image of each group is depicted in Figure 6.18. The ground truth of all image groups is provided by mapping the regions detected to a reference image of each group using homographies. The matching between two regions is based on the amount of overlap between the detected region in the reference image and the detected region in the other image, projected onto the reference image using the homography relating them. In order to compute this overlap, every region size is normalized to a radius of 30 pixels, and if the overlap error is smaller than 40%, it is considered a match.

(a) Graffiti

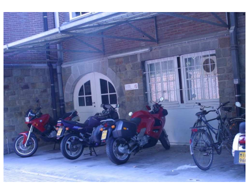
(b) Boat



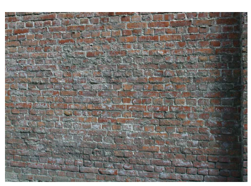
(c) Leuven



(d) UBC

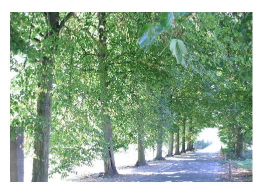


(e) Bike



(f) Wall

(g) Bark



(h) Trees

Figure 6.18: Sample images of the benchmark dataset for comparing affine region detectors.

The MSER regions where computed using the binary file provided in the paper (Mikolajczyk et al., 2005). It uses connectivity $C4$, therefore the Max-Tree used in these experiments also uses connectivity $C4$. In the first round of experiments, it was analyzed the use of EF as a pre-processing step before the extraction of MSER regions. The image was filtered using the EF and the $MSER+$ regions were computed. Then, the negative of the image was filtered and the $MSER-$ regions were computed. The results were compared to the usual MSER method. The parameter n of the EF was tested with 15%, 10%, 5%, and 1%. The results of the group *Bike* of the dataset using Height, Area, and Volume EF are depicted in Figure 6.19, Figure 6.20 and Figure 6.21, respectively. The results for all groups in the dataset are illustrated in Appendix A.

We can see that either using the Height or Area or Volume EF the repeatability scores using n equal 15%, 10% and 5% are very similar to the MSER results. This is also another strong evidence that most of the image information is contained at the most relevant extrema. The repeatability results only have a more accentuated drop for n equal 1%.

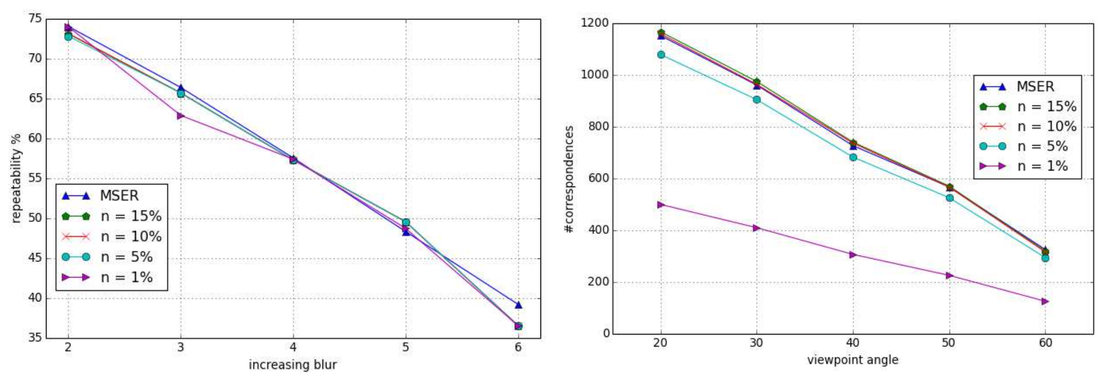


Figure 6.19: Repeatability results of the group of images *Bike* using Height EF as MSER pre-processing.

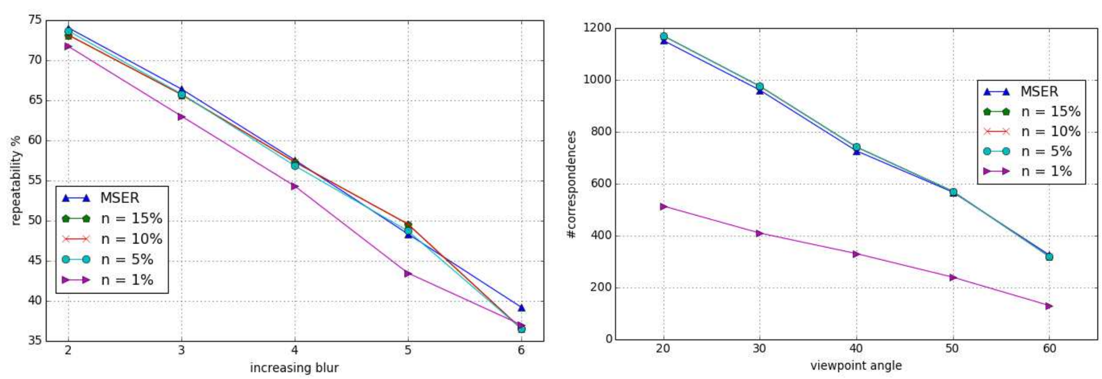
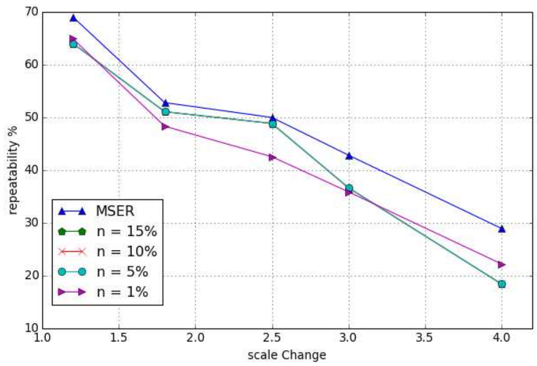


Figure 6.20: Repeatability results of the group of images *Bike* using Area EF as MSER pre-processing.



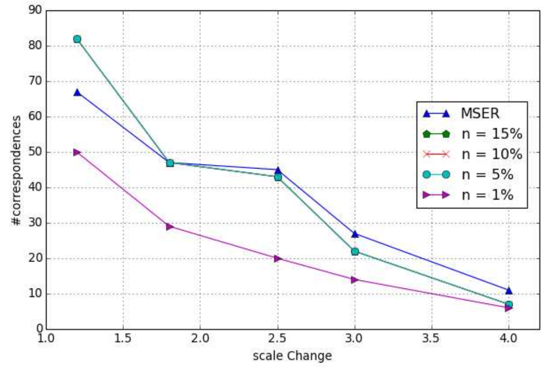
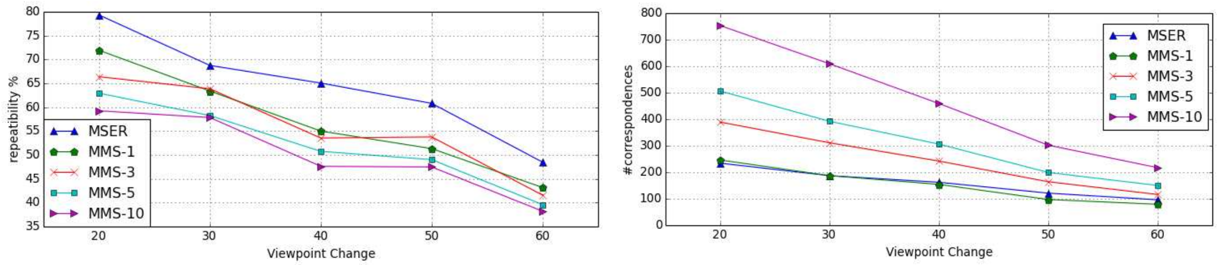
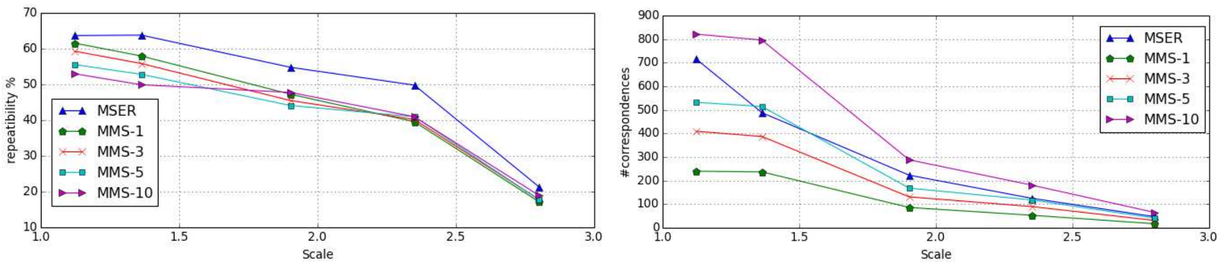


Figure 6.21: Repeatability results of the group of images *Bike* using Volume EF as MSER pre-processing.

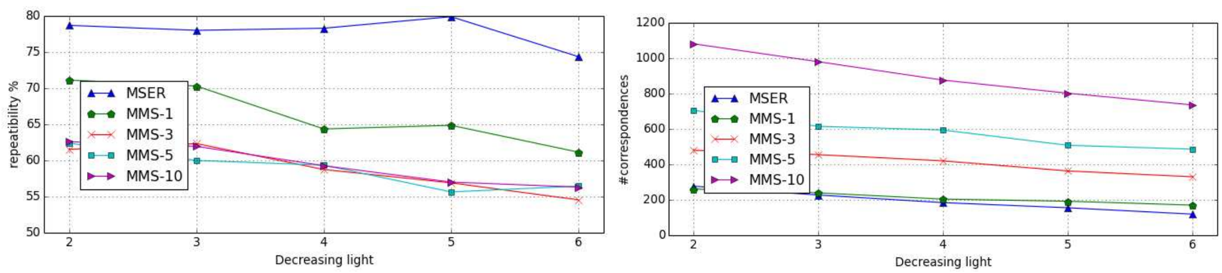
In the second round of experiments, the MMS methodology was tested using the Volume EF and the MMS-T filter with $t = 0.5$. The parameter n of the EF was set as 1%, 3%, 5%, and 10% of the number of Max-Tree leaves. It was considered a distinguished region, all the remaining Max-Tree nodes with area higher than 30 pixels and smaller than 1% of the total image area, and that have a diversity higher than 20%, i.e. at least 20% of the pixels in the region detected differ from any other region detected. For every region found, an ellipse having the first and second moments as the detected region is fitted. This is an affine covariant construction method. The absolute and relative repeatability results are depicted in Figure 6.22 and Figure 6.23. In most cases the MSER relative repeatability was higher than our method, but by setting the parameter n, we were able to obtain higher absolute repeatability rates, which is a desirable property, since the MSER detector has the problem of detecting fewer regions than other methods (Mikolajczyk et al., 2005).



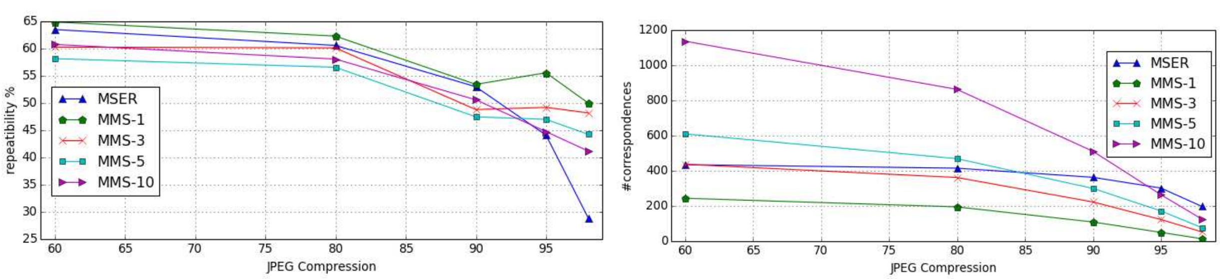
(a) Graffiti



(b) Boat

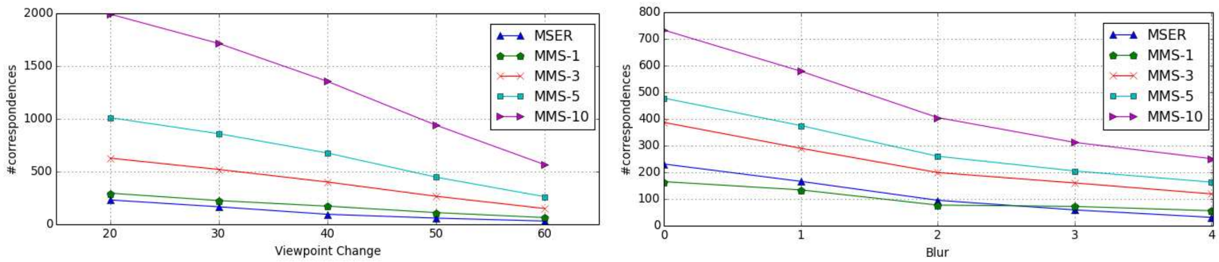


(c) Leuven

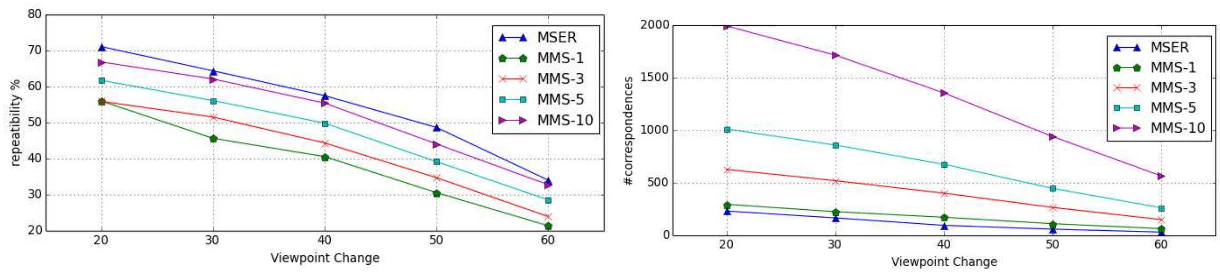


(d) UBC

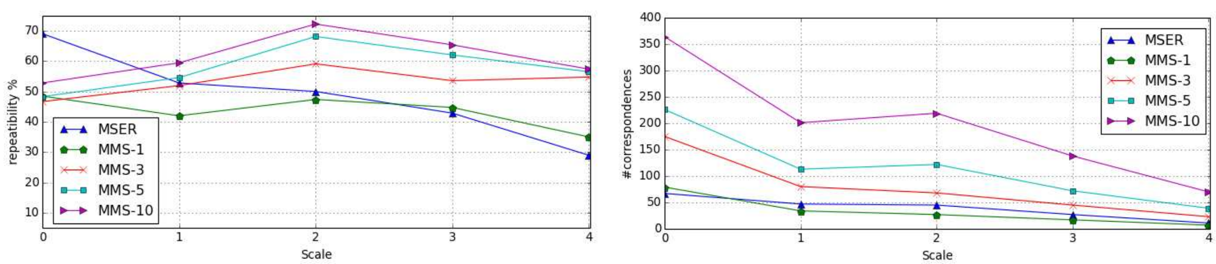
Figure 6.22: MMS methodology repeatability results, part 1.



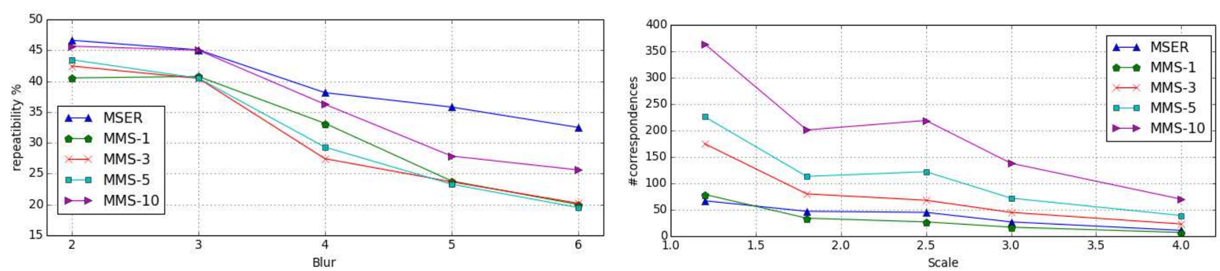
(a) Bike



(b) Wall



(c) Bark



(d) Trees

Figure 6.23: MMS methodology repeatability results, part 2.

6.3 Conclusions

This chapter presented possible applications of the methodology that associates the EF and the MMS filter proposed in Section 5.3. It was seen that this combination greatly simplifies the number of Max-Tree nodes, and it is usually able to keep the necessary information to solve the problem at hand. The considerable reduction of Max-Tree nodes makes viable the extraction of invariant descriptors to be used by a classifier, in order to solve image analysis and recognition problems. One of the disadvantages of the methodology is that when looking for structures that may be either bright or dark, it is necessary to perform a dual processing with the original image and its negative.

It was shown through a simple example that the Max-Tree topology is not robust, but the hierarchical relationship always holds. The repeatability scores showed that the MMS methodology has the advantage of controlling the number of detected regions. We also saw that the EF as a pre-processing step before detecting the MSER regions using n values as low as 5% obtains repeatability scores similar to the ones obtained by the usual MSER method. The conclusion is that most of the image robust information is located on the path of the most persistent extrema in the Max-Tree.

Chapter 7

Conclusions

7.1 Conclusions

This work formalized the definition of Extinction filters, and defined the Maximal Max-Tree Simplification filter. In order to do that, new Max-Tree concepts were introduced, such as composite nodes, and sub-branches. Their interpretation in terms of image processing were explained. It was seen that the methodology proposed greatly simplifies the Max-Tree structure. The EF removes irrelevant extrema from the image, and its effect on the Max-Tree is to reduce its number of nodes and sub-branches. Also, it increases the average sub-branch length. It was shown that the number of Max-Tree nodes after applying the EF followed by the MMS filter is bounded between $n+1$ and $2n$, where n is the number of tree leaves kept by the EF. The relationship between Extinction filters and attribute filters was also analyzed, and we explained why use Extinction filters instead of attribute filters in our methodology.

The potential of the methodology was illustrated through applications involving text location and recognition, object recognition, and image segmentation. Exploratory studies showed that the methodology has potential to be used as a detector of distinguished regions, and that the EF set with low values of n still preserves most of the image information. Also, the Max-Tree topology is not very robust, when analyzing the same image that suffered different transformations, but the hierarchical relationship is always preserved. The overall conclusion is that the methodology proposed greatly simplifies the Max-Tree structure, while preserving the necessary information to solve image analysis and pattern recognition problems.

7.2 Future works

The considerable reduction of the nodes in the Max-Tree resulting of the simplification methodology proposed makes viable the extraction of invariant descriptors to be used by a classifier. An immediate continuation of this work is to continue the exploratory studies presented in this work by investigating ways to better characterize the Max-Tree nodes, and apply it to the detection of more robust distinguished regions that profit from the tree hierarchy information, which many descriptors do not consider, and compare it with popular detectors/descriptors, such as SIFT (Lowe, 2004), SURF (Bay et al., 2006) and MSER (Matas et al., 2002).

Other interesting continuation would be the extension of the maximal simplification filter to other image representing structures, such as the Tree of Shapes (Monasse and Guichard, 1998), which is a contrast independent and non-redundant representation of the image, and the proposal of other criteria, such as a shape criterion, to compute the filter.

Furthermore, another possible continuation is to compare our methodology results with state-of-the-art methods in the literature, in this work possible applications were illustrated, but they were not investigated in depth.

7.3 Publications

As a result of this work, it was published a paper entitled "Maximal Max-Tree Simplification" (Souza et al., 2014) to the 22^{nd} International Conference on Pattern Recognition. The paper defines the EF and the MMS filter. Also, it introduces the simplification methodology, and illustrates the text location and recognition, and the object recognition applications.

Bibliography

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

M. Donoser H. Bischof. 3d segmentation by maximally stable volumes (msvs). In *18th International Conference on Pattern Recognition, 2006. ICPR 2006.*, volume 1, pages 63–66, 2006.

E. Carlinet and T. Géraud. A fair comparison of many max-tree computation algorithms (extended version of the paper submitted to ismm 2013). *Computing Research Repository (CoRR)*, abs - 1212.1819, 2012.

M. Donoser and H. Bischof. Efficient Maximally Stable Extremal Region (MSER) Tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 553–560, 2006.

M. Donoser, C. Arth, and H. Bischof. Detecting, tracking and recognizing license plates. In *Proceedings of the 8th Asian conference on Computer vision - Volume Part II*, ACCV'07, pages 447–456, Berlin, Heidelberg, 2007. Springer-Verlag.

J. Fabrizio and B. Marcotegui. Fast implementation of the ultimate opening. In *International Symposium on Mathematical Morphology (ISMM)*, 2009.

P. Forssen and D. Lowe. Shape descriptors for maximally stable extremal regions. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

M. Grimaud. New measure of contrast: the dynamics. volume 1769, pages 292–305, 1992. doi: 10.1117/12.60650.

R. Jones. Connected filtering and segmentation using component trees. *Comput. Vis. Image Underst.*, 75(3):215–228, 1999.

D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, pages 384–393, 2002.

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.*, pages 761–767, 2004.

C. Merino-Gracia, K. Lenc, and M. Mirmehdi. A head-mounted device for recognizing text in natural scenes. In *Proceedings of the 4th international conference on Camera-Based Document Analysis and Recognition*, CBDAR'11, pages 29–41, Berlin, Heidelberg, 2012. Springer-Verlag.

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65:2005, 2005.

P. Monasse and F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Trans. on Image Proc*, 9:860–872, 1998.

L. Najman and M. Couprie. Quasi-linear algorithm for the component tree. In *In SPIE Vision Geometry XII*, pages 98–107, 2004.

L. Najman and M. Couprie. Building the component tree in quasi-linear time. *Trans. Img. Proc.*, 15(11):3531–3539, November 2006.

P. Salembier and A. Oliveras. Practical extensions of connected operators. In *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 5 of *Computational Imaging and Vision*, pages 97–110. Springer US, 1996.

P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing*, 4(8):1153–1160, 1995.

P. Salembier and M. Wilkinson. Connected operators. *IEEE Signal Processing Magazine*, 26(6):136–157, 2009.

P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, 1998.

J. Serra and P. Salembier. Connected operators and pyramids. pages 65–76, 1993.

A. Silva and R. Lotufo. New extinction values from efficient construction and analysis of extended attribute component tree. In *XXI Brazilian Symposium on Computer Graphics and Image Processing, 2008. SIBGRAPI '08.*, pages 204–211, 2008.

R. M. Souza, L. Rittner, R. Machado, and R. Lotufo. Maximal max-tree simplification. submitted pending acceptance, 2014.

E. R. Urbach and M. H. F. Wilkinson. Shape-Only Granulometries and Grey-Scale Shape Filters. In Hugues Talbot and Richard Beare, editors, *Proceedings of the VIth International Symposium - ISMM 2002*, volume 6. CSIRO Publishing, April 2002.

C. Vachier. Extinction value: a new measurement of persistence. In *IEEE Workshop on Nonlinear Signal and Image Processing*, volume I, pages 254–257, 1995a.

C. Vachier. *Extraction de Caracteristiques, Segmentation d'Image et Morphologie Mathematique*. Phd, L' École Nationale Supérieure des Mines de Paris, 1995b.

L. Vincent. Morphological area openings and closings for grey-scale images. In Ying-Lie O, Alexander Toet, David Foster, HenkJ.A.M. Heijmans, and Peter Meer, editors, *Shape in Picture*, volume 126, pages 197–208. Springer Berlin Heidelberg, 1994.

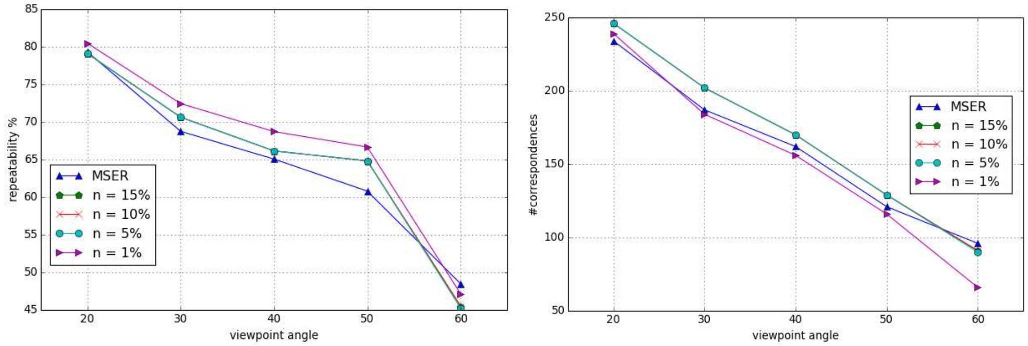
W. Wang, Q. Jiang, X. Zhou, and W. Wan. Car license plate detection based on mser. In *International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 3973–3976, 2011.

M. A. Westenberg, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Volumetric attribute filtering and interactive visualization using the max-tree representation. *IEEE Transactions on Image Processing*, 16(12):2943–2952, 2007.

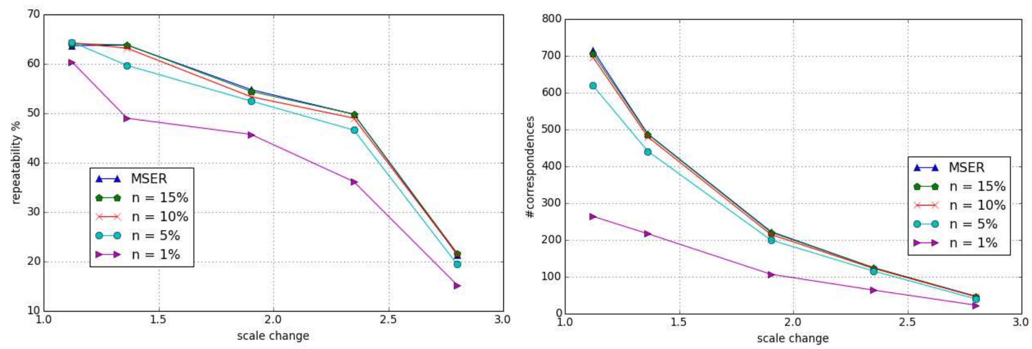
Appendix A

Repeatability Results

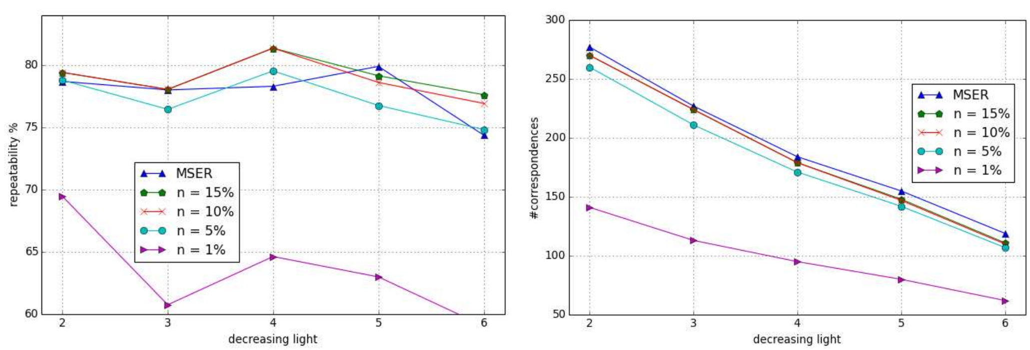
This appendix illustrates all the results of using the EF as a pre-processing step before the extraction of MSER regions that were initially presented in Section 6.2.2 of this dissertation. The results using the Height EF are depicted in Figure A.1 and Figure A.2. The results using the Area EF are depicted in Figure A.3 and Figure A.4, and the results using the Volume EF are depicted in Figure A.5 and Figure A.6.



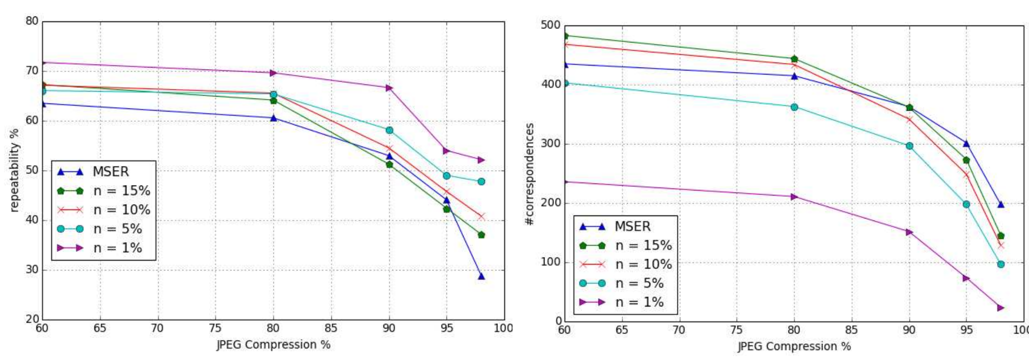
(a) Graffiti



(b) Boat

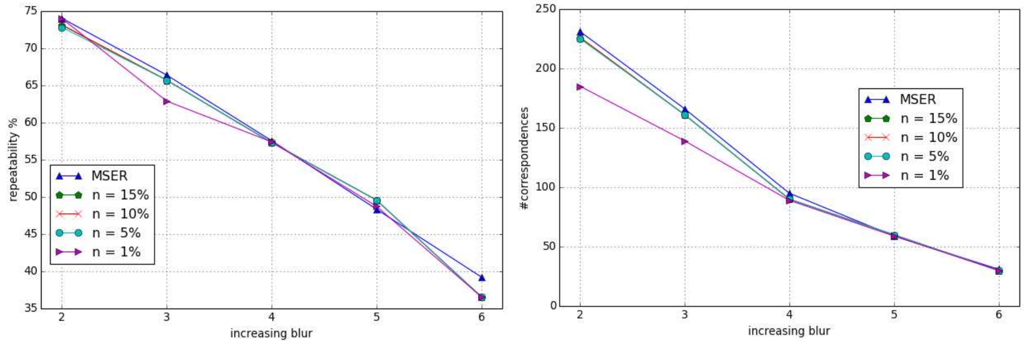


(c) Leuven

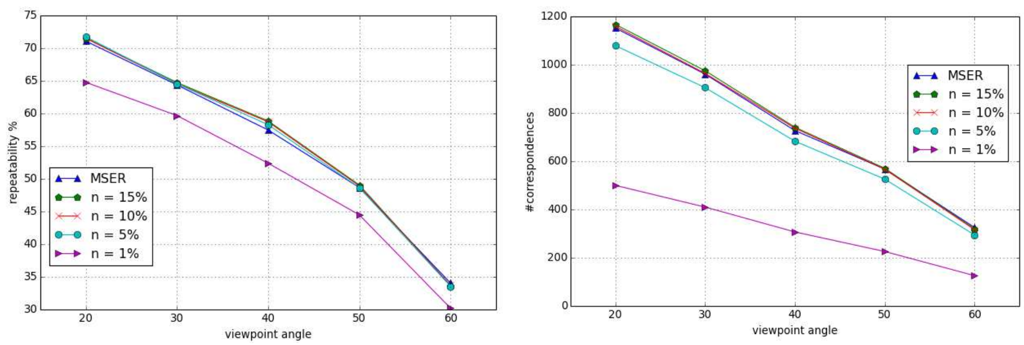


(d) UBC

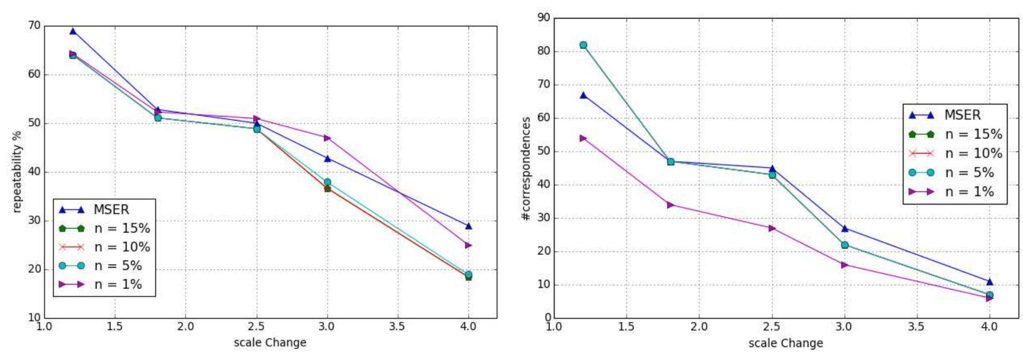
Figure A.1: Repeatability results using Height EF as MSER pre-processing, part 1.



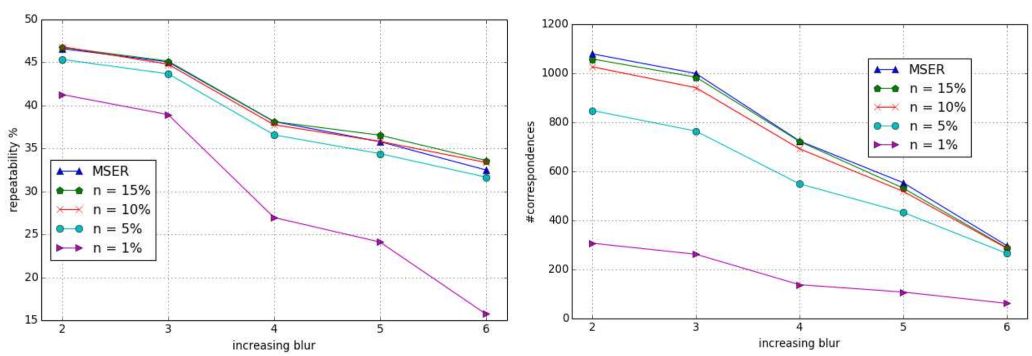
(a) Bike



(b) Wall

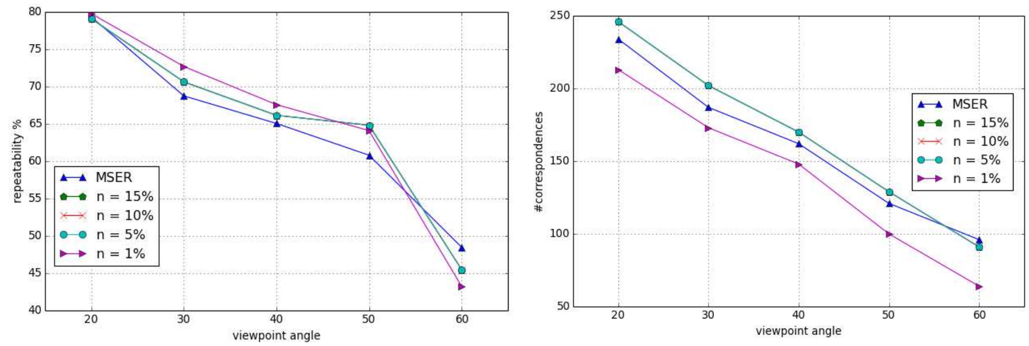


(c) Bark

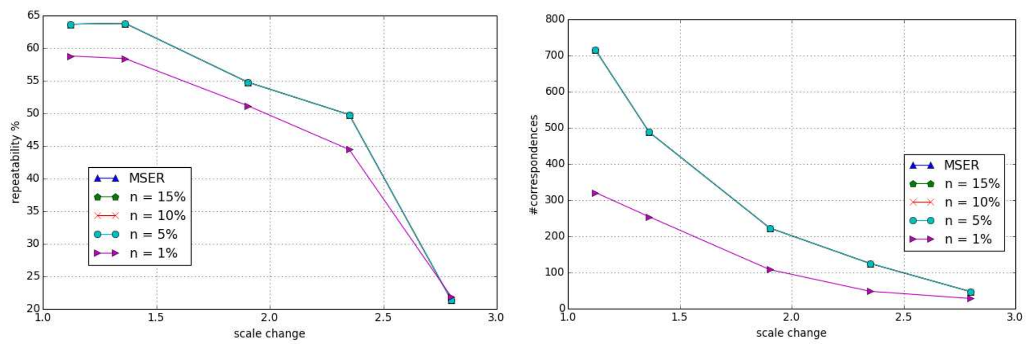


(d) Trees

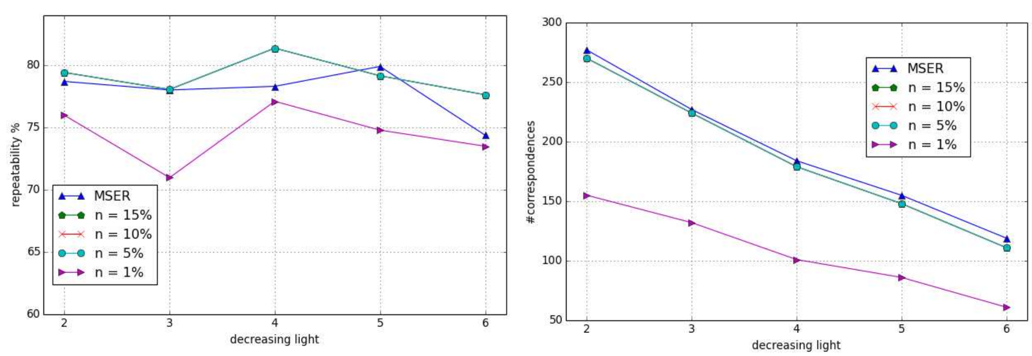
Figure A.2: Repeatability results using Height EF as MSER pre-processing, part 2.



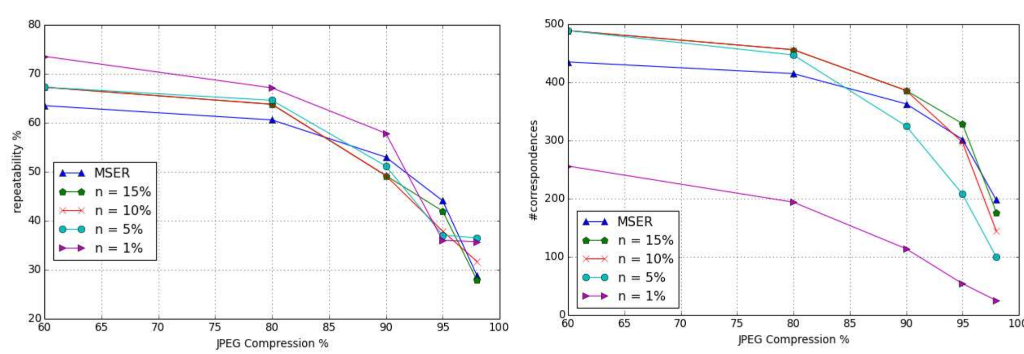
(a) Graffiti



(b) Boat

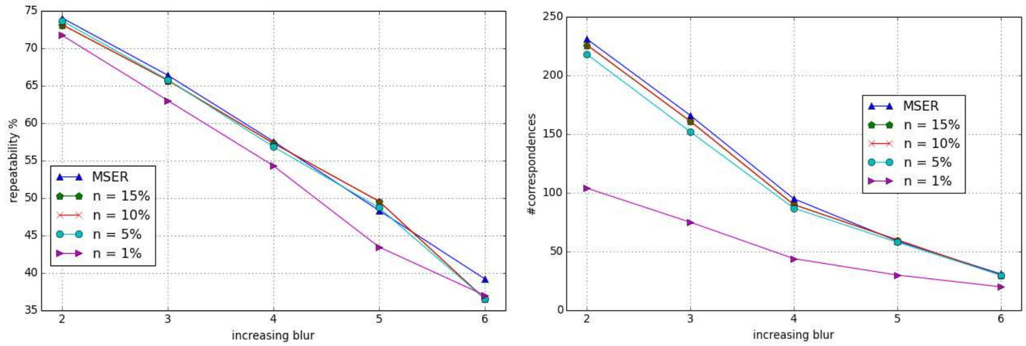


(c) Leuven

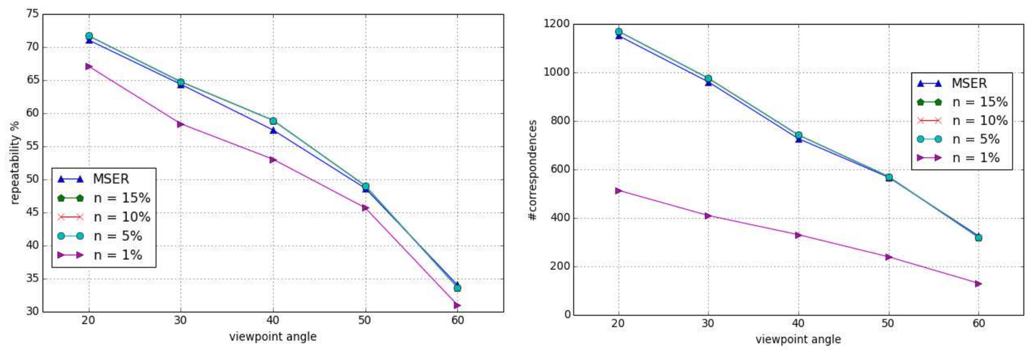


(d) UBC

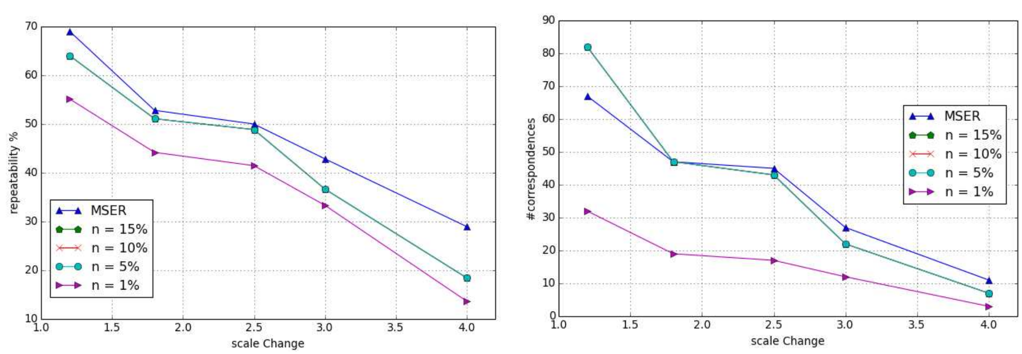
Figure A.3: Repeatability results using Area EF as MSER pre-processing, part 1.



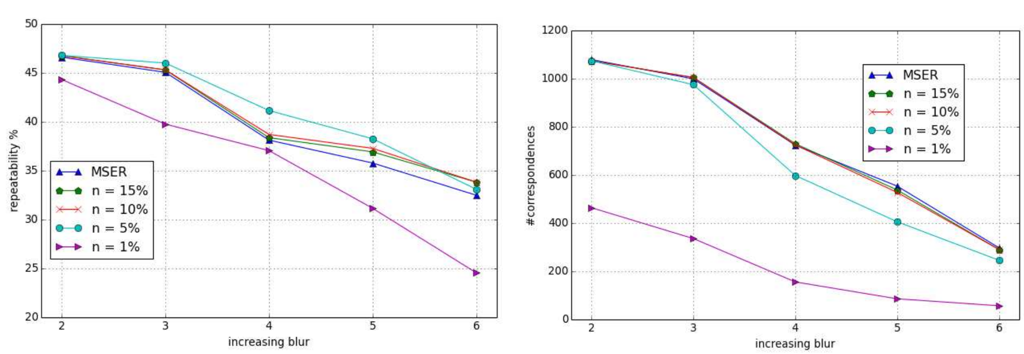
(a) Bike



(b) Wall

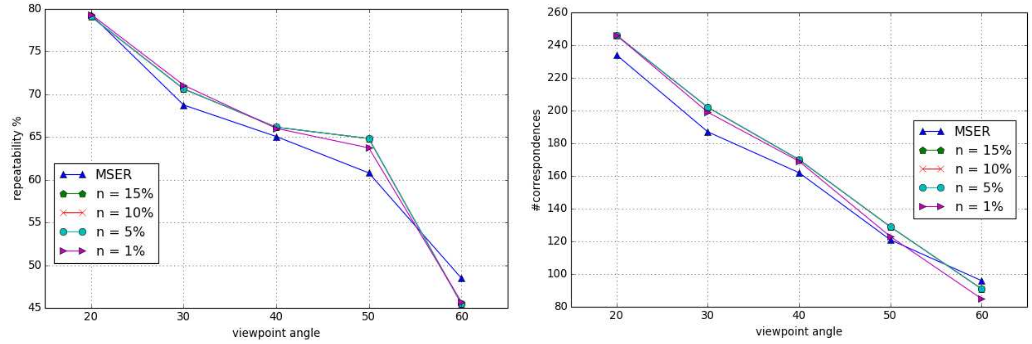


(c) Bark

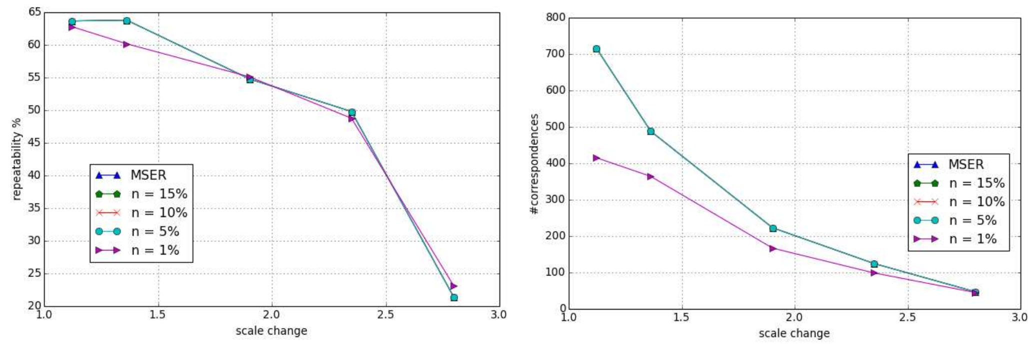


(d) Trees

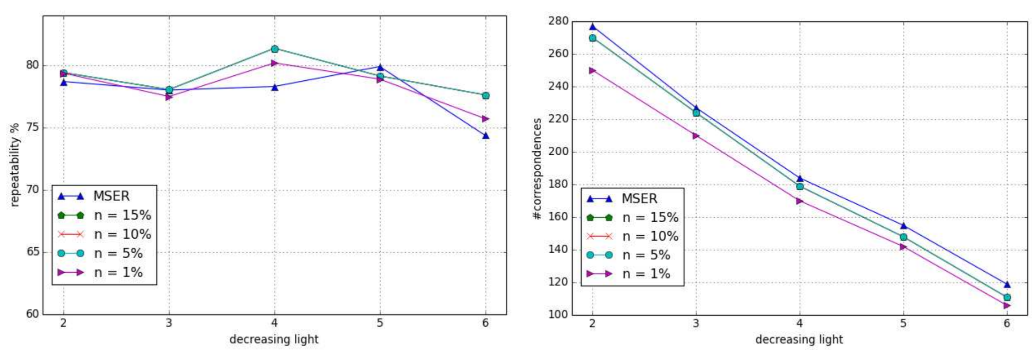
Figure A.4: Repeatability results using Area EF as MSER pre-processing, part 2.



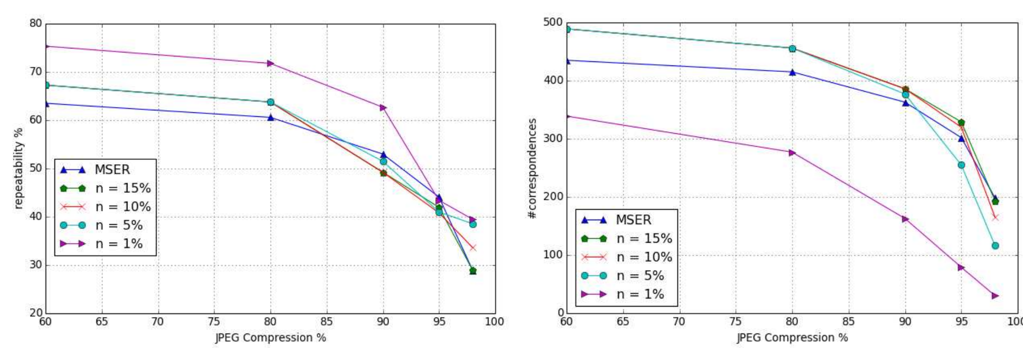
(a) Graffiti



(b) Boat

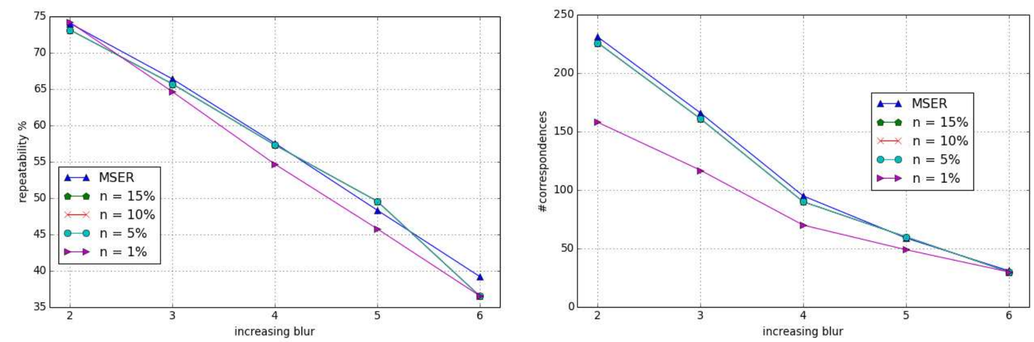


(c) Leuven

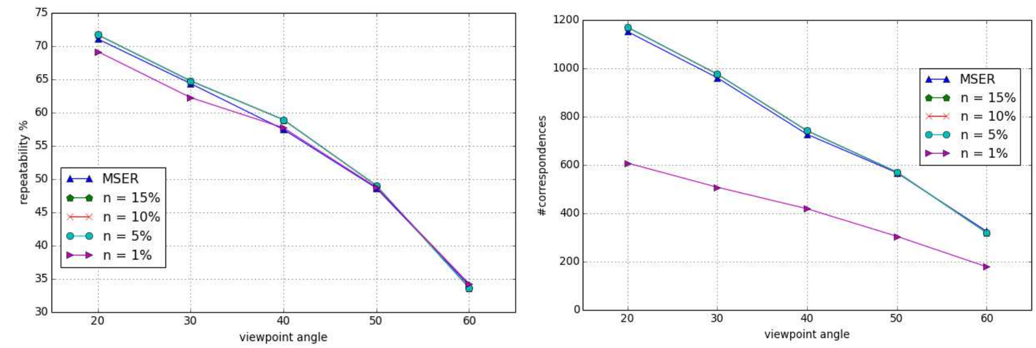


(d) UBC

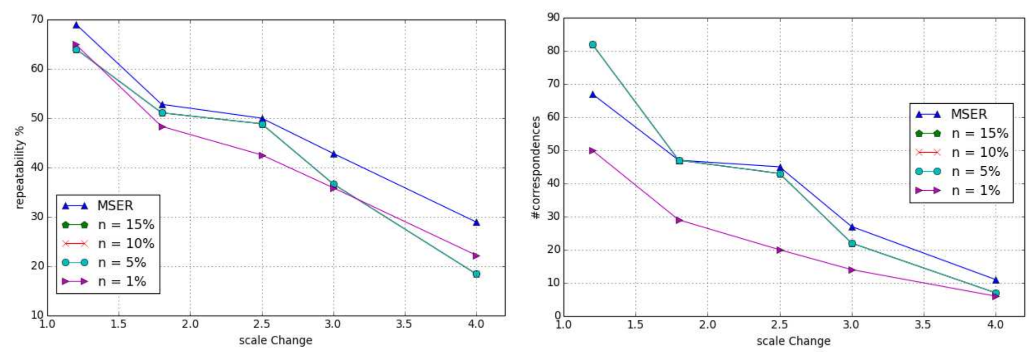
Figure A.5: Repeatability results using Volume EF as MSER pre-processing, part 1.



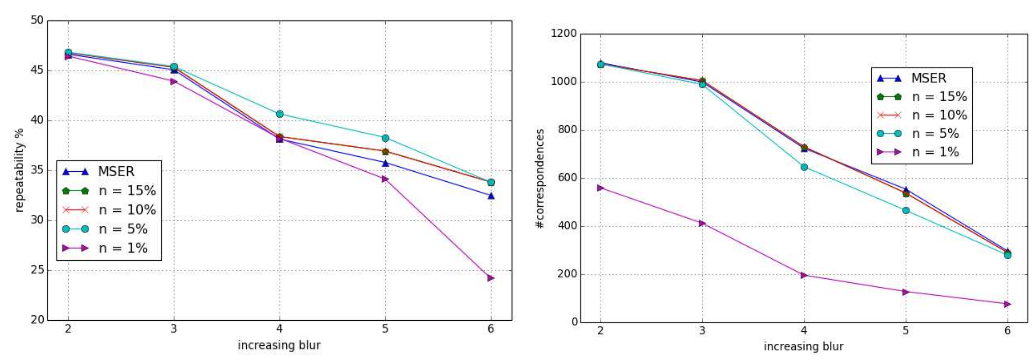
(a) Bike



(b) Wall



(c) Bark



(d) Trees

Figure A.6: Repeatability results using Volume EF as MSER pre-processing, part 2.

Appendix B

Processing Time

The average processing time curve of the methodology proposed englobing the Max-Tree construction, the EF and the MMS filter application for the *cameraman* image (Figure B.1) using different values of n is depicted in Figure B.2. The simulations were made at the Adessowiki platform. The number of repetitions to compute the average processing time was of 10 and the parameter t of the MMS-T was set as 0.5, and the Δ used to compute the MSER stability was set as 5. The Max-Tree construction code was developed using the language C/C++ and the filters were programmed in Python, which is a slow interpreted language. Although the complexity and time analysis of the algorithms was not the main focus of this dissertation, we believe that an optimized code would allow the methodology proposed to be employed in real time applications.

Figure B.1: Cameraman image. Dimensions: 256×256 pixels.

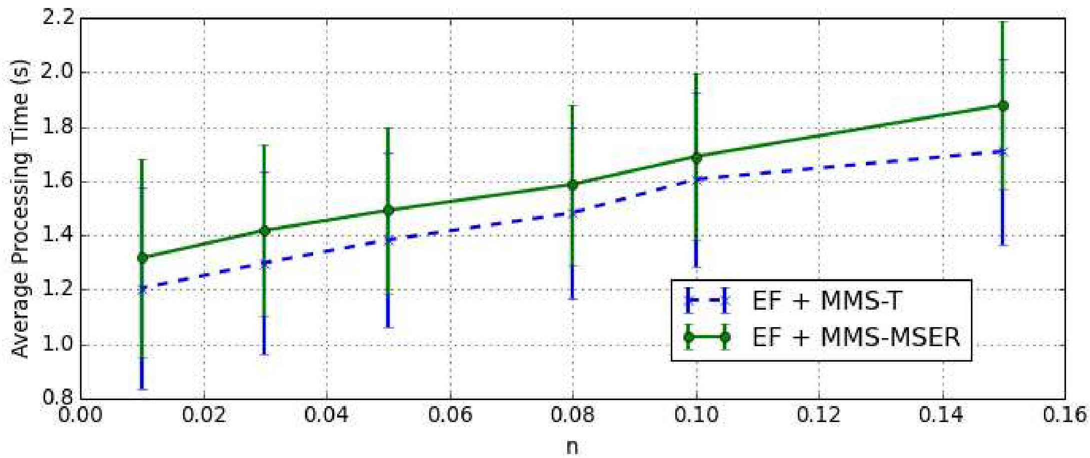


Figure B.2: Average processing time englobing the Max-Tree construction, the EF and the MMS filter.