Colorido: Identificação da Cor Dominante de Fotografias

José Carlos Xavier Abelho Amante

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Júri

Presidente: Professor Joaquim Armando Pires Jorge
Orientador: Professor Manuel João Caneira Monteiro da Fonseca
Vogal: Professora Maria Teresa Caeiro Chambel

Novembro, 2011
Resumo

Nos últimos anos a necessidade de bons sistemas de recuperação tem ganho cada vez mais importância. O enorme volume de imagens que todos os dias coleccionamos nos nossos computadores e aparelhos móveis faz da recuperação de imagens uma tarefa difícil mas necessária. Em geral, a cor é uma das características visuais mais distintas, fazendo com que a maioria das soluções existentes para recuperar e apresentar imagens aos utilizadores sejam baseadas nas cores dominantes.

No entanto os sistemas actuais não têm em conta a forma como as pessoas identificam as cores dominantes. Este problema muitas vezes leva a resultados mediocres e abaixo das expectativas. Existem muitos estudos sobre as cores dominantes em imagens, mas muito poucos adotam a percepção da visão humana como a principal preocupação. Aqui é onde nos propomos a inovar. No âmbito deste trabalho, desenvolvemos uma solução para identificar cores dominantes em imagens, o mais próximo possível da forma como as pessoas as percebem.

Descrevemos uma solução que combina a segmentação não-uniforme do espaço de cor HSV com teoria fuzzy e uma paleta com 12 cores. Para identificar a melhor solução desenvolvemos seis algoritmos e avaliámo-los. Testes experimentais com utilizadores provaram que a nossa abordagem é adequada para a resolução do problema identificado, e que os dois melhores algoritmos são o 3x3 Segmentation e o Fuzzy Histogram. Estes algoritmos conseguem identificar as mesmas cores dominantes que os utilizadores em 69,1% e 64,7% das vezes, respectivamente.

Keywords: cores dominantes; percepção humana; espaço de cor HSV; teoria fuzzy.
Abstract

In the last years the need for good retrieval systems has gained more importance. The massive volume of images we collect every day in our computers and mobile devices makes the retrieval of images a difficult but needed task. In general, color is one of the most distinguishable visual features, causing the majority of the existing solutions to retrieve and present images to users to be based on the dominant colors (DCs).

However, the current systems do not take into account the way people identify the dominant colors. This problem often lead to poor results, below expectations.

There are a lot of studies about dominant colors in pictures but very few adopt the human vision perception as the main concern. Here is where we propose to innovate. Within this work, we developed a solution to identify DCs in pictures, as close as possible to the way people perceive them.

We describe a solution which combines the non-uniform segmentation of the HSV color space with fuzzy theory and a palette with 12 colors. To identify the best solution we developed 6 different algorithms and evaluated them. Experimental tests with users proved that our approach is suitable to the resolution of the identified problem, and that the two best algorithms are the 3x3 Segmentation and the Fuzzy Histogram. These algorithms can identify the same DCs as the users in 69,1% and 64,7% of the times, respectively.

Keywords: dominant colors; human perception; HSV color space; fuzzy theory.
Acknowledgements

I would like to express my gratitude to all the participants directly involved in this work. In the first place, to Professor Manuel João Fonseca for the monitoring and guidance provided. His advises were of great value. With his enthusiasm, his inspiration and his great effort to explain things clearly he helped a lot throughout my thesis-writing period. He provided support and lots of good ideas. In the end, quite honestly, I only regret not having had the pleasure of attending his course of RVIM (Recuperação e Visualização de Informação Multimédia) at Instituto Superior Técnico.

I would also like to thank the many people who patiently answered to both the surveys we performed. The time spent by all was quite valuable and outstanding. The opinions provided in the early stages of the surveys were also highly appreciated.

Last but not least, I would like to thank my parents and my girlfriend, who stood with stoicism and patience especially over the last months.
Contents

1 Introduction 1
  1.1 Problem Description ........................................... 1
  1.2 Objectives ...................................................... 2
  1.3 Developed Solution ........................................... 2
  1.4 Contributions and Achieved Results ......................... 3
  1.5 Organization of the Document .................................. 4

2 Related Work 5
  2.1 Histogram Based Methods ..................................... 5
    2.1.1 MPEG-7 color descriptors overview .................... 7
    2.1.2 Histogram calculus with structured bins .............. 8
    2.1.3 Another dominant color extraction based on MPEG-7 .... 10
    2.1.4 Unsupervised clustering algorithm for histogram estimation .... 11
    2.1.5 Fuzzy and neuro-fuzzy logic approaches .............. 12
  2.2 Segmentation Based Methods ................................ 13
    2.2.1 Spatial localization of color regions ................ 15
    2.2.2 Color regions using Color Coherence Vector ........... 16
    2.2.3 Dominant color regions using the Sub-Block technique ... 17
    2.2.4 Segmentation into homogeneous colored and textured regions ... 18
    2.2.5 Classification of dominant colors through fuzzy linguistic expressions ... 19
    2.2.6 Dominant colors segmentation using morphological clustering ... 21
    2.2.7 Dominant color selection by spatial and color domain analysis .... 22
  2.3 Correlogram Based Methods ................................... 24
    2.3.1 Spatial correlation between pairs of colors ........... 25
    2.3.2 Dominant color regions by a co-occurrence matrix ... 26
2.3.3 Dominant colors extracted from auto-correlogram based wavelet decomposition ........................................ 27

2.4 Hybrid/Statistical Based Methods ................................................................. 29
2.4.1 Color indexing based on central moments of each color channel .......... 30
2.4.2 Ant colony clustering algorithm for dominant colors extraction ........ 31
2.4.3 Dominant colors by a dual segmentation using crisp and fuzzy approaches 32
2.4.4 Dominant color extraction based on multi-dimensional particle swarm optimization ........................................ 34

2.5 Discussion .................................................................................................. 36
2.6 Summary .................................................................................................... 38

3 Dominant Colors Identification by Users 41
3.1 The Pictures .............................................................................................. 41
3.2 The Procedure ............................................................................................ 42
3.3 Results ......................................................................................................... 45
3.4 Discussion .................................................................................................. 46
3.5 Summary .................................................................................................... 48

4 The Developed Algorithms 49
4.1 Preprocessing ............................................................................................. 50
4.2 Segmentation of the Color Space ................................................................. 51
4.2.1 The 12 Colors Palette ................................................................................ 51
4.2.2 The Strict Achromatic Colors .................................................................. 52
4.2.3 The Fuzzy Membership Functions ............................................................ 53
4.3 Algorithms for Dominant Colors Identification ............................................ 58
4.3.1 The Fuzzy Histogram Approach ............................................................. 58
4.3.2 The 3x3 Segmentation of the Image Space .............................................. 60
4.3.3 The Ellipsoid Mask .................................................................................. 62
4.3.4 The Saliency Map Based Mask ............................................................... 63
4.3.5 The K-Means Clustering ........................................................................ 63
4.3.6 The Classic Strict Histogram ................................................................. 66
4.4 Results ......................................................................................................... 66
4.4.1 The Fuzzy Histogram Approach ............................................................. 68
List of Figures

2.1 The 9 fuzzy membership functions of Hue. ........................................ 20
2.2 Fuzzified HSI color space. ................................................................. 33

3.1 The first series of 5 pictures from which people had to write the names of the
dominant colors. .................................................................................. 42
3.2 An example of the first series of questions of the on-line survey, showing the
picture and the space for users to write the name of the DCs (below). ........ 42
3.3 The second series of 20 pictures from which people had to choose the dominant
colors. .................................................................................................. 43
3.4 An example of the second series of questions of the on-line survey, showing the
palette of 12 colors (on the right) and the colors and the corresponding rating
scales (below) ........................................................................................ 44
3.5 The 3 Ishihara test pictures. ................................................................. 44
3.6 The age groups of the 39 effective answers. ........................................... 45
3.7 Statistic values for the first series of 5 pictures of the on-line survey. ........ 46
3.8 All the statistic values from the on-line survey results. Note the average number
of DCs per picture increased from 1,94 in the first series of pictures to 3,39 on
the second series. Also the most frequent number of DCs identified (the mode)
increased from 1 to 3. ........................................................................ 46
3.9 Known vs. unknown color names (per picture) the users wrote in the first series
of the survey. ....................................................................................... 47

4.1 Modules of the developed solution. ..................................................... 49
4.2 The RGB and HSV values for the palette of 12 colors. ......................... 52
4.3 The HSV color space represented with our 3 thresholds, which delimit the black,
gray and white colors. ......................................................................... 54
4.4 Plot of the 9 Hue membership functions. ............................................. 54
4.5 Representation of the circular Hue when Value is maximum. ................. 55
4.6 The difficulty with the boundaries of two consecutive hues. .................... 56
4.7 Plot of the 3 Saturation membership functions. ................................. 57
4.8 Plot of the 3 Value membership functions. ........................................ 57
4.9 Some hues can belong to two different membership functions. ............... 59
4.10 A typical fuzzy histogram (on the right) for the picture on the left. The 3 more
dominant colors are: red, gray and green. ............................................. 59
4.11 An example of a picture with the 3x3 segmentation represented (the numbers are
only representative of the 9 divisions). .................................................. 60
4.12 The fuzzy histograms for each of the 9 sub-images. Note that the order of the 9
histograms is the same as in the image presented before, i.e., in the first row we
have sub-images 1, 2 and 3, in the second row 4, 5 and 6 and in the last row 7, 8
and 9. .................................................................................................. 61
4.13 The normalized output from the application of the 3x3 Segmentation algorithm.
From our voting system we get: red as the most dominant color, gray as the
second dominant color, etc. ................................................................. 61
4.14 The bright zone of the ellipsoid mask is where the weight is maximum. ...... 62
4.15 The Saliency Map based mask (on the right) for the picture on the left. ...... 64
4.16 The K-Means clustering for 3 clusters (on the right) for the picture on the left.
Note that the "green" of the trees and the "gray" of the road are clustered together. 65
4.17 The result of the application of the K-Means Clustering algorithm on the image
from the previous Figure. Note that the algorithm only finds 3 DCs and this
number is an input of the method. ......................................................... 65
4.18 All the algorithms against the survey results, for Pictures 1-5. .................. 67
4.19 All the algorithms against the survey results, for Pictures 6-25. ............... 67
4.20 An example of matching of the DCs of an algorithm with the DCs from the users.
We only consider the DCs if their relative weight (on the right of each color) is
above 10%, and until it reaches the number of DCs of the survey above 10% (in
this case: 4). ..................................................................................... 68
4.21 An example of the missing colors between a given algorithm and the results from
users. Here, for this algorithm, 3 colors are missing (Orange, Yellow and Black). 69
5.1 The 20 chosen pictures used in the new survey to evaluate the algorithms. . . . . 76
5.2 An example of one version of the answers document, where we can see the numbers of the pictures, the 3 corresponding DCs, and the rating scale for the evaluation of users. ................................................................. 77
5.3 Results from the evaluation of the 4 chosen algorithms. ............................. 78
5.4 Student’s t-test with the associated probabilities between the 3x3 Segmentation algorithm and the others. ................................................................. 79
5.5 Average grade and standard deviation for each image of each algorithm. ....... 80

A.1 The architectural modules of the developed software. ............................... 100
B.1 The first page from the first survey, where we asked age, sex and the use of glasses or contact lenses (all required answers). ................................. 104
B.2 The second page from the first survey, where we tested for color blind people. . 105
B.3 The third page from the first survey, where we asked people to write the dominant colors they saw (1/2). ................................................................. 106
B.4 The third page from the first survey, where we asked people to write the dominant colors they saw (2/2). ................................................................. 107
B.5 The fourth page from the first survey, where we asked people to classify the dominant colors they saw (1/4). ................................................................. 108
B.6 The fourth page from the first survey, where we asked people to classify the dominant colors they saw (2/4). ................................................................. 109
B.7 The fourth page from the first survey, where we asked people to classify the dominant colors they saw (3/4). ................................................................. 110
B.8 The fourth page from the first survey, where we asked people to classify the dominant colors they saw (4/4). ................................................................. 111
B.9 The fifth page from the first survey, where we asked people to classify the dominant colors they saw (1/4). ................................................................. 112
B.10 The fifth page from the first survey, where we asked people to classify the dominant colors they saw (2/4). ................................................................. 113
B.11 The fifth page from the first survey, where we asked people to classify the dominant colors they saw (3/4). ................................................................. 114
B.12 The fifth page from the first survey, where we asked people to classify the dominant colors they saw (4/4). ................................................................. 115

C.1 The first page from the evaluation survey, where we asked people for their age, sex and a very simplified color blind test. ................................. 118

C.2 The 20 chosen pictures used in the survey to evaluate the algorithms.  .......... 119

C.3 The A version of the answers document, used for people answer to the second survey. ...................................................................................... 120

C.4 The B version of the answers document, used for people answer to the second survey. ...................................................................................... 121

C.5 The C version of the answers document, used for people answer to the second survey. ...................................................................................... 122

C.6 The D version of the answers document, used for people answer to the second survey. ...................................................................................... 123
1

Introduction

In this Chapter we describe the context of the identified problem on the identification of dominant colors (DCs) in pictures, we present the objectives to fulfill, we summarize the main results achieved from the evaluation of the algorithms and finally we outline the organization of this document.

1.1 Problem Description

The problem we identified and propose to work out is related to the way the dominant colors are identified by the current works reviewed in Chapter 2. In all of them the authors accomplish this identification by using an algorithm-centered approach, instead of considering the complex nature in which people perceive the DCs. All the revised works fail by not considering the human perception, not only in the final evaluation of their procedures, but also in the understanding on how the DCs are seen by people. In general these solutions are only concerned with some system-centered measure, usually the well known Precision and Recall ratios, and do not consider the user. Besides, the majority of these works consider a large number of DCs (> 12) in pictures, which common users cannot distinguish.
In summary, after reviewing the related works in this area, we were able to define a set of good practices for the resolution of this problem. These practices lead us to the definition of a set of requirements for our solution.

1.2 Objectives

The main goals of this work are to review, understand, and evolve the processes of identification of the dominant colors in pictures. By analyzing the works of other authors we realized the lack of the human perception in the development of their solutions for DCs identification. Here we propose to develop a solution where the DCs of pictures are obtained from the point of view of real users, instead of a sightless automatic extraction from the point of view of a system. As we said before, this automatic and “blind” extraction is just what have been done by the several authors we reviewed.

For such objective we first asked users how they saw DCs in pictures, then we took this into account in the development of our solution, and finally we used the classifications from users to evaluate the algorithms we developed. Instead of only one algorithm, we developed six algorithms for the identification of the DCs with the aim of match the perception of users. We also compared the several algorithms developed. Our final objective is having an algorithm able to identify the real dominant colors that people see when looking to a picture.

1.3 Developed Solution

Our developed solution is built around the following principles: fuzzy segmentation of the HSV color space, a palette of 12 dominant colors, the counting of the dominant colors using the fuzzy theory, and the approaches on the segmentation (division) of the image space. By combining all these basic choices we believe the DCs of an image are well identified, according to the users point of view. From the evaluation with real users, results show that the DCs are identified from a human perspective.

From an architectural point of view our solution consists of two main modules: in the first the input image is re-sized, converted from RGB to HSV color space, and filtered; and in the second module the 12 DCs are identified. These 12 DCs dominances form the output of this second and last module. This second module in fact implement six algorithms, all (but one) based on our fuzzy segmentation approach. The exception is the Classic Histogram without any
fuzzy approach. The five remaining algorithms are based on the proposed Fuzzy Histogram, but with deviations on the way that the pixels are considered, by setting different weights on them. These weights are assigned through the different masks to which the input image is subjected, or the way the image space is considered. These masks go from an Ellipsoid based mask, where more importance is given to the center of the image, to a Saliency Map Based Mask, where more importance is given to those regions prioritized by a selective attention mechanism. Another algorithm tries a different approach, segmenting the image space into 3x3 sub-images and applying the proposed Fuzzy Histogram on each sub-image. The fifth algorithm identifies regions based on the (K-Means) clustering of the colored pixels. Finally the sixth algorithm, called Classic Histogram, uses a similar approach to the Fuzzy Histogram algorithm, except that it does not use the membership grades in the segmentation process.

In short, the second module of the developed solution is responsible for the identification of the DCs of the input image and therefore it outputs its DCs, in the form of a set of normalized 12 DCs dominances, which can be seen as a 12 bins histogram.

1.4 Contributions and Achieved Results

Our main contribution in this work is directly tied to the different way we conducted the development of the solution. Our solution for DCs identification was guided by the way people perceive the DCs. We started by asking them how they saw the DCs in pictures, then we developed the algorithms, and finally we asked them again if the identified DCs were correctly assigned to the correspondent pictures. These first tests with common users give our work a solid base, and from that base, and also including the decisions we selected from the reviewed works, we then developed the algorithms for the DCs identification.

To achieve this we use a palette of 12 colors suggested by Ware [Ware 04], and we segment the HSV color space using several fuzzy membership functions, one for each color, except for black, gray and white. The scheme we followed in this segmentation, though more or less resembled to the work by Chamorro-Martínez et al. [CM07], has also a novelty, as we included three crisp zones (for the black, gray and white colors). We also found in the first survey that most users identify only 3 DCs per picture, which is very far from what other authors use on their works.

In the end, the best algorithms, from the six implemented, were the Fuzzy Histogram and the 3x3 Segmentation. The first is the major algorithm and the base for the others, including
the later one, the 3x3 Segmentation. The main difference between them is that the later divides
the image into 9 sub-images and then applies the algorithm of the Fuzzy Histogram on each
sub-image. Experimental results show that these solutions are the best for the identification of
DCs in common pictures.

The proposed architecture can be easily extended to be included in an image retrieval ap-
lication.

Also given the amount of high-definition data that we gather and hold today in our devices,
new ways of data retrieval are welcome in the near future. These data can be either pictures
or video. In this work we only deal with pictures but some easy modifications can be made to
model our solution to a video based application.

Also the fuzzy approach in the segmentation of the color space can be extended to include
linguistic variables (natural language expressions) for a content-based retrieval system. By using
fuzzy logic we can use natural language terms in the colors extracted from the images. And this
allows the queries to be made in natural language, bridging the known “semantic gap” problem.

1.5 Organization of the Document

The solution we propose is described and justified in this work. In this document we start by
reviewing the related works in Chapter 2. From the analyzed works, we intend to study the
solutions, bearing in mind the advantages and disadvantages of each, and show the identified
problems. In Chapter 3 we present the first survey where we asked users the dominant colors
in pictures. With this survey we validated the palette of colors used, and we found the more
frequent number of DCs people identify. In Chapter 4 we detail the development of the algo-
rithms for the DCs identification. We present the modules of the architecture and we justify
our choices. In Chapter 5 we choose a set of four algorithms, among the six we developed, and
submitted them to evaluation by users. This was accomplished with an experimental evaluation
where we present to each user the DCs of pictures and ask them to rate those DCs. Finally, in
Chapter 6, we present the conclusions and the future work identified.
Related Work

In this section we will detail the known major works in the field of dominant colors extraction and retrieval. We organized them within four categories: the first is dedicated to histogram based methods; the second to segmentation based methods; the third to correlogram based methods; and the last to statistical or cross-related based methods, from now on known as “hybrid/statistical”, which might use one of the other methods with a more impelling statistical approach, or even methods from other computer science domains, such as Ant Colony Algorithms or Particle Swarm Optimization techniques.

2.1 Histogram Based Methods

By definition, an histogram of a colored image is the number of pixels from a given color of it. For a space of N colors, the histogram is an N-dimensional vector \((h_1, ..., h_N)\), where each element \(h_i\) is the portion of pixels of color \(i\) in the image.

For the distance function between two histograms several alternatives exist. In [Swain 91] authors proposed the \(L_1\)-norm, which basically defines that the distance between two points is the sum of the absolute differences of their coordinates. It is a simple and fast calculus but the
distance between two images that have similar (but not exactly equal!) colors is large, leading to errors, known as false negatives. A more frequently used distance function is the $L_2$-norm [Niblack 93], also known as Euclidean distance, and it corresponds to the length between two points using the Pythagorean formula. If the color space is selected appropriately the use of this metric reduces the number of false negatives inherent to $L_1$-norm. However this metric is still noise-sensitive. There is a representation less sensitive to noise, $L_{(inf)}$-norm [Stricker 95], which is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension. This is also known as Chebyshev distance. But experiments [Vassilieva 09] showed that the use of this metric yields insignificant gain compared to previous and simpler ones.

Human eyes are unable to see individual pixels or to perceive large amount of color levels. In [Swain 91] authors proposed the first color histogram, which solves this sensitivity problem. In their work, color histograms are extracted and an histogram intersection method is used for comparing two images. In general, one major disadvantage in terms of histogram’s practical application is the amount of data needed for representation. Frequently different research groups proposed more compact representations. The first step in these methods is transformation from RGB space into a more perceptually uniform color space. The image is then quantized and represented only through a smaller subset of colors. However, these histogram based algorithms still do not contain information about the spatial relationships of image pixels.

Another major performance drawback with the histogram based approach is concerned with the use of static quantization, where the color palette boundaries are determined empirically or via some heuristics - yet nothing based on human color perception rules. Another problem is the computational deficiencies with these methods’ practical application, due to the hundreds (or even thousands) of redundant bins created for each image.

In the remainder of this section we describe in detail some of the most important works for dominant color extraction from images, based on histograms, by the chronological order they were published. In the first presented paper Manjunath et al. presents the MPEG-7 color descriptors, which in some sense constitutes the basis to the following histogram based works. Finally the last two papers develop different clustering techniques, by using an unsupervised clustering algorithm and a fuzzy inference system, respectively.
2.1.1 MPEG-7 color descriptors overview

In [Manjunath 01] authors present an overview of color and texture descriptors that have been approved for the Final Committee Draft of the MPEG-7 standard [(FCD)” 01]. The four color descriptors are: Color Structure Descriptor (CSD), Scalable Color Descriptor (SCD), Color Layout Descriptor (CLD), and finally the Dominant Color Descriptor (DCD). The first two are strictly histogram-based, the CLD captures the distribution of colors, and the DCD the dominant colors. Back in 2001, the MPEG (Moving Picture Experts Group) committee realized the generic histograms had too many user independent variables, such as the choice of the color space, choice of quantization and merging of histogram bins, so they decided to set four description schemes (also known as layouts) for the coding of the descriptors, each one with its own purpose. The most appropriate to dominant color extraction is the DCD, even though some works among the others three exist. Nowadays the DCD still have a word to say in the context of CBIR systems. Its main weakness is related to its similarity measure, based on quadratic histogram distance. In detail it is unable to control the upper bound of the distance between two descriptors and its similarity coefficients are unable to use color distance to fine-tune the distance between two images. In its blessing it is very compact since there is no redundant information for non-existed colors, and similar colors are grouped into a palette color.

We will now detail each one of the 4 color descriptors from MPEG-7, with particular emphasis on the dominant color descriptor (DCD). The CSD (Color Structure Descriptor) only aims at identifying colors using a small structuring window and a refined color space, called HMMD (Hue Min-Max Differences), which is a normalized HSV color space. To this aim, an $8 \times 8$ sized structuring block scans the image in a sliding window approach. With each shift of the structuring element the number of times a particular color is contained on it is counted, and a color histogram is constructed this way. The SCD (Scalable Color Descriptor) provides less flexibility than the previous descriptor, by limiting the color space to HSV and by using a fixed quantization of 256 bins ($16 \times 4 \times 4$). It uses the Harr transform encoding. It is called scalable because the precision of histogram bins can vary from 16 to 1000 bits per bin, for different requirements. The CLD (Color Layout Descriptor) tries to capture the spatial layout of colors, by dividing each picture in a $8 \times 8$ grid and then calculating the average values of each sub-image. Then a Discrete Cosine Transform (DCT) is performed on each block, and the resulting coefficients are encoded into the descriptor. Similarity between two CLDs can be measured by
the root of squared differences of each matched coefficients. Finally, the DCD (Dominant Color Descriptor) goes a bit further, by extracting the distribution of the dominant colors in images. These are extracted by using clustering and color quantization. It can use any of the color spaces supported by MPEG-7, such as RGB, HSV, YCbCr and the HMMD. It works as follows: colors are clustered into a small number of representative colors and the extraction is based on the Generalized Lloyd Algorithm (GLA). Its feature descriptor consists of the dominant colors, their percentages, variances and spatial coherency of the dominant colors. A similarity measure based on Euclidean distance is also defined. The standard defines a maximum of 8 dominant colors, even though Manjunath et al. believe that “3 or 4 colors provide a good characterization for an image”.

The MPEG-7 Draft standard set off in the beginning of 2001, when the worries about storage and scarce processing power were very relevant. Nowadays, and of course depending on the application we are dealing, this concerns are less important. So their storage requirements led algorithm efficiency and real-time performance to second place, in particular with the DCD (Dominant Color Descriptor). For example, the fact that some parameters like weights in DCD should not be constants but rather dynamic for each set of images and for each particular application. Thus the color descriptors are far from tuned for a particular purpose but rather generic for all kinds of images, especially those who work on RGB color space. Authors admit that the several descriptors reported are mainly suited for natural images. For synthetic images or for very specialized domains such as bio-medical imagery, for instance, they admit refinements on the existing descriptors may be needed. The spatial coherency (or homogeneity), a unique feature in DCD descriptor, helps differentiate between large color blobs versus color blobs that are spread all over the image.

2.1.2 Histogram calculus with structured bins

Maybe taking the previously MPEG-7 based papers as inspiration, in [Wong 07] authors combine the dominant color features of Dominant Color Descriptor (DCD) and the spatial color distribution structure of MPEG-7 Color Structure Descriptor (CSD) to design a new color descriptor, called Dominant Color Structure Descriptor (DCSD).

The number of quantizing colors and the descriptor size are limited by the maximum number of clusters allowed in Generalized Lloyd Algorithm (GLA) splitting. To maintain the minimum distortion achieved by GLA, no agglomerative clustering is applied to the resulted clusters as in
DCD. As a result, there is no restricted minimum distance between quantized colors.

In this method color structure bins are formed as in MPEG-7 CSD, i.e., by accumulation using an $8 \times 8$ structuring window. The window scans through every pixel position of the image. A set of color bins is maintained for the scanning. At each pixel position, the existing colors in the region enclosed by the structuring element are recorded by incrementing by one the corresponding color bin. An histogram will be formed with the bin values counting the times that the structuring element enclosed the scanned colors. The histogram will then be normalized by the number of pixel positions that the structuring element scanned. It is somewhat different from conventional histograms because the histogram sum is being normalized. This leads to a significant difference in the interpretation of the bins. Each color structure histogram bin in DCSD represents the area covered by the color in the image, while in the conventional histogram each bin represents the number of detected pixels of each color.

The proposed DCSD descriptor in [Wong 07], besides the color values and the fact that it is almost like the MPEG-7 DCD, is defined with three parameters: the number of dominant colors, the structure values of dominant colors and an optional spatial coherency parameter. The major difference between DCSD and DCD is that, in the proposed DCSD, there are color structure bins instead of the color percentage of the DCD. A new similarity measure, which uses color matching based on the color distance between colors, is developed for DCSD. This color distance is measured by the Euclidean distance between the matched DCSD colors in CIELuv color space. The color of each pixel is quantized to the closest color in the dominant colors set if the color distance between the original pixel and the closest dominant color is smaller than an empirical threshold in the CIELuv color space. Otherwise, the pixel is ignored.

All in all the main difference for the known original MPEG-7 DCD is only that this one has the dominant color percentages within and the proposed one has structure bins, which represent the area covered by each color, for achieving the same purpose. In terms of the size of bytes occupied per descriptor, the proposed DSCD method stays in the middle of the table, where it is compared with Color Layout Descriptor (CLD), and several Scalable Color Descriptor (SCD) configurations. In terms of performance, using the Averaged Normalized Modified Retrieval Rate (ANMRR), the proposed method equals SCD, when considering an ultra-compact configuration, with only 5 colors. But CLD has the smallest descriptor size, with only 8 bytes, against 14 of the DCSD 5. When using 8 colors then the proposed DCSD and the CSD 32 both achieve the best ANMRR rate, but the former has a smaller descriptor. We can conclude this histogram based
descriptor almost achieves the compactness of the MPEG-7 DCD (this one is even smaller), and has a good retrieval rate, up to the level of MPEG-7 CSD, although this latter has a descriptor 30% bigger.

2.1.3 Another dominant color extraction based on MPEG-7

In order to extract the dominant colors from an image, in [Yang 08] Yang et al. use the modified Generalized Lloyd Algorithm (GLA) with clusters merging, to obtain a small number of representative colors and their percentages. In this work the dominant colors are extracted as an agglomerative result of adjacent partitions of the color clusters. They propose a scheme based on a coarse division of the RGB color space. The initial clusters correspond to the centroids of the most representative color bins, using the average of the color distribution.

Authors believe that by using a splitting algorithm, as K-Means clustering [Kanungo 02], is a possible way to overcome the problem of the initial number of clusters of the GLA based approaches. Since the K-Means algorithm starts with arbitrary random centroids, they do not need initial conditions. However, both algorithms may converge to local optimum and usually require high computational complexity.

According to their own experiments, the selection of a color space is not a critical issue for DCD extraction, they say. Therefore, for simplicity they use RGB color space. This RGB color space is uniformly divided into 8 coarse partitions/ blocks (known as “colorBins” in MPEG-7). If there are several colors located on the same partitioned block, they are assumed to be similar. After the coarse partition step, the centroid of each partition/block is selected as its quantized color by averaging the color distribution. After the average values are obtained, each quantized color can be determined. Then they calculate the mutual distance of two adjacent quantized colors, and then merge similar “colorBins” using a weighted average agglomerative function. The merge process is iterated until a minimum Euclidean distance between the adjacent color cluster centers is larger than some empirical threshold. As result the final number of dominant colors is 4-5 on average. Authors call the proposed method the Linear Block Algorithm (LBA). They defend that their similarity measure is consistent with human perception and is defined as the difference of the percentages of two colors, using the intersection (minimum) between the two colors.

In the experimental results presented, comparing to MPEG-7 and a classic palette histogram similarity measure, the proposed LBA achieves an ARR (average retrieval rate) improvement
rates of 4.2% and 6.6%; and ANMRR (average normalized modified retrieval rank) improvement rates of 5.4% and 4.8%, respectively. So the experimental results indicate that the proposed technique improves retrieval performance in terms of both ARR and ANMRR.

Even though authors in [Yang 08] do not have results exclusively for the dominant color extraction part we can accept that the proposed algorithm is effective, although we cannot be sure if it is capable of being used in another context. In the proposed method it is clear that the use of the RGB space instead of one more close to human perception, as the HSV/HSI, is a drawback because the RGB space is not linear. Dividing the RGB space in only 8 equal regions is not justified and is far from the way humans judge colors in pictures. And the crisp boundaries could be avoided in these partitions. Nevertheless there is a promising achievement with this approach: the number of dominant colors obtained is not very close dependent on the input parameters, as they obtain, on average, 4-5 dominant colors.

2.1.4 Unsupervised clustering algorithm for histogram estimation

In the same area of MPEG-7 based methods, the article [Min 09] tries to innovate in the field of relevance feedback technique by applying a fuzzy support vector machine for that purpose. A new color space based on the known HSV color space is used, because in authors opinion the original HSV components “are not equally distributed and not suitable for direct build of an histogram”. So they create this new Modified HSV (MHSV) color space with the following transforms from the classic HSV color space: $X_{MHSV} = Scos(2\pi H)$, $Y_{MHSV} = Ssin(2\pi H)$, $Z_{MHSV} = V$.

Next they apply an unsupervised clustering algorithm known as Graph-Theoretic Clustering (GTC) [Koontz 76], adapted to run on 3D histograms, to obtain the clusters. After building one histogram for each of the three components of the proposed MHSV color space and applying the GTC algorithm to find the cluster centers they obtain the dominant colors: the centroid of each cluster corresponds to a dominant color. They must previously set the number of dominant colors as an input in their method. Authors suggest as a rule of thumb to use 8 dominant colors as in other MPEG-7 based approaches. As the MPEG-7 DCD, their dominant color descriptor (DCD) has the number of dominant colors, the percentage of each dominant color, the spatial coherency value and the variance of each dominant color. In the similarity matching step they tried two methods: the classic Euclidean distance and the optimum distance computed with dynamic programming. But, as authors quickly regard, the latter one optimum distance is too
slow compared with the Euclidean distance measure, so in the end they just use the Euclidean one.

The presented results combined together color, texture and shape features. Their method has the same retrieval accuracy as a simpler HSV color histogram, but with a smaller descriptor. Authors also show that their new proposed DCD has better performance than the original DCD in MPEG-7. But against it there is one major problem: results reveal that several iterations imply high computation requirements, higher for the highest number of dominant colors chosen as input.

2.1.5 Fuzzy and neuro-fuzzy logic approaches

Completely different from the previous solutions, in the work [Bhoyar 09] Bhoyar et al. propose two systems for color histogram computation, both with aim on fuzzy color semantics: a fuzzy classifier and a neural-fuzzy classifier.

The color approach is based on the research by Chang et al. that shows that the main colors that can be named by all cultures are limited to 11 [Chang 00]. In addition to black and white, the discernible colors considered are red, yellow, green, blue, brown, purple, pink, orange and gray. In this previously mentioned work, Chang et al. use the JNS acronym (Just Not the Same) for those basic colors, which came from the fact that for two totally different colors their similarity grade is zero, so they are considered totally different, i.e., “Just Not the Same”. In the present reviewed work [Bhoyar 09], the 11 JNS color histogram bins are computed for each image and used as the color feature descriptor (or “image signature”, as they named it).

Bhoyar et al. believe that the uncertainty and vagueness present in image analysis suggest fuzzy logic as a natural paradigm. Here, fuzzy if-then rules are used to decide the color of a pixel with a given R, G and B values. The Mamdani’s fuzzy inference system [Mamdani 99] is used for finding the color confidence of each pixel. It has three inputs (R, G, B) with 4 membership functions generating 64 rules and one output with 11 membership functions (for JNS colors). These 4 membership functions define the linguistic variables “dark”, “somewhat dark”, “light” and “bright”, interpreted respectively as intensities “around 20”, “around 70”, “around 150”, and “around 250”. Each JNS bin is supported by up to 6 fuzzy rules, depending upon rough human perceptions of the respective color. For example the black, white and gray bins have a single rule each and the purple bin requires 6 rules to represent its variations. For the queries 5 membership functions are defined, closely related to 5 linguistic variables, in the
form of “hardly”, “somewhat”, “more or less” and “mostly” and “totally”.

Finally, in the neural network section, they do not stress it too much, maybe because the results are not quite satisfactory, as we will see in the next paragraph. The architecture of the three-layer feed forward neural network used for color classification is the following: 3 neurons in input layer, 45 neurons in hidden layer and 11 neurons in output layer, one for each of the JNS colors. The network is trained by using the Error Back Propagation Training algorithm.

The four membership functions for “dark”, “somewhat dark”, “light” and “bright”, are interpreted respectively as intensities, which is quite a non-sense, since they used the RGB color model, where intensities are difficult to set. Maybe in the background they transformed this RGB color model to a more perceptually uniform one. The presented results by the end of the paper reveal an irony because they are as fuzzy as the main subject of the paper: authors only present the top 6 results for four queries, in the form of images only. They should contemplate some common measure for an easy comparative against other approaches. They say it is possible to have joint queries like “mostly green AND somewhat red”, using standard intersection and/or union operators but they do not stress this fact, and again they do not present any specific results but only one single query. Finally, authors admit that the fuzzy approach has better performance and better results than the neural approach. The fuzzy is faster in the histogram calculus (60% less than the neural) and the system is more easily extended for more than the 11 JNS colors considered. Comparing the fuzzy system with the neural approach, the only drawback from the former is as follows: the time for computing the color descriptors increases as the number of rules defining perceptual colors increases, although the precision of the system also increases. The neural approach, as already mentioned, is slower, it requires using a calibration image for each one of the 11 colors and the used number of hidden layers needs a lot of experimentation, as the authors assume.

2.2 Segmentation Based Methods

As it was evidenced previously with histogram methods, the strict use of global color properties, like dominant colors and their coverage areas or distributions, alone are not enough for characterizing and describing the real color composition of an image since, by definition, they all lack information of spatial relationship among those colors. In other words, describing “what” and “how much” color is used will not be sufficient without specifying “where” and “how” the
perceivable colors (or DCs) are.

As stated in the previous section, it was clear that histogram approaches have several disadvantages. First, although they are simple to compute, they do not match human perception very well. Second, they need a huge amount of data for representation, unless a more compact representation is used, for instance with quantization.

There are several approaches to address such drawbacks. Segmentation-based methods may be an alternative. However, in most cases, they might not be a reliable nor robust solution simply because of the automatic or “blind” segmentation in the algorithms. Usually the practical applications are applied only on a very small universe of images and only work in that particular universe.

Some solutions use the local positions of image regions, also known as blocks or sub-images, for characterizing the spatial distributions. For instance, in one of the earlier works [Gong 96] authors divided the image into 9 equal sub-images and represented each of them by a color histogram. In a similar manner, Stricker et al. split the image into 5 regions: an oval central region and four corners [Stricker 96]. They tried to combine color similarity from each region but giving more weight to the central region. A similar approach is proposed in [Valova 04] where authors split the image into $16 \times 16$ blocks and each block is represented by one unique dominant color. Due to the fixed partitioning, such methods become strictly domain dependent solutions.

In [Ooi 98] authors enhanced the idea of using a statistically derived quad-tree decomposition to obtain homogeneous blocks but again comparing the matching blocks (in the same position) to obtain color descriptors similarity. Such a scheme is neither rotation nor translation invariant.

In [Pass 96] authors presented the Color Coherence Vector (CCV), which partitions the histogram bins based on the spatial coherence of the pixels. A given pixel is “coherent” if its color is similar to a colored-region, and “incoherent” otherwise. Another variation of this approach is characterizing adjacent color pairs, i.e. color boundaries. Nagasaka et al. in [Nagasaka 92] developed a color matching technique to model color boundaries. Thus, two images are expected to be similar if they have similar sets of color pairs.

Next we will cover in detail some of the most important methods that use some kind of segmentation to better identify dominant colors in images. Typically these methods impose some kind of spatial localization of colors and sometimes are used in combination with other features, such as texture or shape.
2.2.1 Spatial localization of color regions

In [Smith 95] authors propose a method to automatically extract the color content of isolated regions in images and build an index to retrieve those regions from a large collection of images. The spatial localization of the regions allows for queries on a database to include spatial positions and relationships between color regions, that is, compositions of several regions.

The method is simple and is as follows: (1) conversion from RGB to HVS color space, (2) uniform quantization of this HVS space, (3) application of a color median filter, (4) conversion back to indexed RGB space. In the end, the image has dominant color regions emphasized, authors ensure.

For color extraction Smith et al. use the HSV color space because “it is more perceptually uniform and the transformations from RGB are non-linear but easily invertible”. Quantization of the HVS space is done by the typical division in several color bins. The authors consider that an uniform quantization of the Hue component at 20 degrees sufficiently separates different tones of Red, Green and Blue, so they achieve 18 different Hues (because $\frac{360°}{20°} = 18$). Saturation and Value are simply quantized to 3 levels each. In total, after quantization, they obtain 18 Hues, 3 Saturations and 3 Values, plus 4 levels of grays, which it sums up to 166 different possible colors.

To identify color regions, the images are down-sampled to approximately 196 × 196 pixels, such that the original aspect ratio is preserved. They believe this reduces the number of colors to less than 50. Even after the transformation it is still premature to isolate color regions because small noises interfere. They clean most of this insignificant details by applying a 5 × 5 median filter on each of the HSV channels. According to the authors, this non-linear median filtering in HSV space does not introduce false hues. The color image is then converted back to an indexed RGB space, just for displaying proposes.

Having the dominant colors already exposed, the final step involves the extraction of the color regions from the images. The regions are classified based primarily on region size (area), but also on some minimum contribution of each color inside the selected region, among other classification attributes.

It is not clear why the authors sub-sample every image to 196 × 196 pixels, which we can generally consider too small. This sub-sampling leads to a deterioration in images with a lot of details to get missing and can also distort the results. Manly the area of colors is used to
classify each colored region, but they ignored any heuristic, even the naif “Rule of 3rds”. It is not clear why they chose a $5 \times 5$ window element for the color median filter. Authors do not justify any of these constraints.

### 2.2.2 Color regions using Color Coherence Vector

In the paper [Pass 96] Pass et al. describe a method which imposes additional constraints on histogram based matching. In their so-called “histogram refinements” the pixels for a given bin are split into classes based upon some local property: color, for instance. Within a given bucket, only pixels with the same color are compared. Concerned with image retrieval they try two different approaches: centering refinement, and color coherence vector (CCV). In the first one, authors consider the 75% of pixels in center of pictures, but they fail by not clarifying exactly how they do it. With the second one, pixels are partitioned based upon their spatial coherence. This particular classification defines that a coherent pixel is part of some sizable contiguous region. They achieve this pixel grouping by measuring connected components with 4-connected neighbors. The rationale for this is related with the idea of “scattered” vs. “compact” pixels of some color in pictures. It is similar to what have been done, a few years later, by the MPEG-7 committee in their dominant color descriptor (DCD), which has one parameter named “spatial coherency”, and it helps differentiate between large color blobs versus scattered color blobs which are spread all over the image.

The main objective with this histogram refinement is simply to create a finer distinction than classic color histograms, because frequently two different images may have similar histograms: it is sufficient that the number of some different colored pixels is equal. The two presented methods fit to a CBIR system, and in particular for dominant color extraction, as both analyze the spatial localization of colors in images. Both used techniques, i.e., the centering refinement and CCV achieve good results, in comparison with classic histograms, having the former a slightly better ranking than the second one. When CCV produce worse results it is always due to changes in the overall image brightness, because CCV use discretized color buckets for segmentation, and they are “more sensitive to changes in image brightness than color histograms”, authors assure. We believe that this difficulty could be surpassed by using a better color space than the RGB. We must be aware that authors fail by not specifying how do they figure the 75% image center: circular, oval or lozenge? In terms of computational cost authors only made evident that CCV is approximately 7.5% slower than color histograms, for the same database. We guess both
approaches could be coupled together and adapted to search for a few dominant colors within some centered weighted ratio in pictures, but authors do not go further this more ambitious concept.

2.2.3 Dominant color regions using the Sub-Block technique

In [Ravishankar 99] authors start by admitting color alone is not enough to characterize an image, so they try the spatial localizations of the identified dominant color regions in images. The strong motivation for using color to perform selection comes from the fact that it provides region information and that, “when specified appropriately, it can be relatively insensitive to variations in normal illumination conditions and appearance/viewpoint of objects”. Dominant color region in an image can be represented as a “connected fragment of homogeneous color pixels which is perceived by human vision” [Sm92]. The technique of the authors to index images is based on this concept of dominant color regions. The location of the regions is determined using the self-named Sub-block technique. This technique works by finding the distance from the region’s Minimum-Bounding Rectangle (MBR) to the coordinates of the center of the location map.

The entire RGB color space is discretized using a small set of 25 color categories, from the original 256 color palette. The reason to use this set is just that way “it gives a coarser description of the color of a region”, they believe. Their method segments each image into regions according to the perceived color. This is a segmentation procedure that involves mapping all pixels to some of the 25 categories in color space, and at the same time by grouping pixels belonging to the same category. Briefly, for each pixel, it searches for the nearest color in the lookup-table of 25 categories, by calculating the minimum Euclidean distance. Then they mark the region of every calculated pixel using the 8-connected neighboring region growing method [Adams 94], where a bounding rectangle is (conceptually) drawn for each selected dominant color region; the area of boundary rectangle is used in the calculus of the normalized area of the dominant color region. They apply this algorithm to find up to 3 dominant colors, i.e., the 3 regions with the biggest area.

Hardly the table of 25 hard-coded colors is an optimal choice. It fits the target images they test but it should not be static but dynamic, since it is some way dependent on the palette of colors of the considered image. The fact that they find up to 3 dominant colors in each image is not very far from what is obtained in some other papers, where this number may be 4 or
even 5, but we are not convinced about this result in the paper. Some of the images have not matched properly because of the color distributions. From the presented results we see that the percentage of color matching is less with complex pictures (of flowers, for instance) than with plain images (flag pictures, for instance). This demonstrates that this technique is more suitable for images with fewer variations in composition than real world pictures with variable subjects.

2.2.4 Segmentation into homogeneous colored and textured regions

In [Liu 04] Liu et al. want to develop a region-based image retrieval system with high-level concepts obtained from colored region features. This paper includes the initial experimental results of the authors using semantic color names. Firstly, each image is segmented into homogeneous regions. Then, for each region, a dominant color is defined. The dominant color is then converted to a semantic color name (for example, “grass green” or “sky blue”). This way, they believe the “semantic gap” is reduced. As Datta et al. cite in [Datta 08] “the semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data has for a user in a given situation”.

Their system has three components: image segmentation, color naming and query processing. Only the first two are useful in this research. For the image segmentation authors use the JSEG method [Deng 99] to segment images into homogeneous regions in color and texture. With this method authors obtain 5.84 regions per image, on average.

Liu et al. use a modified HSV color space. They define a perceptual color model in HSV space for each region and then convert each color to a semantic color name. Four different ways to define perceptually dominant colors are developed: 1) the average HSV value of a region as its dominant color; 2) the average RGB value of a region; 3) the average HSV value of all pixels in the bin with maximum size, from the color histogram of a region; and 4) the same as previous but selecting all bins with size no less than 68% of the maximum size bin.

Next, authors design a color naming model as follows. First, they define 8 base colors (red, orange, yellow, green, cyan, blue, purple and magenta) with the range of the Hue values as [345,8], [8,36], [36,80], [80,160], [160,188], [188,262], [262,315] and [315,345], respectively. Saturation and Value are quantized into 2 bins each. And there are special cases: when \( S = 0 \) and \( V = 1 \), it is “gray”, when \( S = 0 \) and \( V > 80 \), it is “white” and when \( V = 0 \) it is simply “black”. So, with this model they obtain \( 8 \times 2 \times 2 + 3 = 35 \) different possible colors.

The main advantage of this method is that it allows users to perform queries by keywords,
achieving and bridging the classic “semantic gap” in multimedia retrieval system. Liu et al. observed that in most cases the resulting four perceptual colors are very similar. From the presented results we see that this kind of system only works reasonably when the region of interest is recognized and the defined color names can describe it well. Nevertheless results show slightly better performance of the fourth developed method, i.e., when they use the average HSV value of those bins with size greater than 68% of the maximum sized bin. Besides, in some special cases, the first method, i.e., the average HSV, results in a color visually very different from the region original color. The crisp approach in the partitions of the color space is not sustained with any practical/empirical base. Their method for extraction of dominant colors is only tested on a small subset from Corel images, all of them with ground-truth data available, which may reveal a weakness, since results for images with varied content are very poor. Even only considering this subset in which they choose to test their method with, they compared it against a classic global color histogram system for image retrieval with the HSV color space uniformly quantized into 18, 4, 4 bins, respectively, instead of using their own 8, 2, 2 (+3) bins, respectively, and with their own segmentation of the HSV values range. So this comparative is not fair since it is difficult to know the real causes of the differences encountered on each of the results.

### 2.2.5 Classification of dominant colors through fuzzy linguistic expressions

Using an analogous approach as the previous work, the aim of Younes et al. in [Younes 06] is not to achieve a simple classification but to retrieve images according to their dominant color expressed through fuzzy linguistic expressions and to adapt the classification to the user sensibility, since, as authors defend, “color perception is subjective”. As opposed to other works in the field of CBIR, here authors do not use a query image but instead use a query from the user as in [Liu 04], expressed by a certain color tone. These are linguistic names for colors, in plain English.

Authors use the HLS (Hue, Lightness, Saturation) color space, and argue that the use of the hue component is enough to characterize a color, except when it is very pale or very sombre. Their strategy is the following: they limit themselves to 9 “fundamental” colors, which correspond to: “red”, “orange”, “yellow”, “green”, “cyan”, “blue”, “purple”, “magenta” and “pink”. This set is represented in Figure 2.1 and corresponds to the seven colors of [Roire 00] to which they have added “pink” and “cyan”.
Nevertheless, HLS is not a completely uniform color space. This means that a variation in the color scale is not directly equivalent to human perception, i.e., in practice our eyes do not perceive small variations of hue when color is green or blue, while they perceive them very well with orange/reddish, for example. It has to do with the way we perceive colors and the biologic characteristics of the human vision, but this goes off the subject of this report. So they choose to use HLS color space with the color definitions from the French site Pourpre.com, where the visible spectrum of light is divided in almost equal proportions. With these definitions established, trapezoidal and triangular functions are used to represent fuzzy membership sets. In particular, as can be seen in Figure 2.1, all fundamental colors are represented with triangular functions and only “blue” and “green” use the trapezoidal functions, because these two are wider in terms of the correspondent angular hue. As any other regular membership functions they varying from 0 till 1. For the other two dimensions of the color space, Lightness and Saturation, both are divided into three subintervals: the first subinterval corresponds to a low value, the second to an average value and the last to a strong value. This division gives for saturation S: “dull”, “moderately dull” and “saturated”; and for lightness L: “pallid”, “heavy” and “gloomy”. These two scales are then aggregated to give 9 qualifiers for colors defined by the following set: “somber”, “dark”, “deep”, “gray”, “medium”, “bright”, “pale”, “light” and “luminous”. Finally, for the achromatics, there are black, white, and 3 shades of gray: “dark”, “medium” and “light”. In summary: to every colored image can be assigned, from the basic fundamental ones till every refinement, a total of 96 color classes, defined by a fuzzy approach with the former linguistic variables.

In terms of processing, for each pixel of the image the method aims to determine the values taken by the various membership functions of the categories. For each category the value obtained corresponds to the ratio between the sum, on all the pixels of the image, of the membership functions values and the number of pixels, which gives a quantity between 0 and 1. This
quantity is the membership degree of an image to the given class. The fact that they used the color categories from the site “http://Pourpre.com” is a more empirical approach but fits human perception well, as results show. The full set of 9 basic colors (or “fundamental”, as they named it) is intuitive since it approximately corresponds to the rainbow colors. But there are some drawbacks: authors do not present any comparison against other similar works in the area of image retrieval; they argue that others frequently use an image query and compare not only colors but also textures and/or shapes. Conveniently or not, that is the reality, i.e., the great majority of studies evolve query images and multiple features over an image. Even though, they present good results in terms of recall and precision and, even though we stress again the fact that they do not have any comparison base, these results should not be considered perfect because of the following two main reasons. First, the proposed algorithm implementation sets more dominant colors than the used database originally has, so it finds more dominant colors. Second, the other problem is neighborhood related, since the perception of a color depends on the neighbor colors and they did not take this into account. There are three more small issues: 1) aberration pixels around the edges of some object in image, which they propose to solve with an edge detection method, namely the known “Canny algorithm”; 2) aberration pixels in uniform zones of images, due to digitalization/compression, which they did not solved successfully; and 3) the classification function, which is performed by an equi-weighted average, that is, each pixel has the same weight/importance, which leads to the neighboring problem previously mentioned. To solve this they could apply different weights to pixels, for example: isolated pixels shall get a weaker weight and aberrant pixels in object edges shall get a zero-weight.

2.2.6 Dominant colors segmentation using morphological clustering

In [Lézoray 09] Lézoray et al. introduce an unsupervised morphological clustering on 2D histograms, where the morphological classifier is the watershed operator. These 2D histograms are simply pair-wise associations of the RGB color space, namely Red-Green, Red-Blue and Green-Blue. After the independent clustering on each one of these 3 pair-wise histograms, region merging is performed on these same 3 histograms with the help of a MST (Minimum Spanning Tree) based algorithm. Then, already with only one merged map, an energy function of the two most similar adjacent regions in this map is iterated, until a minimum is found. This energy model is inspired by Markov Random Fields theory. In this process a smoothing parameter
is experimentally tuned by authors. They state that, in the end, this constitutes a trade-off between the fidelity and the simplicity of the segmentation.

So, basically the dominant colors correspond to the maximums (centroids) in a 2D histogram. With the erosion operation, clusters are progressively thinned until several minimum connected components appear. As authors show, this usually results in several class centers (centroids) without any prior assumption on their number. These centroids are then labeled to obtain the partitioned regions. Remark that this clustering process is applied to each one of the 3 pairwise histograms (RG, RB and GB) in the first part of the algorithm, and these 3 maps are combined altogether (super-positioned) to obtain the final segmentation of the image.

An interesting alternative to simple (1D) and 3D clustering methods relies on the use of three pair-wise projections and especially bivariate histograms (2D histograms) which use two color bands together (pair-wise associations), namely RG, RB and GB, in the RGB color space. The greatest advantage of this three 2D histogram approach is as follows: the search complexity is reduced, compared to the 1D and 3D clustering method. Another advantage to be considered is the fact that a 2D histogram is nothing more than a grey-level image. Therefore, classical grey-level image processing algorithms can be used to cluster them. Still, it is required to compute 3 clustering processes for each one of the 3 different pair-wise histograms and this can be computationally demanding. Also the meaning of each result is more challenging to understand. As for the distance metric, we guess if they use CIELab color space their distance measure performance between colors could improve. And as authors admit in the paper, their algorithm is competitive for the case of impulse noise but not Gaussian noise, which they assume it is not so frequent in their images. In terms of absolute results, it is disappointing to see that the JSEG technique [Deng 99] is better than any of the settings and tweaks they attempt with their own proposed method.

### 2.2.7 Dominant color selection by spatial and color domain analysis

In this paper [Kiranyaz 10] Kiranyaz et al. present a color descriptor which describes dominant colors both in spatial and color domains. For this purpose they developed two spatial color distribution descriptors: the proximity histograms, which compute the histogram of color distances but do not have spatial or directional info, as it is only a scalar distance; and the proximity grids, which represent the spatial occurrence of different dominant colors in a 2D grid, and so it has distance and directional info about colors. This combination of global and spatial properties
forms the final descriptor. Finally a penalty-trio model fuses all color properties in a similarity distance computation during retrieval.

In accordance with the Gestalt law\(^1\), authors adopt a global, top-down approach, in order to “model the whole color composition before its parts”. This way they believe they can avoid the problems of pixel-based approaches. The dominant colors are extracted together with their global properties and a quad-tree decomposition\(^2\) partitions the image, so they recursively sub-divide each cluster into four quadrants, so that, with the right parameters, they avoid to divide them until the pixel level.

For the dominant color descriptor they use an unknown specified clustering method, with the initial clusters determined by a weighted distortion measure. The clustering process is iterated until a maximum number of clusters (dominant colors) is achieved or a maximum allowed distortion criterion is met. So pixels with smaller weights have fewer clusters and are almost suppressed, depending on those stop criteria. Note that they consider 6 as the maximum number of dominant colors. The dominant color extraction method used is similar to what has been done in [Kenney 99], which is known as the Peer Group Filtering (PGF) method. This PGF method is characterized by a non-linear algorithm that applies a weighted average filter with a window across image, and only needs 4 configuration parameters: color similarity, minimum area, minimum distortion and maximum number of dominant colors. Next, in the proposed method follows the merging process. It is done with an agglomerative clustering to merge similar color clusters, and these are, in the end, the dominant colors. The merging criterion is the minimum area of the cluster: small clusters are annexed to others because they are considered too small (too many details in image) and as a consequence they are deleted.

This work [Kiranyaz 10] focus on a descriptor based both on spatial and dominant color distribution. Authors try to apply this to a particular content based application so the presented results may be biased by this fact. They obtain good results comparatively to the MPEG-7 dominant color descriptor (DCD) and classic correlogram, but this later hardly can be practically computed for a database with 20,000 images, because it is computationally heavy. Authors admit correlograms are better for smaller databases because they take the texture of the images more easily into account, and also because after all it analyses images at a pixel based level, via co-occurrence probabilities. In the end, besides the promising results, the proposed method is not

\(^1\) URL: http://www.gestalttheory.com/
\(^2\) A quad-tree is a tree data structure in which each internal node has exactly four children, namely four quadrants or regions.
appropriate for images with a lot of textures, since a more simplistic correlogram can handle more efficiently this kind of images.

2.3 Correlogram Based Methods

From what we already saw, global color properties, such as dominant colors and their coverage areas, alone are not enough for characterizing and totally describing the real color composition of an image, since they all lack the spatial relationships among colors. In other words, describing “what” and “how much” color is used sometimes is not sufficient without specifying “where” or “how” the dominant colors components are. Segmentation based methods, as we saw previously, may be a solution to such drawbacks. However, in most cases, they are not completely feasible since automatic segmentation is an ill-posed problem: it is not reliable and robust for practical applications with large and diverse image databases. A smarter move may be the use of a function that relates on each image the probability of finding one color pixel with the location of that pixel. This is the correlogram. In this section will see the most prominent correlogram based methods.

By definition, the color correlogram is a table, where the $k_{th}$ entry for the color histogram bin pair $(i, j)$ specifies the probability of finding a pixel of color bin $j$ at a distance $k$ from a pixel of color bin $i$ in the image. Since the feature vector size of correlogram is $O(m^2d)$, where $m$ is the number of quantized colors and $d$ is the distance between two pixels, a simplified version is often used, the so-called auto-correlogram. It only captures the spatial correlation between the same colors, and thus reduces the feature vector size to $O(md)$ bytes. This simplified and faster auto-correlogram was first proposed in [Huang 97]. The main advantage of the auto-correlogram compared to the original correlogram emerges when dealing with large image databases.

In another work, Li et al. [Li 08] proposed Markov Stationary Features (MSFs), which is seen as an extension of the auto-correlogram, and compared it with the auto-correlogram and other MSF based CBIR features, such as color histograms, CCV, texture and edge. Among all color descriptors, auto-correlogram (extended by MSF) performs only slightly better than the original auto-correlogram. Another extension is the Wavelet Correlograms proposed by Lee et al. in [Lee 08], which performs slightly better than the original correlogram and surpasses other color descriptors such as classic color histograms and the MPEG-7 Scalable Color Descriptor (SCD). Other authors in [Moghaddam 06] proposed a different approach, called Gabor Wavelet
Correlogram for image indexing and retrieval and further improved the retrieval performance.

Next we will see in detail three of the most prominent works that use correlogram as the base extraction technique of dominant colors. The first one is considered the original paper with a correlogram approach for that purpose, the second is basically an histogram based method with a co-occurrence matrix, and the last one describes an hybrid color and texture CBIR system with auto-correlogram for dominant color extraction.

2.3.1 Spatial correlation between pairs of colors

In [Huang 97], Huang et al. introduce a new color feature for image indexing and retrieval called the color correlogram. The highlights authors emphasize are: (i) it includes the spatial correlation of colors, (ii) it can be used to describe the global distribution of spatial correlation of colors; (iii) it is easy to compute, and (iv) the size of the feature vector is fairly small.

The color correlogram is neither an image partitioning method nor a histogram refinement method. Unlike purely local properties, such as pixel position, gradient direction, or purely global properties, such as color distribution, correlograms take into account the local color spatial correlation, as well as the global distribution of this spatial correlation. A variation on this, less computational demanding and known as the auto-correlogram, captures spatial correlation between identical colors only. Huang et al. in this paper only show results for their best method which uses precisely this auto-correlogram, with distances 1, 3, 5, and 7 pixels. The RGB color space is quantized into $4 \times 4 \times 4 = 64$ color bins. Then, they create four 64 color bins, one for each of the mentioned distances. The first 64 bins are the number of times each pixel of a given color has neighbors of the same color at distance 1. The next 64 bins are for distance 3, etc.

Authors also investigate a different relative distance measure to compare two feature vectors. The $L_1$ distance measure, used commonly to compare vectors, considers the absolute component-wise differences between vectors. The relative distance measure they use computes relative differences instead, and in most cases performs better than the absolute measure.

As results show, such a color feature turns out to be robust in tolerating large changes in appearance of the same scene caused by changes in viewing positions, changes in the background scene, partial occlusions or camera zoom that causes radical changes in shape.

But the size of the correlogram is $O(m^2d)$, where $m$ is the number of quantized colors and $d$ indicates the distance between two pixels; instead, auto-correlogram is only $O(md)$. In the
paper authors use $m$ equals to 64 colors. And as we can see by their results, if $d$ is large, say more than 10, we can have issues with performance, in the case performance is a valuable requirement. For this problem authors suggest parallelism in computation. But a simpler way was just considering the partial calculus of a correlogram, using a small distance $d$, and turning this a localized based method within an image. The implementations for (auto-)correlogram might even take advantage from the fact that the matrices usually are very sparse, and exploiting this with large cuts for speed improvement. In general the invariants presented by the authors in the algorithm seem ambiguous and should be cleared, at least by some experimental basis.

In terms of absolute results, the auto-correlogram performs better than any of the other compared methods in the work. These include: histogram, Color Coherence-Vector (CCV) from Pass [Pass 96] and an improved version of CCV. And we guess the correlogram does not perform even better because authors did not find a way to optimize the best value for $m$ and $d$. Also the RGB color space could be changed to a more naturally perceptually uniform one, CIELab or at least HSV for instance, which we think could bring a closer identification to human similarity perception.

### 2.3.2 Dominant color regions by a co-occurrence matrix

In paper [Hu 00] Hu and Mojsilovic perform extraction of dominant colors through the following steps. First, a color image is transformed from the RGB space into the CIELab color space. The set of all possible colors is then reduced to a subset defined by a color codebook. This codebook is independent of each particular image and is generated by first sampling the luminance axis into 20 levels and then quantizing the plan at each level using a hexagonal spiral Fibonacci lattice.

Apart from their representation, here Hu et al. extract visually dominant colors with the following method. They developed a windowed-based filter algorithm to identify the colors of speckle noise and remap them to the surrounding dominant color. They first segment each image into non-overlapping $20 \times 20$ windows and then proceed independently in each window. For each window, they compute what they called a $m \times m$ "neighborhood color histogram matrix", $H$, in fact a non-symmetric color co-occurrence matrix, where $m$ is the number of colors found in the window. Then, they define $H[i, j]$ as the number of times a pixel having color $j$ appears in the small neighborhood (of 3-5 pixels) of a pixel having color $i$. Each row $i$ in $H$ represents the color histogram in the collection of neighborhoods of pixels having color $i$. Based on this histogram
co-occurrence matrix, speckle colors are detected and remapped in the following manner. For each color $i$, the algorithm examine row $i$ and find the entry $H[i,j]$ that has the maximum value. If $j$ equals $i$, then $i$ is determined to be a dominant color, and no remapping is done. Otherwise, $i$ is determined to be a speckle color, occurring mostly in the neighborhood of color $j$, and then all pixels of color $i$ in the window are remapped to color $j$.

Finally, they developed a method for measuring the distance between two images in terms of color composition. Briefly, they have a mapping distance function, which measures the distance between two colors in the color codebook. They realize they have to minimize this function. This is an optimization problem, which they model as a minimum cost bipartite graph and they solve it using Gabow algorithm [Gabow 74].

The method by Hu and Mojsilovic [Hu 00] is based on the observation that human beings tend to ignore isolated spots of a different color that are randomly distributed among a dominant color, which is a solid, though empirical, principle. In terms of results they compared their method against four other known methods and obtained very satisfactory results. Nevertheless, we wonder what would be these results if they do not use only images with patterns, but instead real nature images, for example. The main drawback with their method is certainly the need of such heavy calculus for each image: for $N = 20$ as authors suggest, each image has $20 \times 20 = 400$ matrices $H$. In real-time this is barely feasible. As other less important drawbacks in the method we note: the non-symmetric characteristic of matrix $H$, which case it was symmetric it would simplify the processing weight. The fact that they partition each image in solid $20 \times 20$ sub-images and treat them independently is a bit disappointing, considering they use a correlation matrix in their own algorithm and so they should be aware of its benefits. It is impressive they accomplish pretty good results although their strict low-level pixel-wise approach. Still about their performance: the metric they developed for measuring distances between two colors for a retrieval purpose is also clever and matches well human perception of colors, as they emphasize with their successful ground-truth test with volunteers but, again, has a lot of complexity and certainly poor performance in real-time applications.

### 2.3.3 Dominant colors extracted from auto-correlogram based wavelet decomposition

In this paper, Chun et al. [Chun 08] propose a CBIR method which uses the color auto-correlograms of Hue (H) and Saturation (S) components of HSV color images. They combine
these auto-correlograms with the block difference of inverse probabilities (BDIP) and block variation of local correlation coefficients (BVLC) moments of the Value (V) component in the wavelet transform domain. Basically the color auto-correlogram is used for the color feature and the BDIP + BVLC for the texture feature.

In HSV color space, each one of the three components H, S and V is wavelet decomposed into a wavelet image. Authors sustain that the H and S components are more closely related to the chrominance information and the V component contains most of the texture information from images. Based on these arguments, they decide to create their composite feature descriptor. The proposed descriptor has the color auto-correlogram extracted from both the Hue and Saturation wavelet images, and with BDIP and BVLC moments extracted from the Value wavelet image. As already emphasized in the beginning of this Section, the main difference from the regular correlogram and the auto-correlogram is that the latter one captures the spatial correlation between identical colors only. The two-level wavelet decompositions of the Hue and Saturation components are then quantized in a non-uniform way with the known Generalized Lloyd Algorithm (GLA). Indeed, each LL band is in fact uniformly quantized and the other HL, LH and HH sub-bands are so non-uniformly quantized. The number of levels of quantization is found by minimizing the sub-band variance of the corresponding sub-band wavelet image. The color auto-correlogram is extracted only after the numbers of quantization levels for all sub-bands are decided. In the end, after the texture part is extracted (deliberately ignored here), the proposed system combines these two feature vectors for a complete retrieval method. The similarity measure they use, for query purposes, is the generalized Minkowski-form distance.

In terms of results they tested their system against a wide set of retrieval methods, namely color histogram, MPEG-7 SCD, MPEG-7 CSD, autocorrelogram, wavelet moments, Edge Histogram Descriptor (EHD), BDIP + BVLC moments, color histogram+wavelet moments, SCD + BDIP + BVLC and CSD + EHD. They also consider a vast databases set: Corel, VisTek, MPEG-7 common color dataset, and the corresponding versions of these three databases but with different resolution images (“multi-resolutions”, as they named it). As accuracy measures they are also overwhelming, with a precision/recall ratio and the well-known ANMRR (average normalized modified retrieval rank).

Authors show that the feature vector combination and the similarity measurement can also be progressively implemented to reduce the computational complexity. So they propose two approaches: a classic non-progressive and a progressive one, where the color and texture de-
scriptors are combined and the similarity measure is applied in run-time. Experimental results showed that the proposed retrieval method with progressive scheme gives 38 ms and with non-progressive scheme gives 44 ms, for the VisTex database. In terms of average precision there is only a small loss of 1.5% between the two approaches, which is not relevant. Therefore, we can see that the performances of both approaches are very similar.

Comparing their method against others, we can also find that the proposed method with progressive scheme needs about half the time for the number of additions and about 2.3 times the number of multiplications and about 20% less the number of comparisons of the CSD method, which authors say it is known to produce the most excellent retrieval performance among the MPEG-7 color descriptors. From other results they present in the paper, we can see that the effectiveness of combining the multi-resolution color auto-correlogram and BDIP-BVLC is notable, for each database they use, and for each method they compare it with. The advantage is more noted for multi-resolutions images, which is their “weapon of choice” target. In this case they stress this fact by comparing the proposed method against the classic auto-correlogram and against BDIP + BVLC, both implemented alone, and using different resolution images: the proposed method achieves more or less the same results for equal resolution images but it achieves a better rank for those different (multi-)resolution images. It is an advantage, indeed. Against all other considered methods it only finds a rightful opponent in the MPEG-7 CSD method, which shows even better ANMRR gain, but only with the MPEG-7 CCD database. Finally it is worth to mention the great performance of the EHD method, which is one of the MPEG-7 texture descriptors. It almost achieves the proposed method ANMRR level, but at a cost of a vector with a large dimension.

2.4 Hybrid/Statistical Based Methods

Although correlogram based techniques achieve good results they all suffer from the following common drawbacks. First, they all have high computational complexity, with massive memory/CPU/storage requirements, especially for huge databases. Second, they all have a pixel-based structure, which it indicates that those methods do not directly take into account the human color perception rules, simply because individual pixels do not mean much for the human eye [Kiranyaz 10].

Just to have an idea about the heavy-weight of the generality of correlogram algorithms: for
a database with 1000 images, each one with 1Mpel (Mega-pixel), using the lowest range setting for correlogram (10%), the computational cost is $O(10^{10})$. In order to apply some correlogram algorithms we must quantize the images a lot. And this, if not properly done, can reduce drastically the color information. So, the balance on the correlogram side is mostly favorable if we do not have real-time performance requirements. As we will see next, some authors try some iterative optimizations with a statistical approach to suppress these issues. It is imperative we must take the subsequent note: the following section is not rigorously about statistical methods on dominant color identification. The following researches use several techniques from other scientific areas that use, with more or less expends, a statistical approach, such as optimization with evolutionary algorithms.

2.4.1 Color indexing based on central moments of each color channel

In this early paper from 1995, [Stricker 95] Stricker and Orengo present two color indexing methods, the second one being a fresh alternative to classic histogram based methods. The first is just an histogram indexing, in which the index contains the complete color distributions of the images in the form of cumulative color histograms. To determine the similarity between two index entries (two distinct histograms) they apply the $L_{\text{inf}}$ metric. The second method is a completely different approach to color indexing. Instead of storing the complete color distributions, authors store only what they call their “major features”. They store only the first, second and third central moments of each color channel in the index. The first moment is the average (color), the second is the variance, and the third is the “skewness” of each color channel. For the distance function, a weighted sum of differences of the corresponding color moments is used. In the retrieval process only these three features are compared and it is significantly faster than a retrieval process based on the comparison of complete color distributions.

With a CBIR application in mind, the quantization of the colors in the histograms and in the cumulative histograms is not intrinsically related to the problem itself. This makes it hard to find an optimal quantization. The search for a method without parameters in the index creation process lead authors in this work to this approach: working with color distribution features, with a statistical modus. From probability theory, we know a probability distribution is uniquely characterized by its moments. So, they interpret the color distribution of an image as a probability distribution function, and they characterize that color distribution by its moments.

From their results we can see that the first method they use, i.e., the cumulative histogram
with a $L_{(inf)}$ metric, does not yield, on average, better match values than the application of the more traditional $L_1$- or $L_2$-metrics. The second technique, taking the color distribution as a probability approach, where they only store the three central moments on the index, achieves a real new advance to color indexing. Their results show that this second attempt outperforms their own first method and also classic histograms, with 256 or 32 bins. The index becomes smaller and the retrieval process is faster and produces better results than the mentioned tested methods. As the main drawback from this second technique we can only evidence the fact that they do not correlate the three color channels, but instead always treat them independently.

2.4.2 Ant colony clustering algorithm for dominant colors extraction

In [Huang 06] Huang et al. introduce an algorithm for extracting dominant colors in the CIELab color space. Their method is based on an Ant Colony Clustering (ACO) algorithm. The ACO-based technique models the behavior of ants collecting corpses, known as Ant Colony Basic Model [Deneubourg 91] and is used for extracting dominant colors as the foundation for image matching. The corresponding similarity metric is done by using an optimal matching algorithm, known in graph theory as Kuhn-Munkres [Lovasz 86].

Using the biology metaphor, the general idea for this ant colony algorithm is that, when an unloaded ant encounters food, say a corpse, it will pick it up with a probability that increases with the degree of isolation of the corpse. When an ant is carrying a corpse, it will drop it with a probability that increases with the number of corpses in the neighborhood. The picking and dropping operations are biased by the similarity and density of data items within the ants’ local neighborhood: unloaded ants are likely to pick up data items that are either isolated or surrounded by dissimilar ones; loaded ants tend to drop them in the neighborhood of similar ones. They define two probabilistic functions for these picking up and dropping actions. Then they compute a density measure for each ant. Each ant acts according to its current state and its corresponding probability. Finally several clustering centers are formed through the actions of the ants collective.

With their novel approach, authors believe that address the two known issues of classic GLA quantization schemes, i.e., avoiding the clustering to get into local optimality and the sensibility to initial clustering centers, which usually must be set a priori. Other advantage of their implementation is that their feature vector is indeed compact, with only 2 elements, i.e., the index of each dominant color and the corresponding area percentage of it. But it does
not include any spatial information, which can also difficult the similarity measure over images. Other drawbacks in their method are: i) the static number of dominant colors in each image, which they consider equal to 8, even though they do not give any theoretic support for this; and ii) the computational complexity of the Kuhn-Munkres algorithm, which takes $O(K^3)$, where $K$ is the number of dominant colors found in an image. But the biggest flaw may be concerned with the inner philosophy in every system where cooperative agents interact: the real-time poor performance issue.

2.4.3 Dominant colors by a dual segmentation using crisp and fuzzy approaches

The methodology proposed in [CM07] to extract dominant fuzzy colors from images consists of two stages: first, the dominant colors are extracted using a crisp approach; then, each color calculated in the previous stage is employed in order to obtain the set of dominant fuzzy colors. At the same time, authors also introduce a solution to store, index and retrieve dominant fuzzy colors.

Authors use the HSI color space. But they argue that HSI representation has two well-known problems: the non-representativity of the Hue when the Intensity or the Saturation are small, and the non-representativity of Saturation under low levels of Intensity. The solution authors find to solve these issues is to perform a partition of the color space based on the chromaticity degree of each point. They split the HSI space into 3 crisp regions: chromatic, semi-chromatic and achromatic, based on some self-defined thresholds. A color is: achromatic if it has small intensity (very dark); semi-chromatic if it has small saturation (grayish to white); and chromatic otherwise.

To extract dominant colors the authors of this paper perform a clustering technique using Batchelor and Wilkins algorithm [Jain 88], where the number of clusters is unknown a priori. For each one of the 3 zones previously considered (achromatic, semi-chromatic and chromatic) the clustering method is initialized with one cluster only and then an iterative split procedure is performed until the stopping criterion is met. This stopping criterion is based on the maximum distance to be achieved between points within each cluster. The centroid of each cluster is estimated as the mean value, and it corresponds to the dominant color if it. To avoid noisy or less relevant colors, the method discards those clusters with a number of points less than an empirical threshold.
In this work, Chamorro-Martínez et al. also have employed fuzzy spaces for hue, saturation, and intensity that are shown in Figure 2.2. The idea is to obtain the fuzzy subset of dominant fuzzy colors from the set of crisp dominant colors. Specifically, they have used as reference the Munsell color space [Hall 12], by dividing in 10, 7 and 9 intervals the hue, saturation and intensity, respectively. Each of these intervals is fuzzified using trapezoidal functions, which define the fuzzy set memberships. This way now a color in a fuzzy sense can be defined as: achromatic if $C_i = \text{“dark”}$; semi-chromatic if $C_i \neq \text{“dark”}$ and $C_s = \text{“very low saturated”}$; Chromatic if $C_i > \text{“dark”}$ and $C_s > \text{“very low saturated”}$. The degree of dominance associated to each fuzzy color $C$ is calculated as the minimum between the membership degree of dominant crisp color to $C$ and the dominant degree of that crisp color. If a fuzzy color $C$ is compatible with several dominant crisp colors, then a different degree of dominance is obtained for $C$, corresponding to each crisp color compatible with it. In this case, the maximum of these degrees will be selected as the final degree of dominance of $C$. In the end, they define the fuzzy subset of dominant colors as the union of the set of dominant crisp colors, for each of the 3 components of the color space, so that users can define queries containing dominant color based conditions. These are, in fact, fuzzy conditions based on fuzzy operators, such as “inclusion” and/or “equality”.

This paper by Chamorro-Martínez et al. shows how to define a new color space with a fuzzy approach and uses the Batchelor and Wilkins clustering algorithm [Jain 88] to extract dominant colors from pictures. In this clustering algorithm the number of clusters in unknown.
a priori. This way they avoid drifting techniques, they believe. Another advantage comes from
the inner philosophy of the use of fuzzy logic in queries: the linguistic terms helps humans to
better understand the selection/extraction of colors. Based on the presented results it is evident
that their algorithm is more proper to retrieve in query-by-example using only one component
of the fuzzified HSI color space. This tendency is subtle but is understandable because of the
difficulty in the evaluation judgment of the fuzzy approaches against the more traditional crisp
ones.

2.4.4 Dominant color extraction based on multi-dimensional particle swarm
optimization

In another paper Kiranyaz et al. [Kiranyaz 09] address dominant color extraction as a dynamic
clustering problem. They use techniques based on Particle Swarm Optimization (PSO) for
finding the optimal number of dominant colors in the HSV color space.

The clustering problem requires the determination of the dimension of the solution space (i.e.
number of clusters, \( K \)) and an effective mechanism to avoid local optima traps, both dimen-
sionally and spatially. This particularly complex in clustering schemes with high dimensions,
e.g. number of clusters \( K > 10 \), authors hold). The former requirement justifies the use of
the proposed MD PSO technique; while the latter calls for the other proposed technique, called
Fractional Global Best Formation (FGBF). This latter FGBF technique used in this paper tries
to avoid the premature convergence problem by collecting all promising components from each
particle and fractionally create an “artificial Global Best” (gBest) particle. Authors believe that
their gBest may guide the swarm better than the swarm’s original gBest particle [Babu 95] in
such a way that the swarm can converge to the global optimum (or near optimum) solution even
in high dimensions and usually in earlier stages.

There is still another prior problem: how many initial clusters should be used? To bypass
this they developed a dynamic clustering technique where the optimum number of clusters is
determined in run-time. The authors call it the Multi-Dimensional Particle Swarm Optimization
(MD PSO) method, which extends the native structure of PSO particles in such a way that
particles can make inter-dimensional passes. Therefore, with this technique, swarm particles
can seek for both positional and dimensional optima. So in this work, authors use both cited
techniques to find the optimal number of dominant colors in images with respect to a clustering
validity index function.
To reduce the computational complexity of RGB to HSV transformation and to speed up
the dynamic clustering process via MD PSO and FGBF, a quantization step, which limits the
color palette, is first performed. This way colors can be reduced to a reasonable number, i.e.
\(256 < N < 512\). For this pre-processing step, the authors use the Median Cut method. After
this, over each image, already in HSV color space, is applied the proposed dynamic clustering
technique (MD PSO) to extract the dominant colors.

Instead of using a typical Euclidean distance metric for all color pairs, they adopt, what
they call, a “perceptual approach”. In [Kisters 04] other authors exploited the fact that humans
tend to think and perceive colors only in 11 basic categories. Even though they cite this source,
they do not take this data into account. Instead they use 8 as the number of quantized colors,
dividing the Hue component in 8 equal segments/colors. The distance metric is obtained from
a fuzzy model. They consider a simplification of the Euclidean distance metric used elsewhere
in the color space except near the achromatic gray/black zone of the HSV color space. But they
fail by not discriminating the membership functions of their fuzzy segmentation.

The choice of the Median Cut method for the quantization in [Kiranyaz 09] was wise be-
because it is fast (\(O(N)\)) and usually the resulting image can hardly be distinguished from the
original for those selected values of \(N\) [Heckbert 82]. With the fuzzy approach in the similarity
metric we believe it reasonably reflects the human perception of the differences between colors.
Unfortunately they did not show the membership functions or the threshold they define over
the gray/black region within the HSV color space. They test their proposed method against
MPEG-7 DCD with 25 (max.) dominant colors. But it is more usual to consider a smaller num-
ber of dominant colors; in particular in the case of the MPEG-7 DCD it is 8 the usual “magical
number” [Manjunath 01]. Besides, they cite a paper where authors defend that humans tend
to think and perceive only 11 basic colors. The proposed algorithm looks for the best solution
between a range of dimensions related with the number of dominant colors. This implies that
each particle has information about the performance of the actual dimension and the best re-
esult from all past dimensions. It is implied that this has an heavy price in performance that
cannot be ignored. Worst, the proposed algorithm can never guarantee that the solution will
not converge to a local minimum simply because of over- or under-clustering. This fact together
with the usual low performance of the iterative evolutionary algorithms brings us serious doubts
about the efficiency of this approach.
2.5 Discussion

In this section we briefly summarize the previously reviewed works, by listing them all and discriminating their major characteristics. We present the next tables with all the works, in the same order as presented before, i.e., by classifying papers into 4 main groups: histogram based, segmentation based, correlogram based and hybrid/statistical based. After reviewing these papers we found that a process of color feature extraction can be summed up with only a few invariants, besides the algorithm himself, namely: the used color space, the number of colors used in the quantization step, and whether the obtained descriptor has some kind of spatial localization information of colors in it. To these invariants, we also add the database size and some evaluation criterion, when it exists in the papers, for comparison purposes. The distance comparison metric is not shown because almost every method uses the Euclidean distance and so it is not a factor of differentiation.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Spatial info.</th>
<th>Color space</th>
<th>Max. colors</th>
<th>Algorithm</th>
<th>Database</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wan 1998</td>
<td>no</td>
<td>RGB+CIEluv</td>
<td>256</td>
<td>Octree graph quantization</td>
<td>2119</td>
<td>- 1</td>
</tr>
<tr>
<td>Manjunath 2001</td>
<td>DCD: spatial coherence, CLD: spatial layout, Others: no</td>
<td>DCD: RGB, HSV, YCbCr or HMMD</td>
<td>8</td>
<td>DCD: clustering with GLA</td>
<td>5000</td>
<td>ANMRR: 0.20 (DCD with spatial coherency)</td>
</tr>
<tr>
<td>Wong 2007</td>
<td>spatial coherency</td>
<td>CIEluv</td>
<td>8</td>
<td>Clustering with GLA</td>
<td>5466</td>
<td>ANMRR: 0.10</td>
</tr>
<tr>
<td>Yang 2008</td>
<td>no</td>
<td>RGB</td>
<td>8</td>
<td>Clustering with GLA</td>
<td>2100</td>
<td>ANMRR: 0.49</td>
</tr>
<tr>
<td>Min 2009</td>
<td>no</td>
<td>modified HSV</td>
<td>8</td>
<td>Unsupervised clustering with GTC</td>
<td>5000</td>
<td>Accuracy: 19.6% (in 50 retrievals)</td>
</tr>
<tr>
<td>Bhoyar 2009</td>
<td>no</td>
<td>RGB</td>
<td>11</td>
<td>Fuzzy inference system to define linguistic variables</td>
<td>unknown</td>
<td>- 2</td>
</tr>
</tbody>
</table>

Table 2.1: Histogram based methods

As we can see from Tables 2.1-2.4 only segmentation based methods and correlograms have spatial localization of colors within its descriptors. Nevertheless, some MPEG-7 based papers, such as Manjunath et al. [Manjunath 01] and Wong et al. [Wong 07] (Table 2.1), have a particular parameter, named “spatial coherency”, which represents the overall spatial homogeneity (dispersion) of the dominant colors. Even though we already discussed the advantages of the HSV and CIEluv over RGB color space, it is convenient to salient that some recent authors still choose RGB and with interesting results. The general approach, in the traditional dominant colors processing algorithms, is to quantize the color space to obtain fewer colors and apply

1 $O(N \cdot \log N)$, $N$ is the number of dominant colors
2 Neural approach requires 60% more time than fuzzy
Table 2.2: Segmentation based methods

<table>
<thead>
<tr>
<th>Paper</th>
<th>Spatial info.</th>
<th>Color space</th>
<th>Max. colors</th>
<th>Algorithm</th>
<th>Database</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith 1995</td>
<td>yes</td>
<td>HSV</td>
<td>50 (5 per region)</td>
<td>Clustering and use of a color median filter</td>
<td>3000</td>
<td>-</td>
</tr>
<tr>
<td>Pass 1996</td>
<td>yes</td>
<td>RGB</td>
<td>64 or 512</td>
<td>Spatial coherence of pixels by connected components algorithm</td>
<td>14554</td>
<td>CCV is 7.5% slower than color histograms (but CCV is better 91% of the times)</td>
</tr>
<tr>
<td>Pass 1996</td>
<td>yes</td>
<td>RGB</td>
<td>25</td>
<td>8-connected neighbouring region growing method</td>
<td>340</td>
<td>Success rate: 54-72%</td>
</tr>
<tr>
<td>Liu 2004</td>
<td>yes</td>
<td>HSV</td>
<td>35</td>
<td>Region segmentation with the JSEG method</td>
<td>5000</td>
<td>40 relevant images (in 100 retrievals)</td>
</tr>
<tr>
<td>Younes 2006</td>
<td>yes</td>
<td>HLS</td>
<td>9</td>
<td>Fuzzy quantization of the color space</td>
<td>unknown</td>
<td>90% relevant images</td>
</tr>
<tr>
<td>Lézoray 2009</td>
<td>yes</td>
<td>RGB</td>
<td>unlimited</td>
<td>Unsupervised morphological clustering</td>
<td>200</td>
<td>Mean square error greater than with JSEG method</td>
</tr>
<tr>
<td>Kiranyaz 2010</td>
<td>yes</td>
<td>CIELuv</td>
<td>6</td>
<td>Quad-tree decomposition with Peer Group Filtering method</td>
<td>10000 + 10000 + 20000 + 1089 = 32089</td>
<td>ANMRR: 0.15 (best) to 0.39 (worst)</td>
</tr>
</tbody>
</table>

Table 2.3: Correlogram based methods

<table>
<thead>
<tr>
<th>Paper</th>
<th>Spatial info.</th>
<th>Color space</th>
<th>Max. colors</th>
<th>Algorithm</th>
<th>Database</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huang 1997</td>
<td>yes</td>
<td>RGB</td>
<td>64</td>
<td>Spatial correlation between colors</td>
<td>14554</td>
<td>Average success rate: 75% ^3</td>
</tr>
<tr>
<td>Hu 2000</td>
<td>yes</td>
<td>CIELab</td>
<td>20</td>
<td>Co-occurrence neighbourhood histogram matrix</td>
<td>835</td>
<td>8 best results in 10 queries (subjective evaluation by 13 subjects) ^4</td>
</tr>
<tr>
<td>Chun 2008</td>
<td>yes</td>
<td>HSV</td>
<td>60</td>
<td>Auto-correlogram based wavelet decomposition</td>
<td>990 + 1200 + 5420 = 7610</td>
<td>ANMRR: 0.08 (best) to 0.19 (worst) ^5</td>
</tr>
</tbody>
</table>

The best histogram based method is a paper by Wong et al. [Wong 07] (see Table 2.1), and it successfully applies a clustering algorithm in CIELuv color space. In terms of segmentation based methods (Table 2.2), we found more accurate extraction algorithms. It is the case of the JSEG implementation from Liu et al [Liu 04], the fuzzy quantization by Younes et al. [Younes 06], or the Quad-tree decomposition with the Peer Group Filtering method by Kiranyaz.

---

^3$O(N^2,d)$, $N$ is the number of dominant colors, $d <=$ image size

^4$O(n^3)$, $n$ is the number of nodes in graph

^5Only 47% of the additions of CSD (but 233% more multiplications than CSD)
et al. [Kiranyaz 10]. But none of these can achieve the uppermost method found, the autocorrelogram based method (Table 2.3) by Chun et al. [Chun 08], which achieves one of the best retrieval rates, an acceptable performance and allegedly it fits well for different sized images. But here color and texture descriptors are combined, which the majority of the reviewed methods do not contemplate. Finally in the hybrid/statistical section (Table 2.4) it is only worth to cite the Ant-Colony clustering algorithm by Huang [Huang 06], which achieves a good ANMRR (average normalized modified retrieval rank), but at the cost of a disappointing real-time performance.

Altogether reviewed papers fail in the following issue: all authors ignore the inner complexity of the human perception of colors, by not considering how a person judges dominant colors in a picture, or just by not subordinating their algorithms to real and unattached tests. Our increased value is to bridge this fault by submitting our solution to complete tests with real users.

**2.6 Summary**

Basically, the existing methods for the identification of the dominant colors of an image can be classified into four main groups: color histograms, segmentation methods, correlogram methods and hybrid/statistical approaches. In general, color histogram methods are fast and simple to compute and in some cases achieve positive results. However, in their simpler form they do not match human perception very well, simply because they do not consider the spatial localization of colors in the images. To overcome this there are the correlogram based methods, but they are more computationally demanding. A particular form of these, the auto-correlogram, brings effectiveness and better performance, but for small databases only. Segmentation based methods

6O(N^3), N is the number of dominant colors
usually achieve good results but only apply to a restrict set of images (domain restrictions). They also consider spatial localization of colors, usually by region or object segmentation. However, quite often this goal can only be achieved with the analysis of other features, like texture and shape. To extend and magnify dominant color identification some authors propose hybrid methods with several theories from different areas, such as evolutionary algorithms, mainly by considering the distance measure between two images as an optimization problem. In practice, these hybrid methods do not produce better results than the segmentation methods and their algorithms usually have slower performance, and in some cases can barely be used in real-time applications.
Dominant Colors Identification by Users

For the development of an effective solution, which will correspond to the way people perceive dominant colors, we first performed a study with users. In this Chapter we detail the survey we have made to test the way people see dominant colors in pictures. This test had two main parts: in the first part we asked users to write the dominant colors of the presented pictures and in the second part a palette of colors was presented next to each picture and people were asked to classify each dominant color(s) they see. Intentionally no definition of “what is a dominant color” or “how many dominant colors” was made, because it would affect the choices of the inquired subjects. Later this study helped us in the evaluation of our solution, and served as a basis for some of our decisions, as we will see in detail in the following chapters.

3.1 The Pictures

To better understand what dominant colors people see in pictures we performed an on-line survey. For this survey we chose a set of 25 pictures, representing real photographs, from the Stock.XCHNG \(^1\) website. All pictures are copyright free and were chosen on a realistic basis to represent what a regular person would keep in his/her personal album. The 25 pictures were

\(^1\)http://www.sxc.hu
selected from a careful pre-selection of over 300 photos: all of them are filter-free, 24 bits JPEG compressed images. The pictures main topics go from close portraits, to wide nature landscapes, sport scenes, Venice gondolas or even a crowded street of New York.

### 3.2 The Procedure

The previously introduced survey was on-line for almost 3 weeks at the Kwik Surveys \(^2\) website. This site provides an acceptable service because it allows the free hosting of the pictures and the use of different types of questions. It also exports the results in Excel format, which became an useful tool in the analysis of the results, as we will see in the following sections. The survey was advertised by e-mail to friends and family, and published on a private Facebook group.

The set of 25 pictures was divided into two series. The first series consisted of 5 pictures (see Figure 3.1) and people were asked to wrote freely the dominant colors they perceive, in their mother language. Figure 3.2 shows an example of this kind of questions.

![Figure 3.1: The first series of 5 pictures from which people had to write the names of the dominant colors.](image)

![Figure 3.2: An example of the first series of questions of the on-line survey, showing the picture and the space for users to write the name of the DCs (below).](image)

\(^2\)http://www.kwiksurveys.com
The second series was composed of 20 pictures (see Figure 3.3). In this case we put, side-by-side to each picture, a palette of 12 colors and a Likert scale with 5 levels, for users to express their level of agreement with each of the 12 colors, relatively to the corresponding picture. An example in Figure 3.4 illustrate this kind of questions. The selection of the pictures among the two different types of questions was made randomly.

The palette of 12 colors used in the second series of the survey is the one suggested by Ware [Ware 04]. It includes the following colors: red, brown, orange, yellow, green, cyan, blue, purple, pink, black, gray and white. This might sound peculiar to some extent because it uses cyan, which could be aggregated to the 'blue’ hue, in our humble opinion. However, the cyan is not a novelty, since some authors already used it on their palettes [Liu 04, Younes 06, Zhu 09]. In truth, this 12 colors palette has the same colors of the 11 JNS (Just Not the Same) palette [Berlin 69, Chang 00, Benavente 08, Bhoyar 09], described in the Related Work Chapter, plus the cyan.

Additionally we asked for personal data, such as age, sex and the use of glasses or contact lenses, for statistical purposes only and to control whether the use of glasses/contact lenses would affect the responses.

Finally, to check for color blindness (daltonism) we presented a test with three pictures taken

---

3These are the 11 basic colors which all modern cultures can appoint
Figure 3.4: An example of the second series of questions of the on-line survey, showing the palette of 12 colors (on the right) and the colors and the corresponding rating scales (below) from the known colored plates of the Ishihara test [Hardy 45]. These three pictures (Figure 3.5) were selected from the original set of 38 plates and only check for the most usual deficiency, which is red-green color deficiency, known as deuteranopia. These three pictures cannot conclude with absolute certain if a person is color blind or not. But they are an indicator of the presence of some kind of anomaly with the human vision. The survey should be answered easily in no more than 10-15 minutes. Meeting this time limit, we could not ask people to answer all the 38 plates. This might undermine the effectiveness of the survey and would frustrate users. However with these three test pictures we presumed that we could have a clue and understand why the results of some individuals could be deviant from others. If one particular person fails all (or part) of the tests we can then decide to remove those answers.

Figure 3.5: The 3 Ishihara test pictures.
3.3 Results

After closing the survey we found that from a total of 52 participants only 39 could be considered effective, i.e., 75% of effectiveness. To be effective, in this sense, a person should answer to more than 80% of the pictures. We decided to purge 12 subject replies because of the huge lack of answers (from completely blank to less than 80% complete answers). All the values and conclusions are based on the 39 effective answers.

We found that almost half of the people were in the 18-29 age group (see Figure 3.6), being 25 males, from 39 people, we remind. Relatively to the use of glasses or contact lenses, 19 use them, i.e., nearly half of the people.

Concerning the daltonism tests, we only detected 2 possible color blind people. These 2 individuals failed the tests. But none of these individuals completed the survey (were in the group of the 12 purged answers) and hence their answers were not considered in our results.

For the first series of pictures the number of dominant colors written, per person, was 1.94 on average. Considering that we only have five pictures in this first series and some people wrote up to four, five or even six dominant colors, it makes more sense to evaluate the mode. The mode is the most frequent number of colors written per each person, and in this first series was 1 for three pictures, 2 for one picture and 3 for another picture. In other words, most people only wrote 1 dominant color, most of the times. As we previously said, the maximum number of dominant colors in this first series was 6, but it only happened once. See Figure 3.7 for a complete view of these results.

For the second series of 20 pictures the average number of dominant colors people pointed was
The most frequent number of dominant colors registered (the mode) was 3. The maximum number of dominant colors someone identified was 12, which we remind is the number of colors the palette has. See Figure 3.8 where these results are side-by-side the results from the first series of pictures.

![Figure 3.7: Statistic values for the first series of 5 pictures of the on-line survey.](image1)

![Figure 3.8: All the statistic values from the on-line survey results. Note the average number of DCs per picture increased from 1.94 in the first series of pictures to 3.39 on the second series. Also the most frequent number of DCs identified (the mode) increased from 1 to 3.](image2)

### 3.4 Discussion

After this study we can conclude that when users have to enumerate the dominant colors they see, they write only 1 color. Actually, the average is 1,94 but this is because some people wrote up to 6 colors for one image. So the most frequent number of dominant colors written is 1. But we saw a fashion on the following: for the first picture presented most people wrote 3 dominant colors, for the second most people wrote only 2 and for the rest only 1. As expected, the standard deviation, a measurement of variability, is also decreasing. Considering that the pictures were set randomly this may indicate that people tend to ignore the second and/or third dominant color after the second picture. Refer to Figure 3.7.

Comparing the colors people describe in the first series with the palette of 12 colors, only in 5,42% of the times subjects wrote colors that did not belong to the color palette names (Figure 3.9). Examples of these were colors such as “beige”, “cream”, “ocher”, or “turquoise”. This

---

4In reality all of the respondents wrote in Portuguese, eg.: “beige”, “creme”, “ocre” or “turquesa”.
means that considering all the 39 subjects, for all the 5 pictures, only 5.42% of the colors they wrote belong to those unknown colors. This gives an average of only 0.06 unknown colors per picture. Inversely, a massive 94.58% of the times people wrote known colors, i.e., color names that belong to the chosen palette. This fact leads us to conclude that the choice of the palette from Ware [Ware 04] with the indicated 12 colors seems appropriate to describe dominant colors in pictures from the human point of view.

![Known vs. unknown color names (per picture) the users wrote in the first series of the survey.](image)

Figure 3.9: Known vs. unknown color names (per picture) the users wrote in the first series of the survey.

About the second series of pictures, some people interpret the classification questions as a 12 colors complete classification but, as we already stated, nowhere in the survey was said that the number of dominant colors one should indicate were the whole 12 colors. In truth, the number of dominant colors a person should set for both types of the series was never mentioned. But 3 different people interpreted the classification that way. However only one of these 3 subjects answered all the survey assigning ratings to all the 12 colors of each picture. As we can see in Figure 3.8 there is an higher variance on the second series of pictures than on the first series. This is related with the previous mentioned fact: when people have the palette right by the side of each picture they intuitively attribute and classify more colors than on the first series. We believe this is a consequence of this type of test with a Likert scale and the presence of the palette. In this second series people knew the colors they had to identify. Besides, these questions are easier for people to answer as they do not have to define the color names of the dominant colors they see.

We encounter no relation between the age, sex or the use of glasses/contact lenses and the given answers. We only found a subtle increase in the number of dominant colors for some female subjects, in the first series of pictures, where some gave 5 or even 6 dominant colors for one of the pictures, but each of these cases only occurred once and so we considered that this issue is not relevant for the study.

As we said in the previous section, the 2 possible color-blind people did not complete the survey and so their answers were not considered. The 3 test pictures used only checked for the most common type of color vision deficiency, the deuteranopia, which affects red-green hue discrimination in 1% of the males [Cassin 90], and in fact only 25 males effectively answered the
survey.

The values obtained from this survey will be used to guide the development of our solution. We will also use these results to compare the algorithms that we plan to create for dominant color identification.

3.5 Summary

In this Chapter we presented the features and details of the on-line survey we proposed to embrace several objectives. These objectives were: to help us understand how people regard the dominant colors in pictures, how many dominant colors they choose and if the palette of 12 colors from Ware [Ware 04] is adequate. This palette is used later in the development of the algorithms to segment the color space. The details of this development are exhaustively reported in the next Chapter. When people have to write the dominant colors they see in a picture they usually write only 1 dominant color. But when they have a palette of colors they choose 3 dominant colors. One last objective with this study was the following: these results from the users allow us to have a solid base to help us compare the several algorithms we will develop to identify dominant colors from the human point of view.
The Developed Algorithms

To identify the most dominant colors in pictures nearby the human perception, we started by asking people what dominant colors they see in pictures, as detailed in the previous Chapter. Then, with these results and using some of the best practices from the works presented in Chapter 2, we started the development of various algorithms for dominant colors identification. In the present Chapter we will explain the main steps we made in the development of these algorithms and how we combine fuzzy logic theory and image processing to create the proposed solutions. The fuzzy logic is used in the segmentation of the color space of the image and the image processing is in charge of dealing with the remaining process of the identification of the dominant colors.

Figure 4.1: Modules of the developed solution.

Figure 4.1 shows the diagram of the architecture, composed of two main modules:

1. Preprocessing, where the input image is resized, converted and filtered properly.
2. Dominant Colors (DCs) Identification, is the main algorithm that outputs the 12 colors dominances.

Next we will detail each of the modules, giving more emphasis on the second and more important one, where the dominant colors are identified and where our contribution is more relevant.

4.1 Preprocessing

Before the input image can be processed by any algorithm it must obey to a few rules. First, its larger side should not be greater than 400 pixels. This operation preserves the aspect ratio. Second, the original RGB components are converted to the HSV color space. This is desirable for the following steps, where the HSV components (Hue, Saturation and Value) are treated separately. And finally, some filtering is done to clean unwanted pixels, essentially noise. It is a necessary effort because digital recording devices, such as common digital cameras, often produce annoying artifacts called noise. Image noise is generally regarded as an undesirable by-product of image capture devices because it causes distortions present in the image that can obscure the subject of the photography [Stroebel 93]. The filtering step aims to enhance the image, becoming more suitable for this specific use than the original image.

We apply 3 different filters: an average filter to blur details and remove outliers, a median filter to remove “salt-n-pepper” noise, and finally a soft morphological opening on the image to remove some sharp textures on large regions. Each filter was carefully refined to give the desired result but with minimal disturbances in the original pictures. The average filter is a low pass filter and it smooths the glittering pixels and outliers [Smith 95]. Although the median filter does a better job of removing noise, with less blurring of edges, comparatively to the average filter [Szeliski 10], we opt to use both, considering that the changes in original colors are small and focus mainly on noise pixels, as we noticed in our tests. Then it is performed a soft morphological opening on the image, to uniform textures. The opening and closing operations are known to leave large regions and smooth boundaries unaffected, while removing small objects or holes and smoothing boundaries [Szeliski 10]. The opening filter is produced by the combined use of the erosion and dilation filters, and is usually used with gray-scale images. Nevertheless it can also produce positive results with colored images, by the separated application of it on each of the color components.
4.2 Segmentation of the Color Space

The first module of the proposed architecture, detailed before, works as the preparation of the input image. Then, in the second and last module (Figure 4.1), the dominant colors are identified. This module is where we propose our solution for the dominant color identification taking into account the point of view of the users.

This module produces, as output, a list of 12 colors with the corresponding dominance weights, which reflect the importance of each color in the image. These 12 colors are those referred by Ware [Ware 04] and the same used in the on-line survey (see Figure 3.4). For simplicity, from now on, we will designate these 12 color dominances as the resulting 12 bins histogram. Note that these resulting bins do not correspond directly to the count of the image pixels, but instead a normalized\(^1\) quantity of each color. The way we compute these normalized bins depends on the algorithm used. We developed 6 different alternatives for this second module, trying different approaches in the analysis of the image and the identification of the dominant colors.

The first approach segments the HSV color space using fuzzy membership functions and counts, for every single pixel of the input image, its membership grades. It is also the base for the other approaches we propose next. To completely understand this, first we will explain the method we used to segment the color space, how it differs from other approaches and the main difficulties we encountered. By inheritance we will explain the details of the palette of colors used in this segmentation process.

4.2.1 The 12 Colors Palette

From the related work study we noticed that each method selects his own number of colors for the quantization process. Some use few colors for a quicker processing, but diverging from original colors. Others use more colors, resembling the original colors found in pictures, but achieving more colors than the human eye can distinguish. Therefore, in the segmentation of the color space, some methods use 7 hues [Roire 00], and others a 9 colors palette [Younes 06] in the quantization step of the HLS color space. Other authors adopt an 11 colors palette [Berlin 69, Chang 00, Benavente 08, Bhoyar 09] in the RGB color space. All these last papers cite the work by Berlin et al. [Berlin 69], which has led to the concept of the JNS (Just Not the

\(^{1}\text{This output is desirable to be always normalized for comparative purposes.}\)
CHAPTER 4. THE DEVELOPED ALGORITHMS

Same) colors. This concept was explained in Chapter 2 and admits that the maximum number of colors that can be named by all cultures is 11. Others also use 11 colors to model color names but in CIElab [Benavente 08], which is a more challenging color space.

Finally the 12 colors palette was introduced by Ware [Ware 04], in the scope of an application for nominal information coding. The recommended 12 colors are: red, brown, orange, yellow, green, cyan, blue, purple, pink, black, gray and white. This palette has the cyan as the extra color, relatively to the previous works with only 11 colors [Berlin 69, Chang 00, Benavente 08, Bhooyer 09]. However, Colin Ware, the author of the previously cited book, did not define the precise RGB/HSV values for each color. To overcome this we use the CSS3 colors specification\(^2\).

The CSS is a style sheet language to describe the look and formatting of a document and it includes a complete specification of document elements such as color, layout or fonts. These specifications are maintained by the World Wide Web Consortium (W3C) and are only published after an active debate. We got the RGB/HSV values for the 12 colors palette from the Color Module of the last CSS3 specification. This CSS3 Color Module was published as a W3C Recommendation on 7 June 2011\(^3\). This Recommendation level is the 5th (and last) discussion level of maturity to which the modules are subjected. The RGB/HSV values of the 12 colors palette are presented in Figure 4.2.

<table>
<thead>
<tr>
<th>Color</th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>H</th>
<th>S</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Brown (saddlebrown)</td>
<td>139</td>
<td>69</td>
<td>15</td>
<td>24</td>
<td>85</td>
<td>54</td>
</tr>
<tr>
<td>Orange</td>
<td>255</td>
<td>165</td>
<td>0</td>
<td>38</td>
<td>120</td>
<td>100</td>
</tr>
<tr>
<td>Yellow</td>
<td>255</td>
<td>255</td>
<td>0</td>
<td>60</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Green (lime)</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>120</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Cyan</td>
<td>0</td>
<td>255</td>
<td>255</td>
<td>180</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Blue</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>240</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Purple</td>
<td>128</td>
<td>0</td>
<td>128</td>
<td>300</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Pink (deeppink)</td>
<td>255</td>
<td>20</td>
<td>147</td>
<td>327</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>Black</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Gray</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>White</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 4.2: The RGB and HSV values for the palette of 12 colors.

4.2.2 The Strict Achromatic Colors

We consider the HSV color space instead of the more traditional RGB because it allows us to focus on each color chromatic (Hue) without worrying about its concentration and luminosity. The first feature (concentration) is related with the Saturation component and the second (lu-

\(^2\)www.w3.org/TR/css3-color

\(^3\)http://www.w3.org/TR/2011/REC-css3-color-20110607/
minosity) with the Value component of the HSV color model. Besides, a research by Hurvich [Hurvich 81] supports the idea that chromatic perception and brightness perception are independent on most people, so it makes all sense to deal with those color components in a separated way.

However the HSV color space has two known issues: the non-representativity of the Hues for small Values or for small Saturations, and the non-representativity of Saturation under low levels of Value. Similarly to what Chamorro-Martínez et al. [CM07] did with their HSI color space, we performed a partition of the HSV colors space based on the brightness degree of each point. However, and contrary to the two zones (black and gray) defined by Chamorro-Martínez et al., we defined three zones (black, gray and white). In practice this means that we extracted 3 zones from the color space for the achromatic colors, i.e., black, gray and white⁴, loosely based on the thresholds used by Chamorro-Martínez et al. We decided to use a third color due to the results achieved with some experiments where the white color was clearly identifiable.

Figure 4.3, represents the HSV color space with our three strict zones. In summary, a color in HSV is black if its Value component is less than $T_{\text{value, inf}}$. A color is gray if its Value component is delimited between $T_{\text{value, inf}}$ and $T_{\text{value, sup}}$, and at the same time its Saturation component is less than $T_{\text{saturation}}$. A color is white if its Value is greater than $T_{\text{value, sup}}$ and its Saturation is less than $T_{\text{saturation}}$.

Then, after these strict definitions for the black, gray and white, the rest of the HSV color space is segmented using the fuzzy membership functions described below.

### 4.2.3 The Fuzzy Membership Functions

We created fuzzy membership functions for each of the 3 components of the HSV color space (Hue, Saturation and Value). For the Hue we have 9 different membership functions, one for each hue⁵. These are all trapezoidal-shaped membership functions. The rationale for the use of trapezoidal-shaped functions is studied by Benavente et al. in [Benavente 08], where the author did not find any worth gain comparatively to Gaussian-shaped functions in the definition of fuzzy membership functions.

Thus picking the 12 colors palette from Ware [Ware 04], the proper RGB/HSV values from

---

⁴Some authors distinguish the black from gray naming the former as “achromatic” and the later as “semi-chromatic”.

⁵We use the term “hue” (or “saturation” or “value”) to nominate only the first (or second or third) of the color components and the term “color” to nominate its common sense meaning, i.e., the result of the combination of its 3 components, namely hue, saturation and value.
the CSS3 Color Module specification, the fuzzy segmentation approach [Younes 06, CM07, Saeed 08, Min 09, Zhu 09, Bhoyar 09, SH10] and the trapezoidal-shaped membership functions as used by several authors [Younes 06, Li 07, Saeed 08, Benavente 08, Puranik 09] we defined our membership functions for the Hue component, as illustrated in Figure 4.4. Black, gray and white are not in this set, because they are identified before reaching these membership functions.

Figure 4.4: Plot of the 9 Hue membership functions.

Note that the universe of discourse (the Hue range) is from $0^\circ$ to $360^\circ$. And accurately there exist 2 functions for the red hue, because red is defined near the 0 (which is equal to $360^\circ$). To clear any doubts about the HSV color model please refer to the Figure 4.5 to a complete insight of the color space representation, where the Hue circle is represented for its maximum Value.

The membership functions of the Hue component (Figure 4.4) were defined taking into account the following rules:
1. The central hue of each trapezoidal membership function is taken from the CSS3 color specifications, as already detailed.

2. The kernel size of each membership function is 10° for the 'brown' and 'orange' cases, 20° for 'red', 'yellow', 'purple' and 'pink', and 40° for 'green', 'cyan' and 'blue';

3. Two membership functions always cross at the 0.5 membership grade;

4. The right slope of one trapezoidal-shaped membership function is equal to the left slope of the adjacent membership function (and vice versa);

5. Three different membership functions never overlap;

The size of each kernel was considered after an approximation of the exact values of each hue. The distances between two consecutive hues, following all the previous conditions, were carefully approximated to fill the space in similar proportions. These hues, as already mentioned, were taken from the CSS3 color specifications.

The Reasons for the Fuzzy Approach

The fuzzy approach in the segmentation of the color space is preferable to other classic strict segmentations, as thoroughly sustained by several authors [Fageth 96, Younes 06, CM07, Li 07, Saeed 08, Benavente 08, Bhoyar 09, Puranik 09, Zhu 09, Kiranyaz 09]. For instance, Bhoyar et al. [Bhoyar 09] says that “The uncertainty and vagueness present in image analysis suggest fuzzy logic as a natural paradigm.”. These authors use 11 Gaussian bell-shaped membership functions with the RGB color space. Younes et al. [Younes 06] use the HLS color space instead, and only 9 membership functions. They equally divide the Hue range in triangular-shaped functions, except for the 'green' and 'blue', which are trapezoidal-based functions. But the Lightness (L) and Saturation (S) components are both divided into 3 membership functions.

\[\text{The kernel of a fuzzy membership function is where its membership grade is equal to 1.}\]
Finally the approach by Chamorro-Martínez et al. [CM07] use the HSI divided into 10, 7 and 9 uniform equal intervals, respectively. These are all trapezoidal-shaped membership functions. Just as the work by Saeed et al. [Saeed 08] they segment the color space with equally-divided intervals and then assign a name to each color, instead of choosing the colors first and then set the intervals accordingly.

It is relatively unanimous among all these previous authors that the fuzzy approach applies well to color segmentation because the Hue circle has a subtle continuous variation of colors. For the case of far apart pure hues, such as red (HSV: 0,100,100) or cyan (HSV: 180,100,100), we almost never have doubts (Figure 4.5) but for some two pure consecutive hues it is not that easy. For instance, for the case of ‘yellow’ and ‘green’, two consecutive hues of the chosen 12 colors palette (Figure 4.2), it is difficult for a person to see the boundaries. Look at the Figure 4.6 from left to right and tell where the yellow ends and the green finally starts? Note that in Figure 4.6 only the Hue is changing and the circles were equally divided into 5º steps, from 50º to 85º. This simple test also shows the non-linear perception which the human vision has of the HSV color space. Among the factors that influence this analysis we may include the amount of light hitting the colors but also the neuro-biological properties that determine the human vision sensitivity [MacAdam 70]. With the fuzzy approach we can have a variable grade from two adjacent hues and gently combine these hues with a proportional weight.

![Figure 4.6: The difficulty with the boundaries of two consecutive hues.](image)

**The Saturation and Value Components of the Color Space**

So, as expected, we also needed to define membership functions for the Saturation and Value components of the HSV color space. They are also trapezoidal-shaped functions and both components are depicted in Figures 4.7 and 4.8. Note that for each we only defined 3 membership functions, as done by Younes et al. [Younes 06]. Those 3 uniform membership functions are enough for the correct separation of brown and pink from the other neighborhood colors in the color space, which by definition are the most problematic colors of our palette, as we will explain next. There is another non pure-hue color, which is purple. As non-pure hues we consider those...
colors which its Hue component alone is not enough to completely define them. In summary, we use these sets of Saturation and Value membership functions for the complete definition of colors like brown, orange, pink or purple.

![Figure 4.7: Plot of the 3 Saturation membership functions.](image)

![Figure 4.8: Plot of the 3 Value membership functions.](image)

### The Singular Cases of Brown and Pink

Not all of the 12 cited colors are pure-hues, i.e., for some of them the definition of the Hue component is not enough for the complete definition of the color. For instance, for the color brown (Hue near 24°) the Saturation is 86 and Value is 54 (Figure 4.2). Note that when Value is maximum we cannot see any brown in Figure 4.5. With regard to brown in the book by Ware [Ware 04] it is said that “When colors in the vicinity of yellow and orange are darkened, they turn to shades of brown and olive green.”, fulfilling the idea that brown is a somewhat dark orange or dark yellow.

The other color in our palette with no pure-hue, i.e., that needs Saturation and Value components different from maximums, is the pink color. It has HSV values equal to (327°,92,100). It is somehow between light purple and red but, as it can be seen in Figure 4.4, has a larger kernel than the slim kernel of brown. Also notice that the pink color is not so problematic because either its Value or its Saturation are practically maximums, and therefore it is a pure-hue color essentially.
To surpass these problems with no pure-hue colors - essentially the neighborhoods of brown and pink hues - we had to impose some constraints. Thus, we deviated what a color is considered to be in the end, by not only accounting its Hue but also its Saturation and/or Value components. In particular when a color has red or brown Hue and has mid or low Saturation and high Value then it is considered pink. It is also considered pink for the cases of bright purple, ie., purple Hue with high Value. In a different way a dark pink (mid or low Value) is considered purple. For the cases of low saturated bright orange, which resembles a light skin color, then it is considered brown. It is also considered brown in the cases of a dark orange, when it has mid or low Value. Inversely for a saturated bright brown (high Saturation and high Value) the color is then considered orange. All these settings were carefully adjusted by extensive experiments.

4.3 Algorithms for Dominant Colors Identification

After understanding the segmentation of the color space with the 12 colors palette and the proposed fuzzy membership functions, we can now detail the first approach of our solution. As we will see, the next algorithm is the base for the following approaches.

4.3.1 The Fuzzy Histogram Approach

In this first approach we evaluate each colored pixel grade, using the preceding explained fuzzy segmentation of the HSV color space. A complete scan of the image is done, considering the membership grade(s) for each single pixel, from the membership functions on Figure 4.4, and then a 12 bins histogram is made by counting those membership grade(s) into the respective bin(s). In the case one color has a hue with a single membership grade we count 1 for that hue. Otherwise, when a color has a hue with two different membership grades, then we evaluate the membership grade of each of the hues, and add those grades (separately) to the respective bins. For instance, for the case of hue equals to 75 in Figure 4.9 we have yellow, with hue membership grade equals to 0.83 and green, with hue membership grade equals to 0.17.

The black, gray and white cases are always decided previously. In the event that a pixel has Saturation and/or Values that fall into the regions delimited by the three thresholds we use then that pixel is promptly considered black, gray or white. Check Figure 4.3 where these 3 regions are represented. In each case 1 is added to the respective bin of the color.

On all other cases, ie., when the color components of the pixel fall out of the “black”, “gray”
or “white” zones then it corresponds to the “color” zone, and the procedure is as explained before. The respective membership grade of each hue is added to each bin, in the case the color has two different membership grades, or 1 is added in the case the color has one (maximum) membership grade. This stands for the Hue component. The Saturation and Value membership grades are used for the constraints in colors explained before, for the neighborhood colors of brown and pink.

In the end, we have all 12 color bins with the respective color quantities that were found from the image. These output quantities are then normalized (0-1) for easier comparison. As an example see Figure 4.10. Remark again that this is not a classic integer histogram counting because of the variable membership grades for those hues which belong to two different membership functions. In those particular cases, each bin weights the respective membership grade of each hue. We called this algorithm the Fuzzy Histogram and it constitutes the base for the next approaches.
4.3.2 The 3x3 Segmentation of the Image Space

In a second approach we tried the segmentation of the image into 3x3 sub-images\(^7\), and then we apply the Fuzzy Histogram algorithm on each of the 9 sub-images. We count the most dominant colors on each sub-image so that in the end, by a voting system, we have the DCs for the whole image. This counting of votes is as follows: for each sub-image, every dominant color can be worth; the dominant color just has to be above a minimal threshold to be worth, which we considered 10%. From our tests usually a color below 10% has very low importance on each sub-image.

![Figure 4.11: An example of a picture with the 3x3 segmentation represented (the numbers are only representative of the 9 divisions).](image)

In detail, the developed voting system counts the DCs of each sub-image, giving more weight to the colors more dominant for each sub-image. The equation is the following: \(\text{weight}_{DC} = N/j\), where \(N = 3\) is the number of DCs per image, obtained from our previous tests with users, and \(j\) is the rank of the identified DC. The more dominant color has rank 1, the second more dominant has rank 2, etc. For example, for the first DC we have \(\text{weight}_{DC1} = 3/1 = 3\), for the second DC we have \(\text{weight}_{DC2} = 3/2\), and so on. This way we give more emphasis to the 2 most important DCs on each sub-image. This equation came after methodical tests where we adjusted the ratio between \(N\) (number of DCs per image) and the weight of each color in each sub-image.

In Figures 4.11-4.13 we show an example of the application of this algorithm. For the 3x3 segmented picture in Figure 4.11 the respective 9 fuzzy histograms are presented in Figure 4.12, and the final result, in the form of a table, is shown in Figure 4.13. This (normalized) output is the result of the voting system previously detailed.

\(^7\)Segmentation in this scope is the division of the image into several sub-images (or regions) and should not be confused with the color space segmentation previously mentioned.
Figure 4.12: The fuzzy histograms for each of the 9 sub-images. Note that the order of the 9 histograms is the same as in the image presented before, i.e., in the first row we have sub-images 1, 2 and 3, in the second row 4, 5 and 6 and in the last row 7, 8 and 9.

Figure 4.13: The normalized output from the application of the 3x3 Segmentation algorithm. From our voting system we get: red as the most dominant color, gray as the second dominant color, etc.

As detailed above, the invariant we tested here was the number of dominant colors on each sub-image. From our tests we concluded that usually 2 colors per sub-region produced the best results, and this was also the technique from the papers by Fonseca et al. [Fonseca 09] and Zhu et al. [Zhu 09]. However, we opt for the voting system explained before instead of a rigid value. With this system we give more emphasis on the 2 most dominant colors from each sub-image but allow other colors (less dominant) to be counted as well.

The choice for the 3x3 segmentation of the image space was based on the works by Gong
et al. [Gong 96] and Mehta et al. [Mehta 03]. We also tried a 4x4 segmentation, as done by Fonseca et al. [Fonseca 09], but we found no improvements over the former 3x3 approach. Some authors use a 16x16 segmentation of each image [Valova 04] or even 20x20 [Hu 00] but from the on-line survey with users we concluded that we cannot define a localization pattern in the identification of the dominant colors.

4.3.3 The Ellipsoid Mask

In a third approach we used a modified Fuzzy Histogram algorithm (already described) where we give more weight to the central zone of each picture. This was accomplished by using an ellipsoid mask, with the same size of the original image (as depicted on Figure 4.14), which has maximum weight in the central part of the picture and fades in the corners, by giving less weight on those peripheral areas. As a consequence, these weights affect the membership grades of the pixels accordingly with their respective localization on the image.

![Figure 4.14: The bright zone of the ellipsoid mask is where the weight is maximum.](image)

The motivation for this approach was based on the work by Stricker et al. [Stricker 96] where the authors consider that the dominant colors seen by a person are more relevant at the central zone of the image. This fact was not verified by the results of our on-line survey. Nevertheless the prominent global results of this approach lead us to think that sometimes, for some pictures, people look at the central zones and identify better the colors in those areas as dominants. This is also sustained by the work of Chang et al. [Chang 00] which holds that the central colored pixels are more important to visual perception than the surrounding ones. To take this factor into consideration they weighted the pixels based on their location in the image, as we did now.
4.3.4 The Saliency Map Based Mask

This is another approach that also uses a mask to select the more important regions of an image, from the point of view of the most dominant colors. Here we make use of a Saliency Map based mask. The general idea of this approach is to choose the parts of a scene/image that are more important, expecting that those regions include the most dominant colors. Like the human nervous system, the selected regions need to be prioritized, with the most relevant being processed first and the less important ones later. This selection and ordering process is called selective attention. The most influential attempt to model this selective attention mechanism, understood here as an instantaneous sensory input, and the underlying neural mechanisms, was made by Koch and Ullman [Koch 85]. They proposed that the different visual features that contribute to attentive selection of a stimulus (color, orientation, movement, etc.) should be combined into one single topographically oriented map, the Saliency map.

So given an input image, the system attempts to predict which locations in the image will automatically and unconsciously catch the human attention towards them. In practice, an input image is decomposed into a set of “feature maps”, which extract local spatial discontinuities from features as color, intensity and orientation. The algorithm used in our approach is based on the work by Niebur and Koch [Niebur 96]. Nevertheless, the use we made here of the saliency maps theory is rather partial, because we only use the 3 feature maps combined (color, intensity, orientation) which we transform in a static filter mask. A complete use would generate the feature maps and then select, in a hierarchy-based order of importance, each of the selected regions, from the more important to the less important. However, this is of no use to us because we only need the emphasized regions and its hierarchy is not relevant. We map these selected regions into a gray-scale mask and we apply it to the input image. After this we use the membership grades scanning process described in the Fuzzy Histogram algorithm. See Figure 4.15 for an example of the Saliency Map Based Mask algorithm.

4.3.5 The K-Means Clustering

Our 5th approach to identify the dominant colors in images makes use of color clustering. The process of identifying the dominant colors in a picture can be seen as a data mining task. The recognition of the different classes of pixels (colors) can be achieved by the well-known K-Means clustering. This is what have been extensively done by several works reviewed in the
CHAPTER 4. THE DEVELOPED ALGORITHMS

Figure 4.15: The Saliency Map based mask (on the right) for the picture on the left.

Related Work Chapter [Wan 98, Deng 01, Manjunath 01, Wong 03, Wong 07, Yang 08, Min 09, Lézoray 09, Kiranyaz 10]. Generally the algorithm must discover similar groups of colors in the data (pixels of the image) without using any other known structures but the color of the respective pixels. Note that the K-Means clustering algorithm is also commonly referred in the previously cited works as Generalized Lloyd’s Algorithm (GLA), but in its standard scheme is basically the same. The standard process is done by alternating between two steps: 1) Assign each pixel to the cluster with the closest mean; 2) Calculate the new means to be the new center of the pixels in the cluster.

This K-Means clustering algorithm is not far from the extraction method used in the core experiments of the MPEG-7 Dominant Colors Descriptor (DCD) [Cieplinski 01]. The Moving Picture Experts Group (MPEG) committee that developed this standard did not define a method for the identification of the “representative colors”, as they named it. However, as seen on the Related Work Chapter, authors usually use the Generalized Lloyd’s Algorithm (GLA) for this purpose. The GLA is similar to what we have done, except for the following details: at the MPEG-7 DCD it is defined a maximum of 8 dominant colors, while we allow a maximum of 12 colors (those from the 12 colors palette). The MPEG-7 DCD does not define the palette of dominant colors. It only imposes the maximum number of dominant colors (8) but not the palette. So, we decided to use the same palette of 12 colors used on the other algorithms. The spatial coherency parameter, made from a connected components analysis, and which is also part of the MPEG-7 Dominant Color Descriptor, was not implemented in our approach.

We tried out this algorithm not only to reach better results than the previous proposed approaches but also for comparative purposes against the preceding cited works. Unfortunately it did not correspond to our expectations, as it gave, in general, the worse results (see next section for more details).

The K-Means algorithm used by us to identify dominant colors assigns each point to the
cluster whose center (also called centroid) is nearest. This center is the average of all the points in the cluster. One disadvantage of this approach is that it may not yield the same results with each run, since the resulting clusters depend on the initial random assignments. Also the initial number of clusters is a rigid input as it cannot dynamically change during the run of the algorithm. We chose 3 as the initial number of clusters because 3 is precisely the number of colors people more frequently identify in our on-line survey. After the calculus of the average (mean) value of each cluster, the algorithm counts the number of pixels each cluster has and sum to the corresponding bin. This is analogous to the calculus of the area of each cluster. The correct color bin is found by the average of the pixels of the cluster. In the end, the resultant 12 colors histogram is normalized.

Figure 4.16 shows a picture and the resulting 3 clusters. Note that the 3 different mappings (white and two shades of gray) represent the resulting clusters (one tone for each cluster) and not the dominant colors perceived. The outcome for this example is represented in Figure 4.17.

![Figure 4.16: The K-Means clustering for 3 clusters (on the right) for the picture on the left. Note that the “green” of the trees and the “gray” of the road are clustered together.](image)

<table>
<thead>
<tr>
<th>DC</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Red'</td>
<td>1</td>
</tr>
<tr>
<td>'Brown'</td>
<td>0</td>
</tr>
<tr>
<td>'Orange'</td>
<td>0</td>
</tr>
<tr>
<td>'Yellow'</td>
<td>0.2475</td>
</tr>
<tr>
<td>'Green'</td>
<td>0</td>
</tr>
<tr>
<td>'Cyan'</td>
<td>0</td>
</tr>
<tr>
<td>'Blue'</td>
<td>0</td>
</tr>
<tr>
<td>'Purple'</td>
<td>0</td>
</tr>
<tr>
<td>'Pink'</td>
<td>0</td>
</tr>
<tr>
<td>'Black'</td>
<td>0</td>
</tr>
<tr>
<td>'Gray'</td>
<td>0.395</td>
</tr>
<tr>
<td>'White'</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4.17: The result of the application of the K-Means Clustering algorithm on the image from the previous Figure. Note that the algorithm only finds 3 DCs and this number is an input of the method.
4.3.6 The Classic Strict Histogram

Besides the 5 algorithms described before, we decided to implement a classic histogram. As we saw, classic histograms typify the simplest way, for a system, to measure the color quantities in a picture. And makes easier the comparisons against the other works, included in the Related Work Chapter, because most of them use a classic histogram (even though each one uses their own palette of colors).

This classic histogram algorithm is a mere count of pixels, collected on each respective bin. Some approaches we reviewed earlier in the Related Work Chapter used equal (uniform) segmentation of the color space [CM07, Saeed 08]. However, now we felt we needed to impose resemblances with the other algorithms we developed. So we used a similar segmentation of the color space, by using the same 12 colors suggested by Ware [Ware 04], but with the difference that now we use a strict non-fuzzy segmentation of the color space. Otherwise we could not use the same palette, as some colors do not have any expression with the definition of the Hue alone. Also, the ranges of the hues differ from each other and an equally divided Hue circle with 12 identical uniform divisions makes no sense at all. As expected, the results from this classic strict histogram, despite its simplicity, are quite interesting, as we will see in the next section.

4.4 Results

In this section we present the results of the comparison of the 6 algorithms used to identify the dominant colors in pictures. We used the results from the on-line survey with users as the comparison basis for all the 6 implemented algorithms. Figures 4.18 and 4.19 summarize all the results achieved. The first table (Figure 4.18) only presents the results for the first series of 5 pictures, where we asked people to write the dominant colors they saw. The second table (Figure 4.19) shows the results for the rest of the 20 pictures, where we asked people to identify and classify the dominant colors using the presented palette of 12 colors. On both tables the first three lines are ratios between a given algorithm and the result from the on-line survey. The fourth and last line, on both tables, represents the average difference between the same colors, from each algorithm and the users.

Now follows a more detailed explanation about both tables. The first line (on both tables) correspond to the average percentage of dominant colors which were common with the dominant colors of the survey, after ordering from the most to the less dominant color. Here we only
considered dominant colors from the survey above 10% ⁸. All the rest (below 10%) were ignored. This 10% threshold result from extensive observations we have made. We looked to the dominant colors that were assigned by users with global weight under 10% and concluded that those colors were not relevant in the context of dominant colors. The order of the colors of the compared sets is not important, i.e., the most dominant color has the same weight as the second most dominant color, assuming both are above 10%. To see an example of one of this ratios see Figure 4.20, where for a given algorithm 2 colors match, from the 4 dominant colors of the survey which were above 10%. Note in this example that the fifth and sixth dominant colors from the algorithm (Yellow and White) also match with those 4 dominant colors of the users. However we do not consider those two colors, as this could lead to some cases where an algorithm outputs all the 12 dominant colors above 10% and in that particular case such algorithm would have a 100% ratio. In the presented Figure, the dominant colors are in descending order and the respective normalized weight is on the right of each color.

The second line (on both tables) corresponds to the average percentage of the 3 most dominant colors which were common with the dominant colors of the survey, after ordering. Here we also considered only dominant colors from the survey above 10%, as before. The third line (on both tables) matches the same principle, but now considering the 5 most dominant colors common with the dominant colors of the survey, above 10%. Again, in these ratios the order of

⁸We normalized the output values of each color dividing all the values by the most dominant color
CHAPTER 4. THE DEVELOPED ALGORITHMS

4.4.1 The Fuzzy Histogram Approach

In terms of results against the on-line survey, we find that this algorithm selects, on average, 59% of the same most dominant colors chosen by the inquired subjects, in the case of the first series of 5 pictures, and 64.7% in the case of the second series of 20 pictures. We recall that these values are for all the most dominant colors above 10%. Considering only the group of the 3 most dominant colors chosen by people, this algorithm obtains a 40% average ratio (or 50%
4.4.2 The 3x3 Segmentation of the Image Space

The results of this algorithm are the most promising of all the 5 tested approaches. Considering all the chosen colors (above 10%) by the subjects, this algorithm selects the same colors in 65% of the times, for the first series of pictures, and 69.1% for the second series. Considering only the 3 most dominant colors this approach gathers a 40% ratio against the first series of 5 pictures, and a brilliant 56.7% ratio for the second series of 20 pictures. This ratio is the best for all the algorithms when running the second series of pictures. If we compare the one-by-one differences to each of the chosen colors from the survey, this 3x3 Segmentation algorithm misses only an average of 0.8 colors per image, and this is the best value obtained. For the second series of pictures, the average number of missed colors is equal to 1.4, and this is also the best result for the second series.

4.4.3 The Ellipsoid Mask

In terms of results this algorithm produced moderate results. Nevertheless it is not the worse algorithm and, in some cases, it even overcomes the original Fuzzy Histogram. Considering first all the colors selected by the inquired subjects, in 60% of the cases the dominant colors are the same, for the first series of pictures (63.5% for the second series). Considering only the 3 most dominant colors selected, we get with this approach 40% of the times the same 3 dominant colors. This ratio rises to 46.7% for the second series of 20 pictures. The number of missing
colors, with the direct comparison, is the second best with an average of 1.2 colors per picture, and is only exceeded by the 3x3 Segmentation algorithm.

4.4.4 The Saliency Map Based Mask

The results of this approach are more exciting than the previous one. Here in 64% of the times we can obtain the same dominant colors like the ones we obtained from the on-line survey, considering all the colors, for the first series of 5 pictures. This result is 65.7%, for the second series. Seeing only the 3 most dominant colors chosen, we have here the best result for the first series of pictures: a ratio of 46.7%. Surprisingly this is also the ratio for the second series of pictures, but here the result of this algorithm is exceeded by the ratio of the 3x3 Segmentation algorithm, which reached 56.7%.

4.4.5 The K-Means Clustering

With the K-Means clustering algorithm we have the worse results from the 6 algorithms we tested. It is even worse than the Classic Strict Histogram. With this K-Means clustering, and considering the first series of pictures, the ratio when considering all colors is only 16%. Fortunately this number rises to 36.8% if we consider the second series of 20 pictures. But this is still the weakest result. Considering only the 3 most dominant colors chosen, we cannot go further than a 20% ratio (or 30% for the second series).

4.4.6 The Classic Strict Histogram

Finally the results from the classic histogram are a surprise because they almost catch up the proposed Fuzzy Histogram approach, but only when considering all the colors and only for the second series of pictures. In all other cases it fails to beat the Fuzzy Histogram. When considering the 3 most dominant colors the results are among the worst, only achieving a 20% ratio, for the first series of pictures, and 43.3% for the second series. The average missing colors is the worst, for both of the series of pictures, obtaining an average of 3.8 missed colors per picture for the first series (or 3.9 for the second series).
4.5 Discussion

In this section we present the conclusions reached from the results shown before. We gave more importance to the results with the 3 most dominant colors because, as detailed in the Survey Chapter, most people chose 3 dominant colors for each picture, on average, in the second series of pictures. We remind that the first series of pictures were intended to check if all the 12 colors from the palette were valid, and the second series of pictures were intended to know how many dominant colors people see on average in pictures. We should also underline that the results from the algorithms presented before were always compared with the dominant colors that users identified in the survey, and that all of them used the 12 colors palette.

Overall the best algorithm is the 3x3 Segmentation, where we divided each picture in 3x3 sub-pictures and applied the Fuzzy Histogram algorithm on each. For the first series of 5 pictures of the survey, when people had to write the dominant colors they saw, this algorithm can select the same dominant colors in 65% of the cases. And it only misses on direct comparison 0.8 colors per picture, on average. For the second series of pictures, when people had the palette of 12 colors right by their side, the 3x3 Segmentation is also the best algorithm, identifying the same dominant colors above 10% in 69.1% of the times, on average. When regarding the 3 most dominant colors, this algorithm selects those same 3 dominant colors in 56.7% of the times, for the second series. For the 5 most dominant colors this algorithm is also the best, matching a ratio of 65% for the first series, and 68.8% for the second series.

The former results are better than the ones from the Ellipsoid Mask algorithm. Here we applied a mask on top of the Fuzzy Histogram algorithm to account the dominant colors most in the center of the image. This means that people do not strictly follow the perception rule of “seeing” dominant colors exclusively in the center of pictures, as was defended by Stricker et al. [Stricker 96]. Still concerning the Ellipsoid Mask algorithm, we see that it is only better than the Fuzzy Histogram algorithm when considering all colors or when considering the 5 most dominant colors, using the first set of pictures only. In all other cases it is worse than the Fuzzy Histogram algorithm. These results led us to believe that people also account the dominant colors in the borders of the pictures and not only in the center.

About the results from the Saliency Map Based Mask we have mixed feelings. On one hand its results are not bad at all, and they are always better (or equal) than the results from the Ellipsoid Mask algorithm. But when considering the 3 most dominant colors ratio, for the second
series, it cannot beat the Fuzzy Histogram approach (46.7% vs. 50% ratio). We expected more from the Saliency Map Based Mask algorithm, since this mask should be a good prediction of the most important regions of an image. We believe this proves that those regions designed from the "stimulus-driven visual selective attention", as some authors refer, do not necessarily group the more dominant colors, at least from the human point of view.

In general, the results that really disappoint are the ones obtained by the K-Means Clustering algorithm. We feel we did not follow the same path of other works in the design of this clustering algorithm. Otherwise the results should be more promising. The similarities with the Generalized Lloyd's Algorithm (GLA) used on the MPEG-7 Dominant Color Descriptor led us to stress the use of this algorithm, as it could be a basis for comparison with other works we reviewed in the Related Work Chapter. Sadly it did not match the expected results, i.e., the dominant colors identified by our K-Means Clustering algorithm are far from the dominant colors identified by people.

Finally the nice results of the Classic strict histogram should not amaze anyone. As detailed previously, we used the same palette, the same segmentation rules, and the main difference to the Fuzzy Histogram is the total absence of the fuzzy membership grades. We opted to use a strict non-fuzzy segmentation of the color space. This led to worse results than the Fuzzy Histogram algorithm, except on one case: when considering all the dominant colors, for the second series. In particular, the average number of missing colors (last line of both tables) is the worst of all algorithms, i.e., on average it misses 3.8 colors and 3.9 colors per picture, for the first and second series respectively.

After obtaining the results for all the 6 algorithms for dominant colors identification we reached the following conclusions. The best algorithm is the 3x3 Segmentation, except for a single ratio, where the Saliency Map Based Mask is better. Overall the results from the algorithms based on the fuzzy segmentation approach are pretty exciting and this fact proves the gains of its use and that our decisions were well taken. The Classic Histogram algorithm did score well in our ratios mostly because of the almost identical segmentation step. The only true disappointing results were from the K-Means Clustering.
4.6 Summary

In this Chapter we presented solutions to identify dominant colors in pictures. After obtaining the results from the on-line survey in the previous Chapter, where we asked people how they see dominant colors in pictures (and how many), here we used these data to build our proposed solution. In fact, we actually propose 6 different alternatives to identify dominant colors. We presented the proposed architecture diagram in Figure 4.1, and we detailed each one of the two modules. The second module is where we give our contribution to solve the problem of dominant colors from the point of view of the human vision. The proposed solution was decomposed into several parts. We detailed the use of the 12 colors palette, the fuzzy segmentation of the HSV color space, the difficulties and the reasons for our choices. The 6 developed algorithms were: the Fuzzy Histogram, the 3x3 Segmentation, the Ellipsoid Mask, the Saliency Map Based Mask, the K-Means Clustering and the Classic (non-fuzzy) Histogram. Then we evaluated each algorithm against the results from the on-line survey shown before (in the previous Chapter), and concluded that the best algorithms were the Fuzzy Histogram, the 3x3 Segmentation, and the Saliency Map Based Mask.
Experimental Evaluation of the Algorithms

After looking to all the previous results we chose the 3 best algorithms and submitted them to another evaluation. The chosen algorithms were: the Fuzzy Histogram, the 3x3 Segmentation, and the Saliency Map Based Mask. We also decided to use the strict Classic Histogram for a more solid comparison base, as this kind of solution is commonly seen on the majority of the analyzed papers.

We evaluated the chosen algorithms through another survey with users. This time we performed the survey on paper, instead of hosting it at an on-line service, to ensure that all answers were given with full awareness of what we were asking for.

5.1 The Pictures

We started by choosing a new set of 20 pictures, different from the one used in the previous survey. Again, all the chosen 24 bits JPEG compressed pictures are real photographs from the Stock.XCHNG (\(^1\)) website. All are copyright free and were chosen on a realistic basis of what could be a personal album a regular person would keep. The full list of the 20 chosen pictures can be seen in Figure 5.1.

\(^1\)http://www.sxc.hu
5.2 The Procedure

We submitted each picture to each of the 4 referred algorithms: the Fuzzy Histogram, the 3x3 Segmentation, the Saliency Map Based Mask, and the (strict) Classic Histogram. From each algorithm we grab the 3 most dominant colors. Then we presented these 3 dominant colors from each algorithm to people, but without telling the correspondence between each algorithm and the identified dominant colors. People were asked to classify, using a Likert scale, each set of dominant colors, from each algorithm, for each image.

As explained before, we decided to produce this survey on paper. We created two different documents: one with glossy photo-paper for the pictures, and another, with regular 90 grams paper, for the answers. The pictures document had all the 20 pictures numbered. The answers document had 4 different versions. Each version had printed the 3 most dominant colors identified by each of the referred algorithms, for each of the 20 pictures, but in different orders. The algorithms were alternated on each version. For each picture, the respondents only had to evaluate their grade of agreement for each set of 3 dominant colors presented in the answers document. This was achieved with a Likert scale with 5 levels.

Accordingly, for the A version of the answers document, the DCs for the first picture were from the Fuzzy Histogram, the DCs for the second picture were from the 3x3 Segmentation, and so on. For the B version of this document, the DCs for the first picture were from the
3x3 Segmentation, the DCs for the second picture were from the Saliency Map Based Mask, and so on. We wanted to avoid any bias on the answers, as subjects could not guess which algorithm produced what dominant colors. This way all the four algorithms were evaluated for each picture. Take a look at Figure 5.2 for an example of one of the answers document.

![Figure 5.2: An example of one version of the answers document, where we can see the numbers of the pictures, the 3 corresponding DCs, and the rating scale for the evaluation of users.](image)

Similarly to the survey done before, we used the CSS3 colors specification\(^2\) for the values to print the dominant colors.

To perform this survey we had to go along with people and ask them to answer it. These subjects were from family to friends, or colleagues and classmates. The answers were given on the work-space of the subjects, or in the case of some friends, in the library where they were. People were told the instructions to answer the survey and among these we stressed the following facts: that they had to concentrate on each picture, and that the 3 dominant colors of each picture had no significance order. Each test with one person should not take more than 6-8 minutes.

Also in this survey people were asked to answer a quick color blindness test to rule out daltonism suspicions. This was achieved with the use of 2 plates from the Ishihara test explained before, in Chapter 3.

### 5.3 Results

A total of 28 people answered to our survey. From these, 16 were males. The average age of the subjects was 30 years old: we had people from 19 to 65 years old. Concerning the daltonism

\(^2\)www.w3.org/TR/css3-color
test, all 28 people answered correctly to both the presented plates.

In terms of overall results, the algorithm with the best average score was the 3x3 Segmentation. It had an average score of 3.91 (in the 1 to 5 Likert scale), closely followed by the Fuzzy Histogram, with an average score of 3.84. The third best was the Saliency Map Based Mask, with an average score of 3.46, and the last one was the Classic Histogram, with an average score of 3.05. These results are shown in Figure 5.3.

![Table showing results of the algorithms](image)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average</th>
<th>STDEV</th>
<th>COUNT 1's</th>
<th>COUNT 2's</th>
<th>COUNT 3's</th>
<th>COUNT 4's</th>
<th>COUNT 5's</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy Histogram</td>
<td>3.84</td>
<td>0.83</td>
<td>1</td>
<td>8</td>
<td>49</td>
<td>39</td>
<td>9</td>
</tr>
<tr>
<td>3x3 Segmentation</td>
<td>3.91</td>
<td>0.78</td>
<td>1</td>
<td>8</td>
<td>43</td>
<td>44</td>
<td>8</td>
</tr>
<tr>
<td>Saliency Map Based Mask</td>
<td>3.46</td>
<td>1.03</td>
<td>5</td>
<td>13</td>
<td>39</td>
<td>48</td>
<td>25</td>
</tr>
<tr>
<td>Classic Histogram</td>
<td>3.05</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.3: Results from the evaluation of the 4 chosen algorithms.

Other results we present in this table are the average standard deviation of the answers and the raw count of each value of the scale. The algorithm with the highest variability was the Saliency Map Based Mask. The algorithm where people answered more 1’s (lowest grade in the Likert scale) was the Classic Histogram, and the algorithm for which people gave more 5’s (highest grade in the Likert scale) was the 3x3 Segmentation.

We remind that we had a total of 28 people that answered to all the 20 pictures from the survey. Given the way the tests were structured, each person answered to 5 images for each of the 4 algorithms. This gave us a total of $28 \times 5 = 140$ answers for each algorithm. So although we have a lot of samples for each algorithm, the simple comparison of the average scores is not enough to conclude which is the best algorithm, because these scores are pretty similar (see Figure 5.3). To overcome this we performed a statistical analysis with the help of the Student’s t-test. To do so we picked the algorithm with the highest average score, which is the 3x3 Segmentation, and compared its results with each one of the results from the other algorithms. See Figure 5.4 and the following paragraphs for more details.

For this Student’s t-test we considered the not-paired type test\(^3\), also known as inter-group test, because every person was able to answer to all of the four algorithms. We used the Microsoft Excel T.TEST function to estimate these probabilities.

\(^3\)The paired type tests are used when we have two related samples set; usually there is an experimental treatment or process between the two measurements.
CHAPTER 5. EXPERIMENTAL EVALUATION OF THE ALGORITHMS

Figure 5.4: Student’s t-test with the associated probabilities between the 3x3 Segmentation algorithm and the others.

From these results we can conclude that the Saliency Map Based Mask algorithm and the Classic Histogram algorithm are statistically different from the 3x3 Segmentation algorithm, because the probabilities of being equal to the 3x3 Segmentation are small. In the case of the Saliency Map Based Mask algorithm this probability is 0.05%. In the case of the Classic Histogram algorithm is only $3 \times 10^{-10}$ ~ 0%. This proves that the 3x3 Segmentation algorithm is better than the other two algorithms, because its average score is higher as we can see from Figure 5.4.

Finally concerning the Fuzzy Histogram algorithm, the probability of the samples were from the same population is 53.79%, which does not allow us to conclude with great certainty that the 3x3 Segmentation is better than the Fuzzy Histogram. We only have 46.21% of confidence. That is, although the 3x3 Segmentation presents a higher average score than the Fuzzy Histogram, ie., 3.91 vs. 3.84, this difference is not statistically significant.

5.4 Discussion

From the results presented in this Chapter we can see that the best algorithm is the 3x3 Segmentation, and the second best (Fuzzy Histogram) is not far behind. The top 3 of the algorithms is completed with the Saliency Map Based Mask algorithm.

Also in the table of Figure 5.3, we included the standard deviation for a measure of the variation of the answers people gave. From the analysis of the standard deviation of each algorithm we see that people are more in line with the 3 dominant colors set produced by the 3x3 Segmentation algorithm. As told before, from that same table we see that the Saliency Map Based Mask has the highest variability, which shows that people had more doubts with the dominant colors from this algorithm.

We can go deeper in the analysis of these results. By analyzing the average result for each

\[\text{Note: } 100 - 53.79 = 46.21\%\]
picture from each algorithm, we see that some pictures have always a good score, independently of the algorithm that produced the 3 most dominant colors set. In Figure 5.5 we show some examples, highlighted with colors. With cyan we highlighted the best algorithm for each picture (the “Sum” for each algorithm is at the bottom of the table). Among others, see for example the Pictures 7 and 11 (outlined with a red frame), which have always a good score for any of the four algorithms, even for the Classic Histogram. However the 3x3 Segmentation algorithm scores always higher than the Classic Histogram, achieving an average of 4.86 for Picture 7 and 4.57 for Picture 11.

On the other hand, if we take for instance Pictures 6 and 17 (both outlined with a green frame in Figure 5.5), we see that the average scores of the four algorithms are in accordance with the results presented earlier. In these cases, the average score of the 3x3 Segmentation reach 4.57 and 4.86, respectively. The second best algorithm in these two examples is the Fuzzy Histogram, the third best is the Saliency Map and the worst one is the Classic Histogram.

Accordingly, we have no doubts that the best algorithm to identify dominant colors is the 3x3 Segmentation, which identifies the dominant colors better than all the others for 10 of the 20 pictures (note the “Sum” line at the bottom of Figure 5.5). It also has the largest 4’s and

<table>
<thead>
<tr>
<th>Picture</th>
<th>Fuzzy Histogram</th>
<th>3x3 Segmentation</th>
<th>Saliency Map based Mask</th>
<th>Classic Histogram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>3.57 0.53</td>
<td>3.71 1.11</td>
<td>3.75 0.95</td>
<td>4.00 0.58</td>
</tr>
<tr>
<td>Stdev</td>
<td>3.40 1.11</td>
<td>3.53 0.53</td>
<td>3.53 0.53</td>
<td>3.43 0.53</td>
</tr>
<tr>
<td>Average</td>
<td>4.00 0.82</td>
<td>4.49 0.53</td>
<td>3.43 0.98</td>
<td>3.00 1.15</td>
</tr>
<tr>
<td>Stdev</td>
<td>3.67 0.53</td>
<td>2.57 0.63</td>
<td>3.86 0.85</td>
<td>2.29 0.85</td>
</tr>
<tr>
<td>Average</td>
<td>3.00 0.82</td>
<td>3.20 0.53</td>
<td>3.57 0.98</td>
<td>3.86 0.90</td>
</tr>
<tr>
<td>Stdev</td>
<td>2.95 0.53</td>
<td>0.79 0.53</td>
<td>1.84 0.53</td>
<td>1.71 0.55</td>
</tr>
<tr>
<td>Average</td>
<td>3.49 1.13</td>
<td>4.57 0.53</td>
<td>3.29 1.11</td>
<td>3.14 0.38</td>
</tr>
<tr>
<td>Stdev</td>
<td>3.49 0.53</td>
<td>2.95 0.53</td>
<td>2.95 0.53</td>
<td>2.95 0.53</td>
</tr>
<tr>
<td>Average</td>
<td>4.24 1.57</td>
<td>4.86 0.38</td>
<td>3.86 0.69</td>
<td>4.71 0.49</td>
</tr>
<tr>
<td>Stdev</td>
<td>4.24 1.57</td>
<td>3.86 0.38</td>
<td>3.86 0.69</td>
<td>4.71 0.49</td>
</tr>
</tbody>
</table>

Figure 5.5: Average grade and standard deviation for each image of each algorithm.
5’s votes (check Figure 5.3), in accordance with the Likert scale we used. From the Student’s t-test detailed before we infer that this algorithm is better than the Saliency Map Based Mask and the Classic Histogram.

Regarding the Fuzzy Histogram algorithm we do not question its good results, although the 3x3 Segmentation seems better in most of the times. Unluckily, from the Student’s t-test we performed, we cannot assure how much better the 3x3 Segmentation is comparatively to the Fuzzy Histogram. This statistical test only gives us the probability of two samples, with similar averages, being different from each other, and in this case it was not conclusive.

We see from the Figure 5.5 that most of the times the Fuzzy Histogram is better (6 of the 20 pictures) than the Saliency Map (4 of the 20 pictures) and has less variation, although with just a small difference. The worst algorithm of the four is the Classic Histogram, which has the weakest votes from people (16 votes 1’s and 31 votes 2’s, in the Likert scale), but amazingly it can still be the best for 3 of the 20 pictures. Nevertheless, the results from this Classic Histogram algorithm are more or less suitable. This is because of the segmentation of the color space step, where we followed all the rules applied on every other algorithms, apart from the fuzzy membership grades.

5.5 Summary

To verify how correctly each algorithm identifies the dominant colors in pictures from the point of view of the human perception, we accomplished a fully realistic experimental evaluation. From the previous Chapter results we selected the best 3 algorithms. These were the Fuzzy Histogram, the 3x3 Segmentation and the Saliency Map Based Mask, plus the Classic Histogram. To evaluate the four chosen algorithms we performed another survey with users. In this survey we asked people to classify the dominant colors from each of the selected algorithms. We presented a set of 3 dominant colors produced by each algorithm for each of the pictures, and asked people to evaluate their level of agreement with those sets of 3 dominant colors. This was accomplished with a Likert scale with 5 levels. To avoid any bias there was 4 different versions of the answers document, where the order of the algorithms was different from each other. After discussing the results we conclude that the best algorithm in the identification of the dominant colors is the 3x3 Segmentation, and this is consistent with the results obtained before, in Chapter 4. We should also emphasize that these exceptional results could only be possible with the help of real users,
which were the valuable source of the quality data. The 3x3 Segmentation algorithm is seconded
by the Fuzzy Histogram, and closely followed in third place by the Saliency Map Based Mask.
The algorithm that produced the dominant colors less in accordance with people perception was
the Classic Histogram algorithm. These results were validated by statistical tests.
Conclusions and Future Work

In this Chapter we present our final conclusions and the several contributions of our work. We discuss the problems that arose in the development of the proposed solution and the new issues that might be approached in the future.

6.1 Summary of This Document

In this work we proposed a new solution to identify dominant colors in pictures from the human point of view. Our solution was guided overall by the nature people perceive the DCs and our contribution was materialized by the algorithms we developed for that purpose.

In the Chapter 2, named Related Work, we started by studying the state of the art on the areas of identification, extraction or retrieval of dominant colors. From the reviewed works, we saw that some use a lot more colors than the human eye can distinguish [Kiranyaz 10], reaching up to 630 different DCs. The disparity of values we found led us to question which is the most adequate number of dominant colors a person can see in a picture. And this was an open question that all the reviewed methods did not bother to support with appropriate studies. From all the works we saw in the Related Work Chapter we classified the corresponding methods
into four main categories. However, from a different point of view, we found that some of these methods try to relate the localization of colors (regions) in the image with the importance of those regions. For these methods the idea is that these regions reflect the most dominant colors in pictures. These are mainly the Segmentation based and the Correlogram based methods. But the algorithms we saw in these methods, despite some more or less artistic processes for the DCs identification, do not objectively achieve the best results. Besides, they commonly suffer from domain restrictions. To overcome these limitations we decided to use a simpler representation: the histograms. We believe a suitable segmentation of the color space, that follows the way people see the dominant colors in pictures, is enough to correctly identify those dominant colors. This led us to the combination of the fuzzy segmentation approach with the use of the HSV color space. First, with the segmentation of the color space into several fuzzy membership functions we can assure that the dubious colors are correctly treated, as their relative weights are the corresponding membership grades. Second, contrary to the RGB color space, the HSV-based color spaces (like HSI/HSB) are the most adequate to the correct separation of the tone from the concentration and from the brightness of a color. We also found that in general the methods for DCs identification only take into account the known Precision and Recall ratios or the Averaged Normalized Modified Retrieval Rate (ANMRR). In other words, they are essentially concerned with the recovery or identification of DCs from the standpoint of the system. The problem with this limitation is that this way the reviewed solutions do not take into account the real human perception of the dominant colors. This restriction led us to think in a different strategy to deal with the evaluation of our own method. From all these considerations, by the end of the Chapter 2, we were able to define a set of requirements for the solution of the problem.

In Chapter 3 we settled an on-line survey where we asked users which dominant colors they see in a set of pictures. The objective was not only to know “which” dominant colors, but also “how many” dominant colors people see in pictures. So we grabbed a set of pictures and divided them into two series. In the first series people were asked to write freely the DCs they saw in the pictures. No definition of “what is a dominant color” or “how many dominant colors must be written” was given. In the second series people were asked to classify the dominant colors they saw, based on a palette of 12 colors, which was shown side-by-side to each picture. The average number of DCs written for each picture in the first series was 1,94 colors, and the most frequent number of DCs written was 1. People wrote the same color names of the palette from Ware [Ware 04] in 94,58% of the times. From this result we may conclude that this palette is
appropriate for the identification of DCs. The average number of DCs identified in the second series of pictures was 3.39 colors, and the most frequent number of DCs written was 3. Users identified and classified more DCs in the second series because they had the palette next to each picture, and this helped people with the responses.

In Chapter 4 we introduced the main modules of the architecture of our solution. The first module is only responsible for the preprocessing of the input image and the second module is where the dominant colors are identified. For this second module we developed six alternative algorithms. The segmentation of the HSV color space was done using fuzzy membership functions for 9 of the 12 colors. The remaining 3 colors were black, gray and white and, resembling to the approach of Chamorro-Martínez [CM07], we used 3 strict zones for these special cases. In the remaining sections of this Chapter we detailed the development of these algorithms. We developed one base algorithm, named Fuzzy Histogram, and the other algorithms are different approaches on this base algorithm. The exception is the Classic Histogram, which does not use fuzzy membership functions in the segmentation of the color space. Since the purpose was to address the problem from the point of view of the users, we evaluated each algorithm against the results we previously got from the on-line survey. So overall, against how users saw dominant colors, the best of the developed algorithms was the 3x3 Segmentation, for both series of pictures. Concerning the other algorithms, we concluded from the Ellipsoid Mask results that in practice people do not identify DCs only in the center of images. The Saliency Map Based Mask algorithm seems better than the Ellipsoid Mask but salient regions do not group all the DCs identified by users, as these results showed. About the K-Means Clustering results: the identified DCs by our algorithm are far from the DCs identified by people. The Classic Histogram algorithm had very satisfactory results because it uses almost all the rules of the base Fuzzy Histogram. However it had worse results than this last one.

In Chapter 5 we submitted the best developed algorithms to the evaluation of real users to conclude which was the best. We picked the following algorithms: the Fuzzy Histogram, the 3x3 Segmentation and the Saliency Map Based Mask. We also put together to this group the Classic Histogram as its results were very reasonable. It made no sense to use the known Precision and Recall ratios, as used by other authors, because we intended to know which is the best algorithm for dominant color identification from the human point of view. So we performed another survey with users. This time we presented people with the 3 most dominant colors identified from each of the four preceding algorithms. Each set of the 3 most DCs was then evaluated by each person.
Overall the best average score was achieved by the 3x3 Segmentation algorithm, followed by the Fuzzy Histogram and the Saliency Map Based Mask. The Classic Histogram algorithm was the poorest of the four evaluated algorithms. These results were in line with our expectations. Comparatively to what was previously obtained from the DCs users assigned from a set of pictures, we already suspected that the 3x3 Segmentation and the prime Fuzzy Histogram were the best algorithms to solve the proposed problem. And this was confirmed with the evaluation we performed and detailed in this Chapter.

6.2 Final Conclusions and Contributions

The results we obtained from the experimental evaluation of our algorithms were shown in the Chapter 5. We concluded that some of the developed algorithms were capable of identifying reasonable well the dominant colors in a similar way as users do. In particular, both the Fuzzy Histogram algorithm and especially the 3x3 Segmentation algorithm, which is no more than the application of the former on the 9 sub-images that come from the division of the image, had the best results. These two algorithms obtained the best ratios in comparison to the DCs people identified in the first of our surveys. And these results were confirmed later from the evaluation users made of the DCs identified by those algorithms. As detailed in Chapter 5 a total of four algorithms were evaluated by users. However the remaining two - the Saliency Map Based Mask and the Classic Histogram - were not so well classified by people. After all these we are in position of taking these conclusions because all of the four selected algorithms were compared and directly evaluated by real users.

We must emphasize the innovative way how we approached the problem of the identification of the DCs. Instead of querying a database for images that matched some desired features, as it is done by the majority of works we reviewed earlier, we asked users how they saw the DCs in a set of regular pictures. The goal was to know how many DCs and what DCs users see. Our contribution aimed to show that there is a more natural method to identify the dominant colors in a picture, than the ones used by the reviewed related works. Our solution was materialized on the developed algorithms detailed before in Chapter 4. We validated the palette of 12 colors from Ware [Ware 04], because with this we demonstrated that those 12 colors are enough for most people to assign the DCs of pictures. We also found that most users identify 3 DCs on each picture, which is a solid result very far from what we saw in the Related Work Chapter,
where some authors claimed that common users can distinguish up to 256, 512 or 630 different dominant colors. From the feedback of the users and direct evaluation of the four selected algorithms, we showed a different way to measure the real accuracy of the algorithms. The fuzzy segmentation of the HSV color space, combined with the palette of 12 colors we used, was also an innovative approach that is able to model fairly well the way users see the DCs in pictures.

### 6.3 Future Work

From the experimental evaluation of the selected algorithms detailed in Chapter 5 we can draw some lines for the work to be done in the future.

We have done exhaustive tests on the thresholds of the “brown” and “pink” colors. However, instead of this empirical approach from which we successfully defined the “brown” and “pink”, expert users could estimate it. This could be done with another survey. We remind that “brown” and “pink” were the colors where the main difficulties were found, mainly because they need their color components Saturation and Value (from HSV color space) different from the maximum. These tests should confirm the thresholds we found.

Due to technical limitations we were not able to perform the paper prints for the second survey (for the experimental evaluation) with the Pantone Matching System, a proprietary color space from Pantone Inc. ¹. Despite the reasonable quality of our prints, this could have helped us to obtain more reliable colors on the printings of the survey pictures.

The performance of the algorithms we developed aimed at the best solution to the problem addressed. In other words, we never considered as a requirement how quickly the algorithms identified the DCs. This can be a point to review in the future, depending on the application where the algorithms may be used. Also the implementation of the K-Means Clustering algorithm might be revised and reconsidered, as its results were far from what we expected.

One great evolution for the proposed solution in this work would be the identification of dominant colors in video. Instead of considering only static images, the future work to be done could use the best algorithms from this work in a video based application. Scanning through the video frames is the same of considering a set of several images. Besides the similarity between two consecutive frames can be taken into account in this analysis. This can be done from an adaptation of the statistics of the dominant color in the segmentation process of the video.

¹www.pantone.com
frames, as done by Ekin et al. [Ekin 03] in their sports video application.
Bibliography


[Puranik 09] Parag Puranik, P. R. Bajaj, and P. M. Palsodkar. Fuzzy based color image segmentation using comprehensive learning particle swarm optimization


Appendix

Technology

In this Appendix we will present the details of the technology of the developed software and the general modules of the developed API.

Software

The algorithms detailed above in the Chapter 4 were developed in MATLAB from Mathworks. Later all the modules and functions were converted to Java packages with the help of the MATLAB Builder Java compiler. To use the developed code a complete MATLAB Development Environment is not necessary; only the MATLAB Compiler Runtime (MCR) component is needed in the host machine, as well as the Java Runtime Environment (JRE) v1.6.0 (or higher).

The proposed solution was built with the MATLAB 7.9.0 (R2009b) software. The following 3 toolboxes were used: Fuzzy Logic, Image Processing and Statistics. In particular the Image Processing Toolbox was quite helpful because it contains routines for image analysis, image enhancement, linear filtering, and morphological operations.

Against other tool-kits available the MATLAB software has several advantages. The more
popular is the fact that it allows to perform numerical calculus and visualize the results without the need for complicated and time consuming programming. But we should also emphasize that MATLAB is actively maintained, it is very well documented and has a large base of knowledge in the web. And finally, but not least important, has a wide range of functions designed and optimized for image processing purposes.

Modules of the Developed API

In this section we will detail the functions of the developed software. These architectural modules are represented at Figure A.1. When the software is launched, first the image is loaded and then the readAndResize function is called. Then the image is passed through the applyFilters function. By now the getMembershipFunctions is called and the fuzzy membership functions for the three components of the HSV color space are outputted. These membership functions are used by any of the following modules. Depending on the input argument, one of the following modules is chosen: fuzzyHistogram, fuzzy3x3Segmentation, fuzzyEllipsoidMask, fuzzySaliencyMask, fuzzyKMeansClustering, or classicHistogram. A normalized histogram with the 12 colors dominances is produced as the outcome of the chosen function. This is the only output of the developed software.

Figure A.1: The architectural modules of the developed software.

The orange flow is the input image components, the blue flow represents the fuzzy member-
ship functions, and the green flow is the final 12 DCs dominances.
Appendix

Survey with Users

The first survey we developed was online at the Kwik Surveys\textsuperscript{1} website. Its full contents are presented below in Figures B.1-B.12.

\textsuperscript{1}http://www.kwiksurveys.com
Figure B.1: The first page from the first survey, where we asked age, sex and the use of glasses or contact lenses (all required answers).
Para cada uma das seguintes figuras, escolha a opção mais correcta, de entre as listadas.

**O que vê na figura?**

- A letra G.
- O algarismo 6.
- A letra a.
- O algarismo 8.
- Não consigo distinguir qualquer letra/algarismo.
  Limpar resposta.

**O que vê na figura?**

- O algarismo 98.
- O algarismo 20.
- O algarismo 70.
- O algarismo 29.
- Não consigo distinguir quaisquer letras/algarismos.
  Limpar resposta.

**O que vê na figura?**

- As letras “y6”.
- O algarismo 38.
- As letras “q0”.
- O algarismo 12.
- Não consigo distinguir quaisquer letras/algarismos.
  Limpar resposta.

Figure B.2: The second page from the first survey, where we tested for color blind people.
Figure B.3: The third page from the first survey, where we asked people to write the dominant colors they saw (1/2).
Figure B.4: The third page from the first survey, where we asked people to write the dominant colors they saw (2/2).
Figure B.5: The fourth page from the first survey, where we asked people to classify the dominant colors they saw (1/4).
Figure B.6: The fourth page from the first survey, where we asked people to classify the dominant colors they saw (2/4).
Figure B.7: The fourth page from the first survey, where we asked people to classify the dominant colors they saw (3/4).
Figure B.8: The fourth page from the first survey, where we asked people to classify the dominant colors they saw (4/4).
Figure B.9: The fifth page from the first survey, where we asked people to classify the dominant colors they saw (1/4).
Figure B.10: The fifth page from the first survey, where we asked people to classify the dominant colors they saw (2/4).
Figure B.11: The fifth page from the first survey, where we asked people to classify the dominant colors they saw (3/4).
Figure B.12: The fifth page from the first survey, where we asked people to classify the dominant colors they saw (4/4).
Appendix

Survey for the Evaluation

The second survey for the evaluation of the selected algorithms was performed on printed paper. Its full contents are presented below in Figures C.1-C.6.
Figure C.1: The first page from the evaluation survey, where we asked people for their age, sex and a very simplified color blind test.
Figure C.2: The 20 chosen pictures used in the survey to evaluate the algorithms.
Figure C.3: The A version of the answers document, used for people answer to the second survey.
Figure C.4: The B version of the answers document, used for people answer to the second survey.
Figure C.5: The C version of the answers document, used for people answer to the second survey.
Figure C.6: The D version of the answers document, used for people answer to the second survey.