Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2009

Marco Alexandre Rodrigues Oliveira

Identificação de Objectos em Imagens

Object Identification Within Images

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e
2009 Informática

Marco Alexandre Rodrigues Oliveira

Identificação de Objectos em Imagens

Object Identification Within Images

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Paulo Jorge dos Santos Gonçalves Ferreira, professor catedrático da Universidade de Aveiro e sob a co-orientação do Doutor António José Ribeiro Neves, professor auxiliar convidado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Aos meus pais, pelo apoio e confiança infindáveis.

o júri

presidente	**Doutora Maria Beatriz Alves de Sousa Santos** professora associada da Universidade de Aveiro
	Doutor Paulo Jorge dos Santos Gonçalves Ferreira professor catedrático da Universidade de Aveiro
	Doutor António Fernando Vasconcelos Cunha Castro Coelho professor auxiliar da Faculdade de Engenharia da Universidade do Porto
	Doutor António José Ribeiro Neves professor auxiliar convidado da Universidade de Aveiro

agradecimentos

O presente trabalho propõe-se a divulgar uma solução para um problema actual de engenharia. É um trabalho que à partida, já se sabia, seria complexo e requereria dedicar-lhe muito tempo. No entanto, confiando em mim, os meus orientadores permitiram que simultaneamente realizasse um objectivo pessoal, que foi cumprir um ano lectivo ao abrigo do programa ERASMUS em Barcelona. A eles devo um agradecimento. Aos meus pais e à Maria por suportarem a minha ausência e aos meus amigos devo outro agradecimento. Obrigado.

palavras-chave

Análise de imagens digitais, extracção de características, segmentação de imagens, classificação, pesquisa de imagens baseada no seu conteúdo, imagens semelhantes, histograma de uma imagem, algoritmo para detecção de contornos.

resumo

O aumento de conteúdo digital armazenado em bases de dados é acompanhado por uma elevada importância atribuída à disponibilização de métodos eficientes para a sua pesquisa. No caso da pesquisa de imagens, esta é, normalmente, realizada através de "keywords", o que, nem sempre garante resultados satisfatórios, uma vez que as "imagens estão para além das palavras". Para melhorar este tema é necessário avaliar o conteúdo de cada imagem. Este trabalho propõem-se a divulgar um sistema que, inicialmente, de todas as imagens presentes numa base de dados, obtenha um conjunto de elevada qualidade para posterior processamento. Este método baseia-se na análise do histograma de cada imagem e respectiva distribuição dos contornos de cada objecto presente na mesma. A este conjunto de imagens obtido, para cada instância, são extraídas características que a identifiquem. Este passo, baseia-se na segmentação de imagens e classificação de características através de uma rede neuronal. Para testar a eficiência do método apresentado nesta tese, é feita a comparação entre as características de cada imagem com as restantes, e respectiva devolução de uma lista de imagens, ordenada por ordem decrescente de semelhança. Os nossos resultados provam que o nosso sistema pode produzir melhores resultados do que alguns sistemas existentes.

keywords

Digital images analysis, feature extraction, image segmentation, classification, content-based image retrieval, similar images, image histogram, edge detection algorithm.

abstract

The rise of digital content stored in large databases increased the importance of efficient algorithms for information retrieval. These algorithms are, usually, based on keywords which, for image retrieval, do not work properly, since "images are beyond words". In order to improve image retrieval it is necessary to analyze the contents of each image. This work proposes a system that, firstly, will get a subset of high quality images from the entire database, which will help in further processing. This first method is based in the histogram and edge analysis. In a next method, for each element of the image set obtained, features are extracted. These features will identify each image in the database. In this step, an image segmentation technique and a classification with a neural network are used. This feature extraction process is tested doing comparison between each image features and all the target ones. Each image is associated with a list of images ordered by a similarity level, which allows us to conclude that our system produces better results than some other systems available.

Contents

List of Figures

Chapter 1

Introduction

Digital multimedia data such as music, images and videos have an enormous importance in our quality of life. Devices like digital cameras which allows us to create such type of data can be easily acquired and systems to store and share it are available mostly for free. The success of systems like Flickr, Picasa, Facebook, Twitter, YouTube or LastFm is irrefutable. It is increasing the number of users in the Internet sharing personal multimedia data leading to an exponential increase of digital information stored in large databases. With this phenomenon increases the necessity of systems to search efficiently for a specific document. Moreover, the way we are using to search for it, by keyword querying, is not effective enough. The user should be able to search for a song by a portion of it; to search for an image by an object present in it; or to search for a video by a scene description.

Keyword-based searching systems are dependent of the manual association of words for each image which is not reliable for three major reasons:

1. associating keywords with an image as meta-data information is a very tedious task;
2. "images are beyond words", they cannot be fully described by a list of words;
3. image interpretation varies with each user observation.

In order to find a solution for the referred problem, content-based image retrieval (CBIR) systems have been studied and proposed in the latter years.

At the moment, most of the proposed approaches of a CBIR system consist of extracting visual information from the image. These visual information components are known as *features* and the combination of them is known as the *signature* of the image. CBIR systems are based on the assumption that images which share similar features are similar. So, low-level features are extracted from images and compared with other images signatures. An important drawback of this approach is that low-level feature description of an image content is not able to reach the description that humans create with high-level semantic concepts. This problem is known as *the semantic gap* and is the main responsible for the fact that this type of searching systems has not yet reached the performance of text-based searching systems. Several work have been published about this important computer science problem, as we will describe in a next section. However, it is still unsolved.

1.1 The Problem

The main objective of any system is to achieve user satisfaction with effectiveness and efficiency. A CBIR system is not an exception and below we present a list of the user expectations when he wants to retrieve an image in a large database:

- he usually does not have an example image similar to the one desired in order to query the system with it;
- he usually does not have a region of an image or knows how to create an image that could be used in a region-based image search system;
- he does not know which global or list of local features would be perfect to enter in a system that would find images based on features inputed;
- some times the user does not have a perfect idea of which image he wants to find and the disposal of an initial set of different images would be appreciated;
- he is familiar with the way of finding data in Internet by inserting keywords and thinks it will have to work just fine for images.

Definitely, the way to please the user is to offer him a keyword-based image search system. Our approach allows the user to search the database by keyword (despite of its drawbacks previously commented). Then it would be presented a collection of images that were tagged with the same keyword or are in some way associated to this keyword, and he would be asked to choose the most similar image from the results to the one he wants. The image picked up would have to be processed next in order to extract features from it and compare them with the features of all the target images. This is where some systems fail. This feature extraction would have to be done each time a user queries the database which would drastically reduce the system performance. In this work we propose a way to resolve this important issue.

1.2 Previous Work

The number of works about efficient image retrieval is enormous. We decided to present in this section a selection of the more recent approaches that in some way influenced our own work. For information about earlier CBIR systems or about image retrieval, a comprehensive survey of the early technical achievements is provided in [2], by three of the most well known researchers in this area.

Some of the most dedicated researchers of CBIR systems in this decade, with an important amount of papers published in this field are James Z. Wang, Yixin Chen and Jia Li. They are also behind the development of two well known system: the ALIPR and the SIMPLIcity. In [3], Chen and Wang proposed a fuzzy logic approach, UFM (unified feature matching), for region-based image retrieval. In this system, **an image is represented by a set of segmented regions**, each of one is characterized by a fuzzy feature reflecting color, texture and shape properties. As a result, an image is associated with a family of fuzzy features corresponding to regions. **The resemblance of two images is then defined as the overall similarity between two families of fuzzy features and quantified by a similarity measure**, UFM measure, which integrates properties of all regions in the images. It is wrote that the UFM measure greatly reduces the influence of inaccurate segmentation and provides a very intuitive quantification. This UFM was implemented as part of their experimental SIMPLIcity image retrieval system well known in this community of research. This study alerted us for the fact that **not much attention was being paid to developing similarity measures that combine information from all regions of the image**.

In [4], the same authors proposed a region-based image categorization method using an extension of Multiple-Instance Learning, DD-SVM. **An image is considered to be a collection of regions obtained from image segmentation** using the k-means algorithm. In DD-SVM, each image is mapped to a point in a bag feature space, which is defined by a set of instance prototypes learned with the Diverse Density function. SVM-based image classifiers are then trained in the bag feature space. Though it is concluded that the proposed image categorization method has several limitations, the authors demonstrated that DD-SVM outperforms two other methods in classifying images from 20 distinct semantic classes. An important observation was made: **image semantics are inherently linguistic**, therefore, can only be defined loosely and thus, that **a methodologically well-defined evaluation should take into account scenarios with differing amounts of knowledge about the image semantics**.

In [5], still those researchers made an improvement based on the discover that in a typical content-based image retrieval system, target images are sorted by feature similarities with respect to the query and that similarities among the target images are ignored. This paper introduced a new technique, cluster-based retrieval of images by unsupervised learning (CLUE), for improving user interaction with image retrieval systems by fully exploiting the similarity information. It is proposed to **retrieve image clusters instead of a set of ordered images**.

More information about this group of researchers can be found in – `http://wang.ist.psu.edu/docs/home.shtml`.

Since 2005, Ritendra Datta has been collaborating with Li and Wang in the development of the ACQUINE system. As can be read in the system web site – `http://acquine.alipr.com/about.html` – ACQUINE is a machine-learning based online system of computer-based prediction of aesthetic quality for natural photographic pictures. This system is of an huge importance because it will help to **compose training sets** of high quality images for computers to learn concepts once this scenario will become real.

Datta, Chen, Liu and Weina Ge, in their journey to bridge the semantic gap, wrote [6]. This article makes an overview of the state of the image retrieval systems and discusses their attempt to build an image search system based on automatic tagging. They supposed the existence of different scenarios and different types of queries in a search system with the aim to **make the entire picture collection organized by keywords and to allow all types of searches under a common framework**. Moreover, it is said that they were able to categorize and tag the pictures in a very short time.

In 2008, Datta, Wang and Li with the help of Dhiraj Joshi compiled all the ideas, trends and influences of this new age in the field of information retrieval. The work presented in [7], resumes the researches in CBIR matter did until 2008. It contains an amount of information of incalculable value and comments made by those experts.

Other researchers as Jorma Laaksonen, Markus Koskela and Erkki Oja, the developers of the PicSOM, another important CBIR system, have been also dedicated to the subject. In [8], they proposed the PicSOM, which is based on pictorial examples and relevance feedback. A neural, self-organizing technique for CBIR was presented. As the MPEG-7 international standard was emerging in 2002, **they applied MPEG-7 visual content descriptors** in the PicSOM system and compared their own image indexing technique with a reference system based on vector quantization. They concluded that those descriptors used in the PicSOM system produced positive results even though the Euclidean distance used is not optimal for all of them. Also, they found out that a strong relevance feedback mechanism should be used in order to retrieve images with a good precision.

Kai Uwe Barthel is another specialist in this field who proposed another image search system in [9], [10] and [11] using keyword annotations and low-level visual metadata to generate inter-image relationships. He proposed to **model the degree of similarity between images by building up a network of linked images**. This system improves Internet image search significantly based on what **it learns, from the users interaction with the system**. In an overview of the system made in this paper, it is evidenced that in a first step, the system uses CBIR techniques to sort resulting images according to their visual similarity. In a next step, candidate images are used to refine the result by filtering out visually non-similar images. It is concluded that only features describing color are able to generate sortings that the user would consider useful. Their implementation can be resumed to the following. For each image, the system processes the 16 most representative pairs of neighboring colors. These feature vectors would be matched using the *earth movers distance*. For each new result image, the distances between its feature vector and all feature vectors of the candidate images have to be determined. Then, the minimum of these distances indicates how similar a new result image is compared to the set of candidate images.

In [12] a not conventional way of represent images is presented using a very large set of highly selective features. A framework which represents images and learns key features for any given query using the AdaBoost algorithm is proposed. In addition, it is told that AdaBoost enables a natural interface for relevance feedback by assigning a confidence to the examples. This approach is supposed to be extremely efficient if focusing on a few key features.

As cited in the previous research, relevance feedback can be applied to obtain more reliable results, although it is a task that the user does not like to execute. About this technique, in [13], Eboul Izquierdo and Divina Djordjevic reported relevant developments to help in image annotation and retrieval. It is proposed that **images should be regarded as mosaics made of small building blocks featuring good representations of color, texture and edginess**. Their system would built an object signature that would become very suitable in finding other images containing the same object. They used fuzzy clustering of the image blocks accurately and obtain the object signature. We do not agree with the efficiency of an image signature in describing an object due to the infinite possibilities that it can appear in an image. This will be discussed later in this work. Also about improving relevance feedback techniques, in [14], A. Marakakis, N. Galatsanos, A. Likas and A. Stafylopatis, proposed a new approach using Gaussian mixture (GM) models of the image features and a query that is updated in a probabilistic manner. It is shown that the system would be **based on the models of both the positive and negative feedback images**. Also, the retrieval would be based on a recently proposed distance measure between probability density functions, which can be computed in closed form for GM models. In [15], the relevance feedback is also addressed as an image retrieval technique. An interactive fusion-based search technique is investigated in both context and content-based feature spaces. This technique is based on a user's relevance feedback information to refine multiple textual and visual query. Finally, top ranked images are obtained by performing both sequential and simultaneous search processes in the multi-modal (context and content) feature space. In this work, we found relevant the fact of **extracting global and region-specific local image features in order to represent images at different levels of abstraction**. They say that the two types of image features "are complementary in Nature". The features used were the MPEG-7 Edge Histogram Descriptor and the Color Layout Descriptor as well as moment-based color, namely the mean and standard deviation of each color channel in the HSV. Texture features as energy, maximum probability, entropy, contrast and inverse difference from gray-level co-occurrence matrices were also took into consideration. A segmentation process was implemented based on the K-Means algorithm. The Bhattacharyya distance was used to calculate similarity measures and codebooks were constructed by applying a SOM-based clustering technique. They also shared a personal perspective about this clustering technique which was that it was assumed that **all relevant images belong to a user's perceived semantic category and obey the Gaussian distribution to form a cluster in the feature space**.

A subject that has been present since the CBIR systems early days is the **object ontologies**. Those ontologies try to reach high-level definition of objects but this subject never reached an important relevance. In [16], the authors addressed it. The proposed approach of a CBIR system employs a fully unsupervised segmentation algorithm to divide images into regions and endow the indexing and retrieval system with content-based functionalities. The main proposal is that low-level descriptors for the color, position, size, and shape extracted from each region are automatically associated with **appropriate qualitative intermediate-level descriptors, which form a simple vocabulary termed object ontology**. The object ontology is used to **allow the qualitative definition of the high-level concepts** the user queries for (semantic objects, each represented by a keyword) and their relations in a human centered fashion. Also, a relevance feedback mechanism, based on support vector machines and using the low-level descriptors, is invoked to rank the remaining potentially relevant image regions and produce the final query results. Another attempt to overcome the conventional content based image retrieval system using high-level features is presented in [17]. In this paper, Ying Liu, Dengsheng Zhang and Guojun Lu proposed a region-based image retrieval system with **high-level semantic learning**. As well this system would support both query by keyword and query by region of interest. It is presented that high-level concepts are obtained from features extracted of regions using a decision tree-based learning algorithm named DT-ST. They compared their system with a common content-based image retrieval system using a standard real-world image database and obtained significant improvements. Another proposal that differentiates their systems from others that use a decision tree induction algorithm: it would make use of semantic templates to discretize continuous-valued region features and avoids the difficult image feature discretization problem. Before the release of this paper, the same authors had compiled a survey of content-based image retrieval with high-level semantics [18] that strongly marks the benefits of compiling information about previous works in the field of what is being studied in order to obtain good results.

As we observe in the related researches, all use at least in one phase of the system a classification algorithm. In [19], a study about the best of three content-based image classification techniques is presented. Those three ways of classifying low-level MPEG-7 visual descriptors are: a "merging" fusion combined with a SVM classifier, a back-propagation fusion combined with a KN classifier and a Fuzzy-ART neurofuzzy network. It is evidenced that fuzzy rules can be extracted from images in an effort to bridge the semantic gap. The descriptors considered were: the Color Layout Descriptor, the Scalable Color Descriptor and the Edge Histogram Descriptor. The conclusion that the back-propagation fusion showed the best results was obtained based on the training and evaluation of two different sets.

As we will evidence in Chapter 4, the image quality has a core importance if a system has to analyze its content and extract features from it. In [20], it is discussed the direct relation between the accuracy of a registration method to the number of extracted features and to the precision at which the features are located. It can be read that **in low-resolution images, only a few features can be extracted and mostly with poor precision** and that they proposed new techniques for extracting features in this kind of images. In [21], it is invoked that **the quality of an image can be measured in terms of two components: sharpness and contrast** which can be directly translated to the camera system control variables: focus and exposure. An optimal statistical measure of image quality is developed and tested. The performance of the measure proposed, the absolute central moment, is demonstrated using series of test patterns and compared with other popular measures as the mean, standard deviation and entropy of the gray level image histogram. In [22], the authors reported to the importance of controlling the exposure of a camera. They wrote about the usefulness of the histogram for image segmentation and thresholding. It is assumed that histograms can indicate the nature of lighting conditions, the exposure of the image and whether it is underexposed or underexposed. They profit the conclusion presented in [23] to evidence the fact that the histogram of an underexposed image will be leaning to the left and the one of an overexposed image will be leaning to the right. They propose as addressed in [22] that a measure to well interpret the bars of an histogram is the mean sample value. Finally, the paper [24] gives an understanding of what is the illumination in an image as it tries to determine whether two images come from different object or the same objects in the same pose, but under different illumination conditions. The authors developed a simple measure for matching images under variable illumination, comparing its performance to other existing methods.

This list of previous researches did in the CBIR subject do not intent to represent all the work that have been done until this date. It is a compilation of what we found more relevant while we were trying to build our own CBIR system. We strongly alert that a study in this field should need a more exhaustive collection of information.

1.3 Objectives

In this work, we present, as a main goal, a strong method to analyze and filter images, in order to obtain a more coherent subset. Images from this subset are considered in the next steps, the others are simply ignored. As a second goal, we propose a system for retrieving similar images to a reference, previously picked by the user. The user would be asked to choose an image from the ones presented after a keyword-based search and the system would be able to find similar images in the database and order them by similarity. This solution will not break the user's familiar way to do searches in the Internet which is by keyword, neither will challenge the user's patience because it will produce results rapidly.

1.4 Thesis Structure

In the next chapter, the concepts related to this work are explained. In Chapter 3, our method is described in detail. In Chapter 4, results are presented. In the last chapter, some conclusions are evidenced and future work is suggested.

Chapter 2

Theoretical Background

In this chapter, we will present a set of definitions related to this work. The knowledge of the concepts explained is considered fundamental to the understanding of our method which consists mainly in the analysis of digital images.

A digital image "An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image." [25].

These amplitude values are the elements that compose an image. They are referred as **pixels**. Each pixel represents the smallest item of information inside an image. Therefore, the analysis of an image consists in analyzing the arrangement of this information. Computers consider an image as a matrix of pixels. If the image is colored, each pixel is a combination of primary colors and each position in the matrix has three values assigned to it, corresponding to the red, green and blue channel. It can also be interpreted as three matrices of the same size, one per each color channel. Otherwise the image is in a gray scale and the matrix has only one value per position.

There are three important fields in Computer Science concerned about the study of operations on images: Image Processing, Image Analysis and Computer Vision. The reason why a specific technique should belong to one field or another is a controversial theme because the boundaries of these fields related to images are hard to define. Usually, the classification is based on the purpose of the operation:

- to improve the image in some way (e.g., with better visualization, with only interesting characteristics, recovered from degradation, represented in a more compacted way, with geometrical distortions corrected or with acquisition defects deleted) it is usual to say that the image is processed.

- to extract features or attributes of the image (e.g., edges, contours, and the identity of individual objects) it is more usual to say that the image is analyzed.
- to recognize objects within images and then "perform cognitive functions related with vision" [26], the processing is associated with the computer vision field.

As mentioned before, this work is more related to the analysis of images. Our aim is to extract features from images and use these features in further identification of similar images.

The features goal is to represent an image with less values than all the pixels that compose it. This is known as a **dimensionality reduction** of an image. Instead of being represented by a matrix of pixels, the image will be mapped to a vector of fewer values. This reduction is fundamental to perform image retrieval due to the fact that, in order to find an image, the vectors of features are matched rather than matching all the pixels.

There are two ways of extracting features: taking into consideration the entire image or dividing the image in several regions and analyzing each one. The first method is called of **global features extraction** whilst the second one is called **region based features extraction**. The utility of each method depends on the objectives.

A simple way to extract features is to analyze the image or region histogram in order to obtain statistical measures as the mean, the standard deviation, the percentage of pixels between some levels, the entropy, the absolute central moment and the mean sample value.

An image histogram shows the pixels values distribution. It works as a graphical representation of "how individual brightness levels are occupied in an image" [27], plotting the number of pixels for each level (from 0 to 255).

The Figure 2.1, extracted from `http://pixelero.wordpress.com`, shows a RGB color image and the respective histogram for each channel (the red, the green and the blue channel). As an example of a conclusion that can be extracted from an image histogram is the fact that, in this one, all the histograms are more inclined to the left (the lower values corresponding to the dark pixels), which means that the image is generally dark. In spite of several statistical measures can be obtained from an histogram, as for example the entropy, which can characterize the texture of an image, they are not ideal to infer about the visual information within an image.

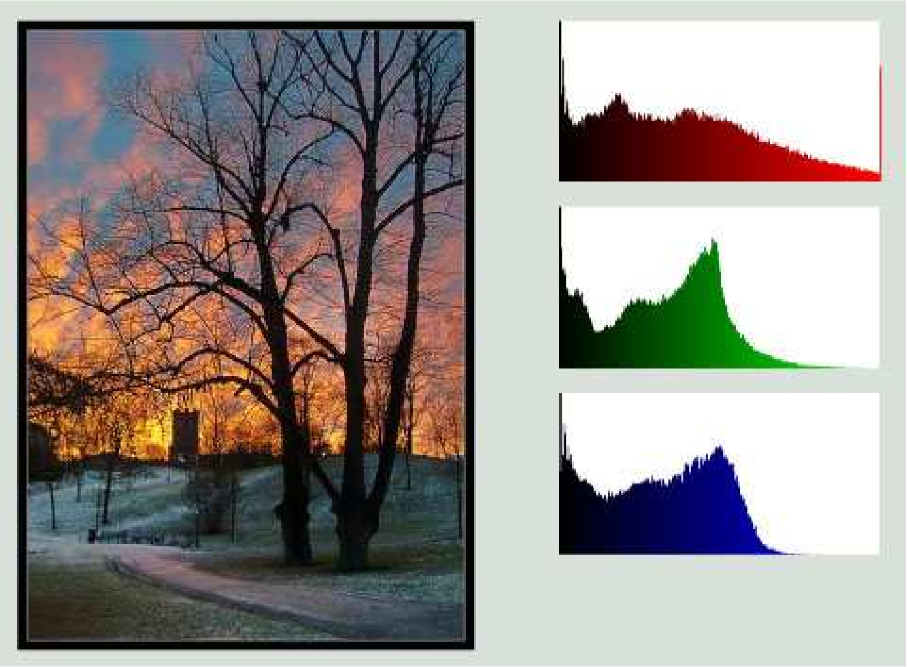


Figure 2.1: Example of an image and corresponding 3 channel histograms.

In order to help describing the content of an image, in the form of color, texture, shape, motion or localization characteristics, a standard was created. The MPEG-7, formally named “Multimedia Content Description Interface”, consists of basic structures and descriptors for multimedia content. In our case, it can be used to extract basic visual features from images.

An image is generally composed of objects and the usual point of analyzing an image is to detect those objects and identify them. There are some techniques to divide an image in parts, namely applying a threshold, detecting edges, detecting corners, detecting lines or image segmentation.

Applying a threshold means transforming an input image in another one with only white and black pixels. Pixels above a specified level are set to white and those below are set to black. It can also be used to “select pixels which have a particular value or are in a particular range” [27]. As mentioned, “it can also be used to find objects within a picture, if their brightness level (or range) is known” [27].

In Figure 2.2, an example of an image thresholded with a value of 110 is shown. The upper image is the original one after being transformed from RGB color to gray scale. In the image below, the white pixels are those with a value under 110 in the original image, and therefore the others are represented with black pixels. This is an example of the threshold utility where the

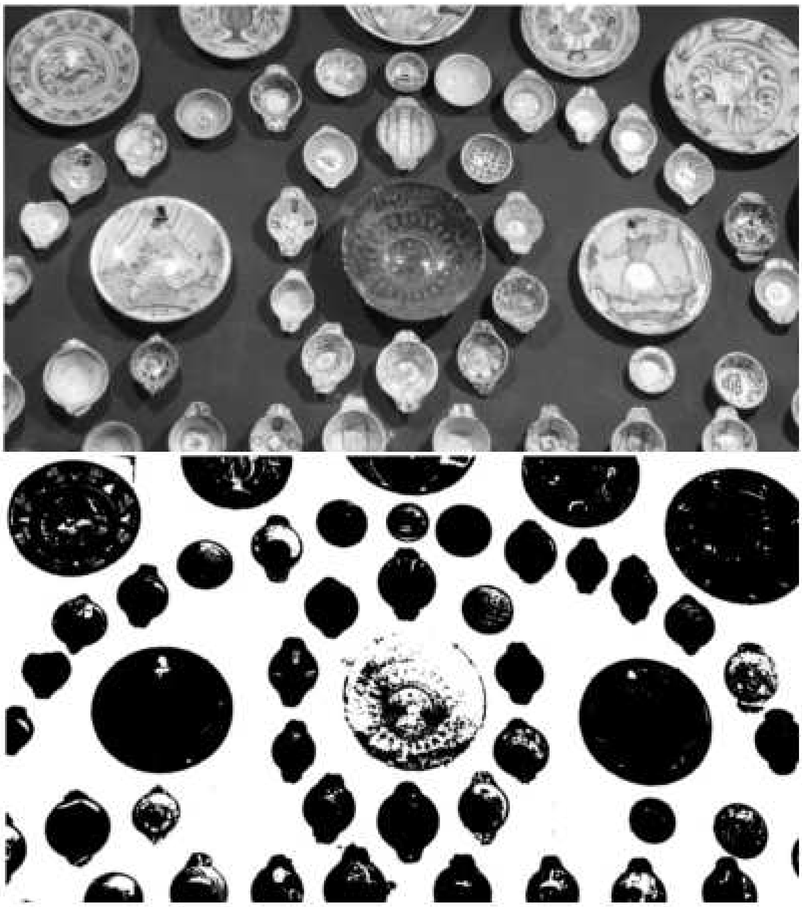


Figure 2.2: Example of the a threshold application.

background is isolated from the objects in the image.

An edge detector tracks regions of high discontinuities in intensity and sets them to white, while the remain of the image is set to black. There are several edge detectors algorithms such as, Sobbel, Canny and Prewitt. In our work we used the Canny Edge Detector which appears to be the most efficient besides of the fact that it is virtually impossible to achieve an exact implementation given the requirement to estimate normal direction.

This algorithm has three main objectives:

- optimal detection with no spurious responses;
- good localization with minimal distance between detected and true edge position;
- single response to eliminate multiple responses to a single edge.

The method can be summarized as follows:

1. a Gaussian filter with a specified standard deviation is used to reduce the noise in the image;

2. the local gradient and edge direction are computed at each pixel. “An edge point is defined to be a point whose strength is locally maximum in the direction of the gradient” [28];

3. the algorithm then tracks along the top of these edge points and sets to zero all pixels that are not actually on the top. The ridge pixels are then thresholded using two thresholds, $T1$ and $T2$, with $T1 < T2$. “Ridge pixels with values greater than $T2$ are said to be “strong” edge pixels. Ridge pixels with values between $T1$ and $T2$ are said to be weak edge pixels”, [28];

4. finally, the algorithm links the strong pixels with the weak pixels that are 8-connected with the previous ones.

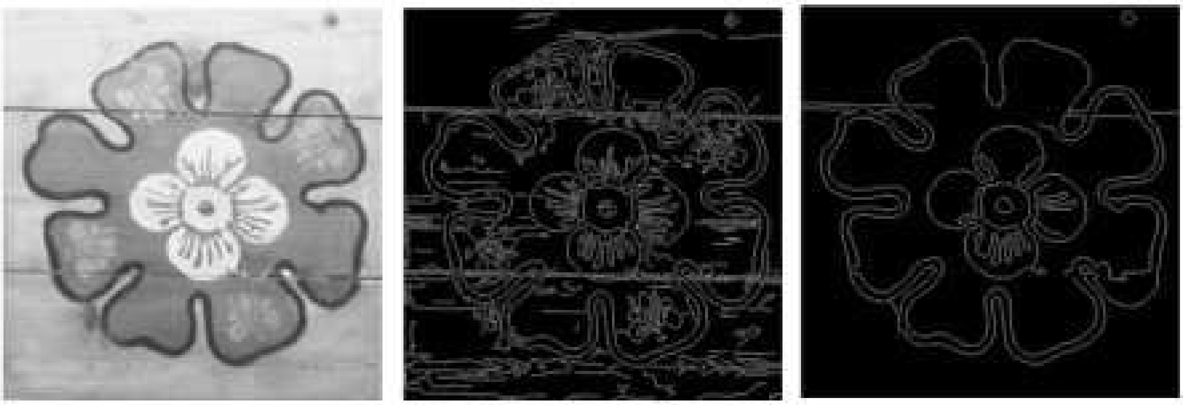


Figure 2.3: Example of edge detection with Canny’s method.

In Fig. 2.3, it is shown an example of the Canny edge detector operator. The image from the left is the original one. The others two are produced as a result of the algorithm with different parameters. The one from the middle was obtained automatically with the standard deviation used in the Gaussian smoothing equal to one. It is visible that an important account of noise is present in the resulting image. The thresholds used were 0.04 and 0.09 for $T1$ and $T2$, respectively. In the image on the right, the edges thresholds were increased to consider only edges with more discontinuity in intensity. The threshold $T1$ was set to 0.05 and the $T2$ to 0.5. We can see that the image on the right as less details than the one in the middle, which is an improvement if the point is to find objects within the image.

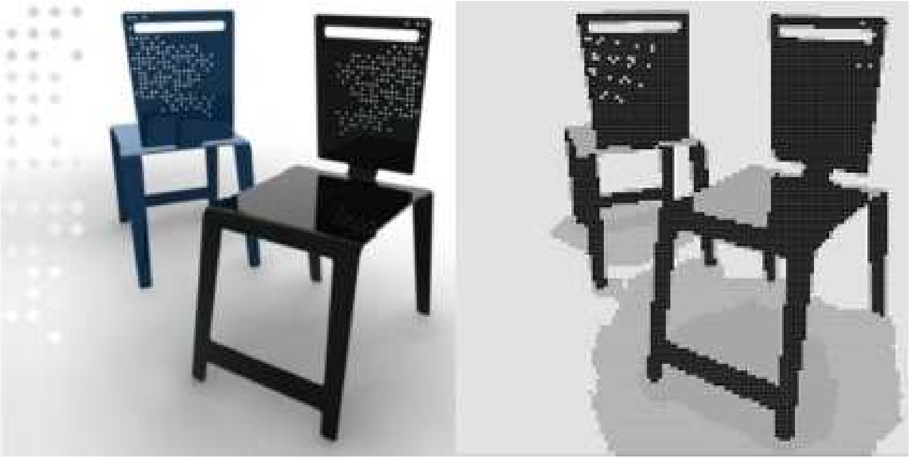


Figure 2.4: Example of an image simplified by image segmentation.

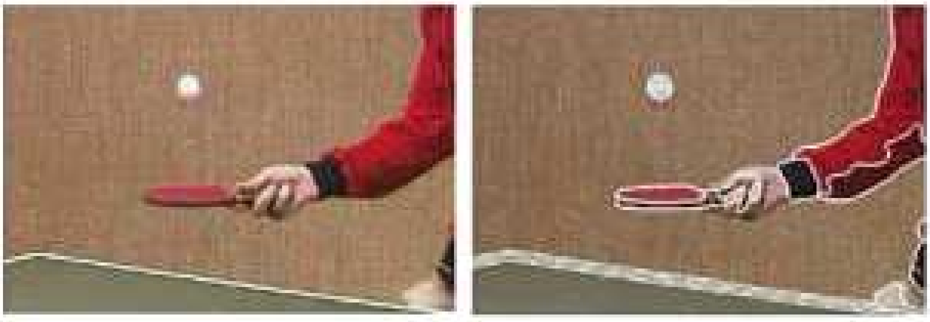


Figure 2.5: Example of an image segmentation result, extracted from [1].

Image segmentation is the process of dividing an image into multiple region (sets of pixels). The goal of it is to transform the original image in a simpler one in order to facilitate further processing. The way this process distinguishes a region from another one is analyzing the pixels similarities based in shared properties such as color, intensity or texture.

The segmentation of an image is a difficult process that, if it is efficiently obtained, clusters an image perfectly in the objects that compose it. With all the objects separated from each other, region-based features can be extracted, which can help to describe properties for further objects recognition. In Figure 2.4, an image of two chairs is processed in order to obtain a simpler one. The image from the left is the original one and the other the resulting from an image segmentation method.

In Fig. 2.5, an image segmented with the JSEG algorithm is presented (on the left) with the respective result (on the right). This algorithm takes into account basic visual properties to separate objects with different colors. The purpose of this segmentation algorithm is not to obtain a simpler image. It is to obtain an image with materials separated from each other. For example, in the figure it separated the same object, the ping-pong paddle, in two: the wood and the plastic.

Usually, the methods to segment an image are based on classification. In order to perform objects separation, features are extracted all over the image and they are progressively compared with each other with the aim of forming groups. In this case, these groups are regions of the image, also known as clusters.

Classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent to the items (referred to as traits, variables, characters, etc) and based on a training set of previously labeled items.

Some of the classification algorithms proposed in the literature are: Linear classifiers, Quadratic classifiers, K-Nearest Neighbor, Decision Trees and Neural Networks. In our work we used Neural Networks to classify region features.

Neural Networks are systems capable of modeling any desired function. Their field of application is the resolution of problems in which a mathematical model can not be obtained:

- Problems with no concrete data which difficults the application of a equations system;
- Problems highly nonlinear for which it is difficult to derivate an algebraic solution;
- Problems in chaotic systems or with high complexity.

A Neural Network can be supervised or unsupervised. A supervised network needs to be trained, comparing the outputs expected with the ones produced, modifying on-line the vector of weights. Those systems have the inconvenient that they need to know the solutions that should be produced (the expected).

In our work we used Self-Organized Maps (SOM) to learn which features better describe a region. These maps are based in the basic idea that neurons, close to each other, process sensory elements that are close to each other as well. Moreover, the connection length between neurons is reduced which permits to organize hierarchically the process of information. These are also known as maps of Kohonen [29]. They are composed by two layers: an input and a competitive and auto-organized layer. Every input neuron is connected to each neuron of the competitive layer. The neurons in neural networks usually have weighs that excite or inhibit them. This

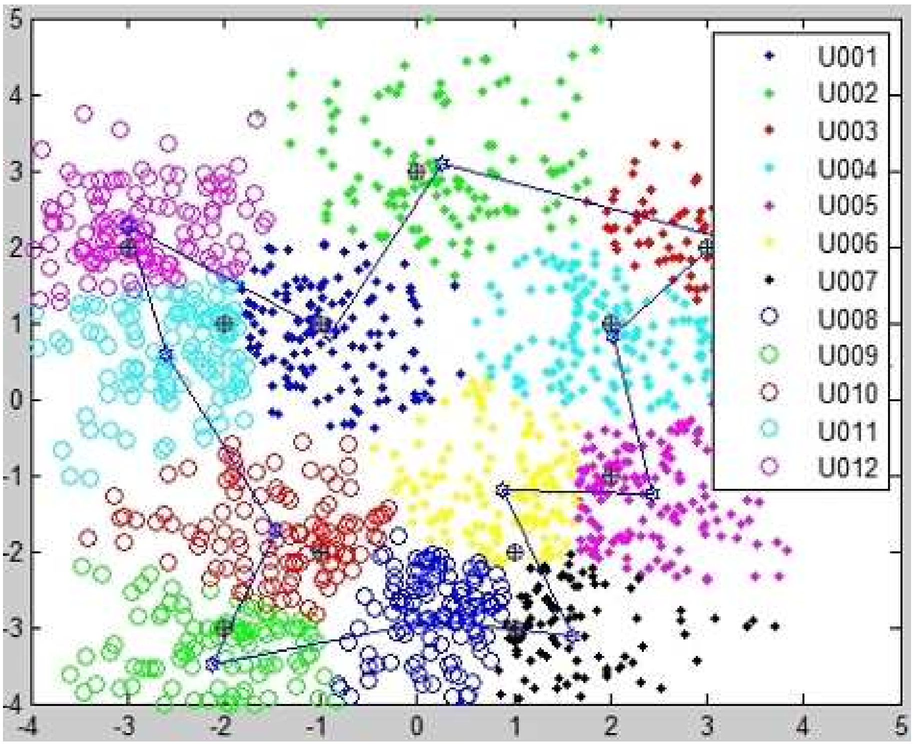


Figure 2.6: Example of a SOM network process result.

competitive layer does not use these weights. It plays a sort of a game to determine the winning neuron which confers its name. The algorithm is based in a neighbor value for each neuron that is progressively used and decreased which, besides of solving the initialization problem, also confers to these networks the auto-organization capacity.

In the Fig. 2.6, a set of 2D features was classified in 12 clusters. These features were inputs in the algorithm as a matrix of $2 \times N$ elements also called **patrons**. Each element was plotted in the graphic with a color and a shape as it was previously determined to belong to one cluster. The blue line links all the centroids of the clusters that the algorithm should have found. These centroids were marked with a red circle filled with green and a blue cross. As it can be seen, the algorithm clusters quite well the input patrons. The algorithm does constantly measures of similarities between vectors. The SOM can use any distance measurement to achieve it. In our work, we used the Euclidean distance.

The reason why we used this specific classification algorithm is the previous knowledge of how many clusters the feature space should be divided. This type of neural network can classify it in an unsupervised way with satisfied efficiency.

Euclidean distance is a way to measure the distance between two vectors. The vectors can

have N-dimensions which is the advantage of using it. This distance value is given by:
If $V = (v_1, v_2, ..., v_n)$ and $U = (u_1, u_2, ..., u_n)$ the distance d is
$d = \sqrt{(v_1 - u_1)^2 + (v_2 - u_2)^2 + ... + (v_n - u_n)^2}$

Our goal is to use these concepts to develop a method for the retrieval of similar images within a large database. Image retrieval is another science behind our method - the science of searching and retrieve images in databases of digital images. There are two types of systems that do this task: those that use keywords and find all the images related to it and those that analyze the content of the image in order to extract features that characterizes the image and find all the ones with similar characteristics.

The keyword-based image retrieval systems are based on matching textual descriptions to those presented in the metadata of an image or in its entourage. In contrast, a content-based image retrieval (CBIR) system do not depend on textual information to do its task and only takes into account the image content such as colors, textures and shapes. CBIR is one of the main topics under research in Computer Science nowadays.

Chapter 3

Proposed Method

A keyword-search for an object in a large database like *Google images* produces as outcome all kind of images related to the keyword. The user expects to find one or more examples of an image with the desired object. Moreover, the images should be returned with good quality, which is directly related with the illumination. Preferably, the user also wants an image that contains the entire object and an image where the object presence is evident.

In order to help the user to retrieve a good match of the object expected, we propose in Section 3.1 a preliminary image analysis and filtering process of all the images queried. The method ignores images that would not pass the filter. A subset of good quality images, related to the keyword, is obtained which will please the user and also be a better sample for further processing as feature extraction or concept learning.

In a next step, the system proposed would need the user to choose one image from those analyzed and presented. The image chosen would then be analyzed in order to find the number of regions in which it can be divided.

As it was mentioned in Section 1.2, global features are not efficient to identify the image content and several surveys, dedicated to finding the better local features set that would correctly characterize an image content, were presented. Knowing in how many regions the image can be divided, permits to extract specific features from each region and therefore, compose an array that defines the image, referred as *signature*. Due to the fact that the regions can have all kinds of shapes and dimensions, features are extracted from blocks of 16×16 pixels and classified using a number of feature vectors equal to the number of regions. Dividing an image into blocks reduces the dimensionality of the feature space. This size of blocks was chosen experimentally, being the best compromise between efficiency of the results and the computer processing cost.

In Section 3.2, the extraction of features and the classification process are presented.

Considering that each image has a correspondent *signature*, it is easier to compare them in order to check if they are similar. The process of comparing them is explained in Section 3.3.

An organization of the system is proposed in the last section of this chapter. There, we reveal how we intent to make viable the conciliation of all this heavy computational processes.

3.1 Image Analysis and Filtering

Analyzing an image consists in checking three main characteristics:

1. the image quality;
2. the existence of content within the image;
3. the distribution of the image content.

As cited in [21], the *Absolute Central Moment* is the best statistical measure that can be extracted from a digital image to quantify its quality. However, despite the fact that some images have an high quality, if they have a too high or to low entropy, i.e., a large amount of changes in contrast or almost none changes in contrast from one pixel to another, it makes the task of finding an object very difficult. In order for our system to provide better results, in this first process, any images that have ACM lower than 10 or higher than 60, as well as those that have an entropy higher than 7 or lower than 1 are ignored.

- A too high *entropy* means that the image has too much information or a lot of texture within it and will difficult the classification task. So an image with *entropy* higher that 7 will be ignored. If an image has a low *entropy* it would mean that it has almost none information, which is also not desired. An image with *entropy* lower than 1 will also be rejected.
- An ACM too low means that the objects in the image do not have a good contrast with the background or that they are not focused. So an image with ACM lower than 10 will be ignored. Neither an image with too much contrast is desired, so if its ACM is higher than 60, the image will not be considered as well.

The use of these values is explained in the next chapter with clear examples of their effectiveness.

In a next step, the system would transform the image in a simpler one but with enough information to accomplish the preliminary analysis and filtering process. In order to achieve it, the system would segment the image with the *Canny Edge Detector Algorithm*. We use this algorithm with the following parameters: $std = 1.0$, $T1 = 0.05$ and $T2 = 0.5$. The standard deviation equal to one was tested as the better value to smooth the image and, therefore, reduce important noise presence within it. The two thresholds were experimentally chosen as those which better inhibit the detection of irrelevant edges. The segmented image is a black and white image, where the white pixels represent the edges detected. If no edges were detected, the image is ignored as not having any object in it.

The content distribution is analyzed by checking if any relevant amount of white pixels were detected in the image contours. If so, it probably means that an object in the image is not completely represented. This is not desired if the object is the one the user is searching for.

A last step is to determine the dimensions of the object in the image. This process checks if the content of the image occupies less than 10% by counting the white pixels presence in each row and columns of the edge detector resulting image. If the sum of the number of rows where white pixels are present is less than 30% of the number of rows of the image and the number of columns where white pixels are present is less than 30% of the number of columns of the image, the image is also ignored.

After this preliminary process, we propose that the images must contain a binary entry in the image metadata that would work as a flag: if the image passes this process, the flag would be changed to one. Otherwise, the image would be ignored and the flag would be equal to zero.

3.2 Feature Extraction and Classification

Firstly, our aim is to find the number of regions in which the image should, visually, be divided. If an image is composed by three different objects, each one with a single color, this process should return a value equal to '3'. With this acknowledgment, the next process will be able to classify an image in an unsupervised way because it will know exactly how many regions composes the image. An analysis to the gray scale image histogram is performed. The histogram is divided in 8 gray intensity intervals ([0-31],[32-63],[64-95],[96-127],[128-159],[160-191],[192-223],[224-255]). For each interval, there is a corresponding amount of pixels, so 8 values are important right now. The number of relevant regions in an image is given by the number of those values that is higher than the total number of pixels (size of the image) divided by 32. The efficiency of these values to detect the number of relevant regions in an image is demonstrated in the next chapter.

Knowing in how many regions the image can be divided, will instruct the system to learn which features may describe better each region. This learning process consists in dividing the image in blocks of 16×16 pixels. For each block, three features are extracted: the *entropy*, the *absolute central moment* and the *mean sample value*. With these features, a vector is created. The goal is to find which is the vector of three features that better describes each relevant region of the image. Since the system knows the number of regions, an unsupervised learning process based in a neuronal network can be applied. The one chosen was the SOM Network. With this network algorithm, all the feature vectors are processed and a number of vectors equal to the number of image regions is obtained. These vectors define the image *signature*. The way SOM Networks work was demonstrated in Chapter 2.

In order to find similar images, these image *signatures* must be compared in such a way that a value of similarity should be obtained.

3.3 Similarity Measurement

Now that the system is able to obtain an array of n features vectors from each image, where n is the number of the image regions, similarities between two images may be calculated. This method has the purpose of returning a value of similarity between an image and each of the others, so at the end these images can be orderly organized. The euclidean distance between images *signature* is in the base of our method. The lower this distance is, the higher is the level of similarity between the two vectors. With this process, if a user chooses an image as the one for which the system must find similar ones, the system can display the results obtained, ordered by the euclidean distance between *signatures* of each of the images processed.

3.4 System Organization

We propose an organization of the system in three stages:

1. All the images in the database are analyzed and filtered as mentioned before. Some images would be ignored in this phase and others would be further processed. An algorithm is used to distinguish between images already processed and those which were not. Moreover, this also permits to know if the image passed through the filter has good or bad quality. Our suggestion is to create a binary entry in the metadata which would prove the image good

or bad quality. Another content of the metadata would be the image signature. If this entry is empty, it means that the image quality were not yet analyzed. Otherwise, it can be filled only with zeros if the process concluded that the image has bad quality or filled with the image signature if the image passed the filter. With this entry in the metadata, it will not be necessary to process an image twice, which is the key for our system to work fast, since feature classification is a computationally heavy process. With all images classified, the system would be ready to receive the user's queries.

2. A process of querying an image by a keyword would be offered to the user, as depicted in Figure 3.1 and referred as the first step. In a second step, the database would return images associated with the keyword. Then, the user should choose one image in order to search for similar ones. In a fourth and fifth step, the system would access constantly the database and calculate the similarity between the image picked up by the user and all the target images. In a last step, the system would return all the images queried, ordered by similarity to the input image.

3. A background process would be in execution in order to rearrange the tags of the images that were not correctly tagged. This would be done according to the similarity measure of the image *signature*. Moreover, images without any tags would be automatically annotated. Finally, this process would be responsible for the detection of new images inserted in the database. They could be found by their empty *signature* in the metadata. These images would be analyzed, filtered, features would be extracted, *signatures* would be made and they would be automatically tagged based on the tags of the similar images.

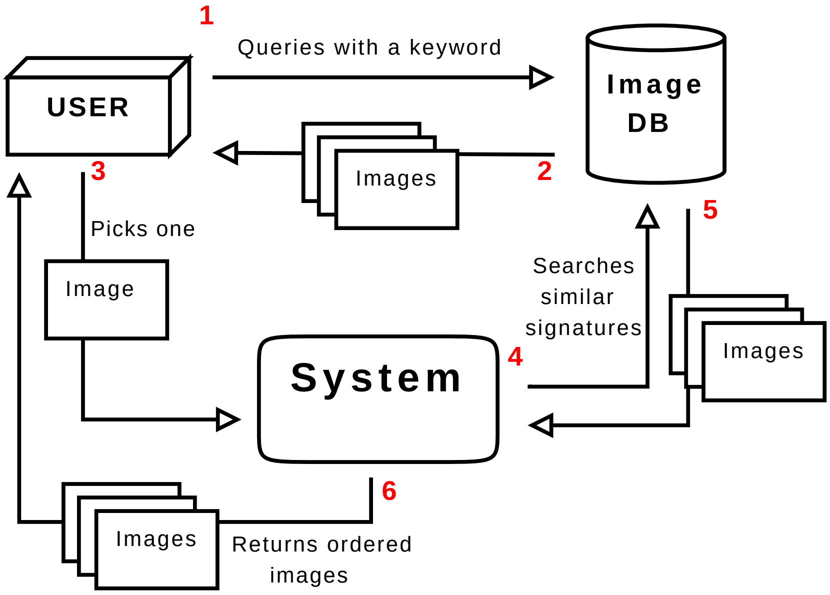


Figure 3.1: Layout of the system.

Chapter 4

Results

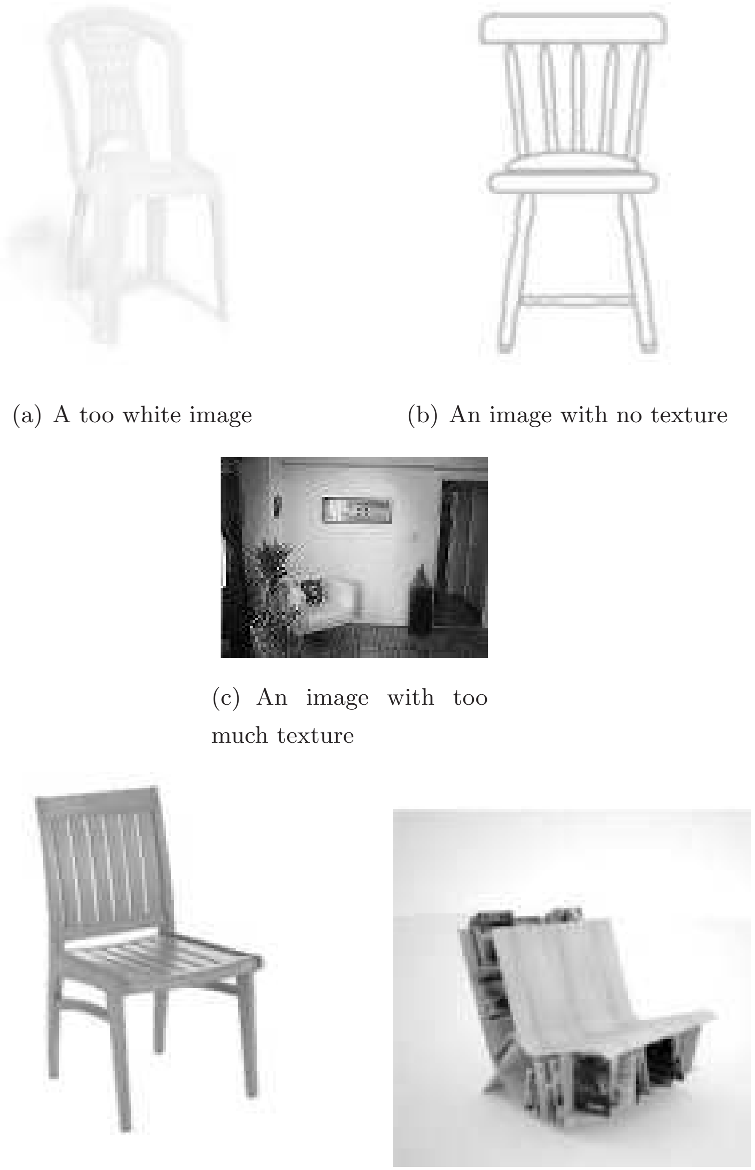
In this chapter, we present the results obtained for each process that composes our method. We chose the object "chair" as an example. Firstly, images of *chairs* with different sizes were downloaded from the Internet to simulate the access to a database. These images were used to obtain almost all the results present in this chapter. In the end of the chapter, a comparison with another systems is made.

4.1 Image Analysis and Filtering

As described in the previous chapter, the image quality analysis is based on the *entropy* and on the *absolute central moment* of the gray level image *histogram.* In order to attain that these two statistical measures are efficient to determine if an image should be consider for further processing as having a good quality, a set of statistical measures were extracted from all the images of chairs downloaded previously. In Fig. 4.1 we can see some examples of chairs and the respective statistical measures extracted.

Analyzing the Table 4.1, we can observe that the system should take the following conclusions for every image:

- Figure 4.1(a): the image is too white - low percentage of pixels in the [50-200] region of the image *histogram.* The object represented has a very poor contrast with the background - too low ACM value. The image is not appropriate for content retrieval.
- Figure 4.1(b): the image is too white - low percentage of pixels in the [50-200] region of the image *histogram.* The object in the image has a very poor contrast with the background -



(a) A too white image

(b) An image with no texture

(c) An image with too much texture

(d) A good example of the a chair

(e) An image wrongly tagged as a chair

Figure 4.1: Images of chairs

too low *ACM* value. The content has almost none texture - the *Entropy* value is very low. The image is also not appropriate for content retrieval.

- Figure 4.1(c): the content of the image has too much contrast - the *ACM* value is high and the image has too much information. There is a chair but also several other objects that will badly influence the features classification: high *entropy*.

Image	Mean	Std. Dev	[50-200]	[50-200](%)	Entropy	ACM	MSV
4.1(a)	251.1536	7.4517	107	0.14	2.3606	2.8893	4.9943
4.1(b)	248.3078	19.8659	6130	7.24	1.6436	6.1613	4.3696
4.1(c)	118.8599	67.8270	4731	63.08	7.7055	61.4286	2.9016
4.1(d)	231.9775	42.1209	17013	21.70	2.8584	18.7001	3.8657
4.1(e)	215.2055	49.3093	19927	17.04	6.3666	26.7047	4.5060

Table 4.1: Statistical Measures extracted from the histogram of the images shown in Fig. 4.1. The column under the label "[50-200]" means the number of pixels that are in the region of the histogram between the value 50 and 200. The column under the label [50-200](%) present the same values in percentage. The other measures are described above.

- Figure 4.1(d): the image is perfect for feature classification and therefore for content retrieval: it has good contrast with the background - normal value of ACM. It has only one object as content with normal texture - normal value of *entropy*.

- Figure 4.1(e): this image will be considered by the system as having a chair in its content. It will be further processed and its features will be classified, despite the fact that it is not a regular chair the object in it. The image has good contrast with the background and not too much information - normal values of ACM and *entropy*.

These results are only a few portion of the images analyzed. However, they are enough to demonstrate the reason why we chose the *absolute central moment* and the *entropy* of an image *histogram* as the measures that will determine if the image will be taken into account in the next phases.

After studying the image quality, its content is analyzed. If an image do not have any content it must be ignored.

The image is processed by the *Canny Edge Detector* algorithm with the previously explained standard deviation and thresholds values ($std = 1.0$, $T1 = 0.05$ and $T2 = 0.5$). This algorithm produces a segmented image from the original one, as the example in Figure 4.2 shows. On the left of the figure is the original image. In the middle is a result produced with thresholds automatically calculated for better edge detection. On the right is another result of the edge detector algorithm with the thresholds set manually in order to obtain a simpler image.

To determine the amount of edges in the image, the sum of each row and the sum of each column are calculated. If one of these vectors does not have any value different of zero (does not count any white pixel), the image has no content and is ignored.

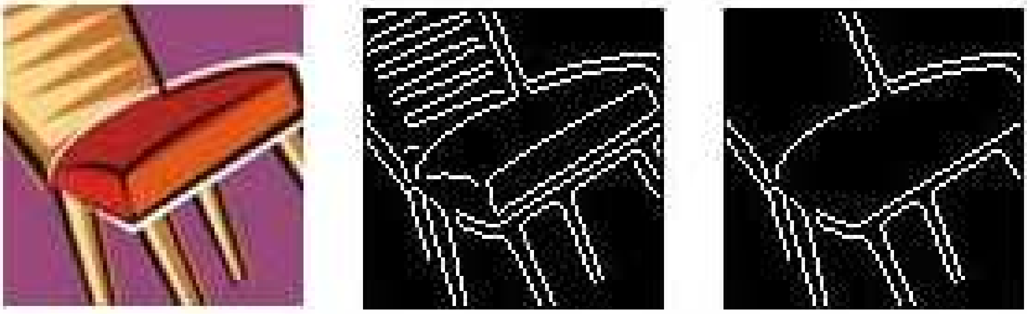


Figure 4.2: Example of application of an edge detector system.

In the next step, if the edges are presented in the image borders, it would probably mean that the content in the image is not entirely represented and the image is also ignored. This operation is performed checking if the first two lines and columns of pixels in the image and the last two lines and columns, have relevant white values. This task is done analyzing the previously obtained sum of rows and columns vectors. If the sum of the two first and last positions of those vectors is different than zero, the image has content in the borders which is not desirable because the system needs good instances, in order to produce efficient results.

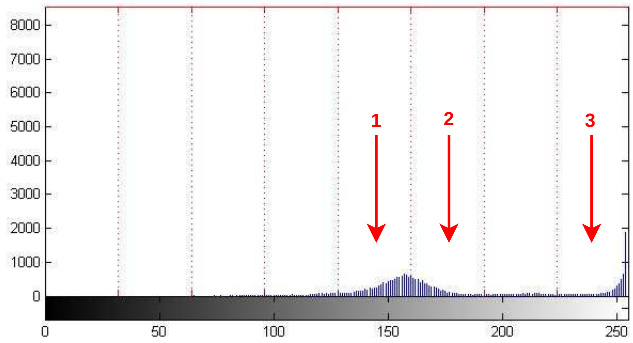
The last step is to check the dimensions of the image content. If the content width or height is lower than 10% of the images size, this image is not taken into account. The system analyzes again the sum of rows and columns vectors, as explained in the previous chapter.

4.2 Feature Extraction and Classification

In this process, the image *histogram* is analyzed again. It is divided in 8 zones, each one with 32 levels of gray. Fig. 4.3 shows an example of an image analyzed and the respective *histogram*. This image has a resolution of 280 × 280 (78400) pixels. The histogram shows the amount of pixels in each value of the gray scale. Table 4.2 contains those amounts in each of the 8 ranges.

Focusing our attention in the Fig. 4.3(a), it can be seen that the system must be able to find three different regions in this image: the white background, the light gray that corresponds to the wood in the chair and the dark gray that corresponds to the shadows in the chair. Considering the number of pixels, 78400, divided per 32 is equal to 4900. As it can be seen in the table, there are three values higher than 4900. The first two correspond to the two gray regions of the chair mentioned and to the intervals marked in the Fig. 4.3(b). The last value higher than 4900 corresponds to the white background an it is marked with a '3' in the histogram's figure. Many

(a) A good example of a chair.



(b) The respective histogram.

Figure 4.3: Example of the segmentation process.

other images were used to confirm that those values are the better ones to obtain correctly any future image. The rate of success dividing the total number of pixels by 32 was 100%. There were analyzed 115 images.

Knowing in how many regions can be divided an image, it will be easy to apply a neural network in our system in order to learn which features better classify each region.

[1-32]	[33-64]	[65-96]	[97-128]	[129-160]	[161-192]	[193-224]	[225-256]
0	1	49	1175	9342	6169	1701	59963

Table 4.2: Example of an image histogram analysis.

In this process the image is divided into blocks of 16×16 pixels. For each block, a histogram is computed and three features are extracted from it: the *entropy*, the *absolute central moment* (the usual suspects) and the *mean sample value*.

A SOM Network is used to cluster these features mainly because it is an unsupervised classification algorithm. This type of neural network is explained in the Chapter 2. Following the example shown before, in the Fig. 4.3, and knowing that the system would divide this image in three regions, the results obtained after feature classification are 3 vectors, one for each regions detected, of coordinates in a 3 dimensional feature space (each vector has 3 features). Each vector is the learned center of a cluster:

- 0.1375; 0.1817; 0.2819 - the cluster of the white background region;
- 5.1704; 36.1278; 3.8607 - the cluster of the dark gray region;
- 4.9286; 16.4167; 3.9258 - the cluster of the light gray region.

Still it is important to save the values in an array with lines of 8×3 ordered values in order to boost the image similarity measure process. This array will be saved in the image metadata. If the image has less than 8 regions, as it is the case of the example, the remain of the row is filed with zeros. This array is the image *signature*. From the 115 images analyzed, only 49 passed the filters and reached this point. From all these images, features were extracted which took almost 6 minutes in a Intel Core 2 Duo CPU P8400 working at its maximum frequency of 2.26GHz.

At this point, all the images in the database were filtered, segmented, features were extracted and classified and the image signature of all these images was saved in its metadata. The last process is, based on an image and its *signature*, search for similar *signatures* and respective images.

4.3 Similarity Measurement

To search for images similar to an example, a comparison has to be made. In this case, the images *signatures* are compared based on the calculation of the Euclidean Distance.

In Fig. 4.4, we can see a global result of the system. The system searched for the five most similar images to those in the left of the figure. The results for each image are presented at the right, ordered by its respective euclidean distance to the example's *signature*.

After inserting in the database several images of the Fig. 4.3(a) rotated, it is presented in Fig. 4.5 (in the bottom column) a new list of similar images to this one. The most similar image found is the same flipped horizontally and the next one is the same but rotated by 90 grads. This proves the system's insensitivity to image transformation which has relative importance.

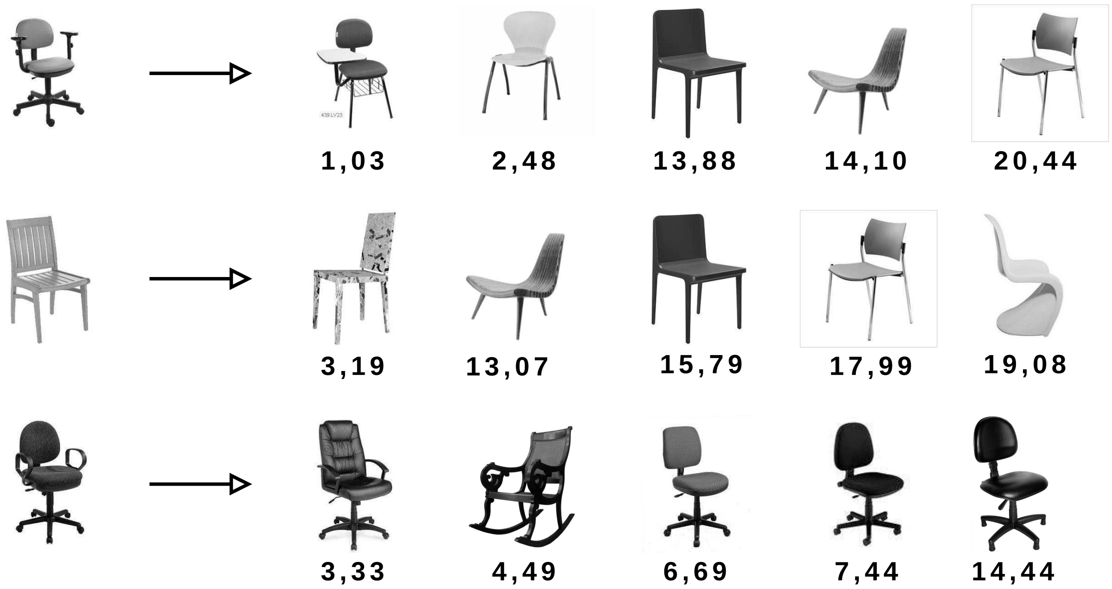


Figure 4.4: Some results of images containing a chair.

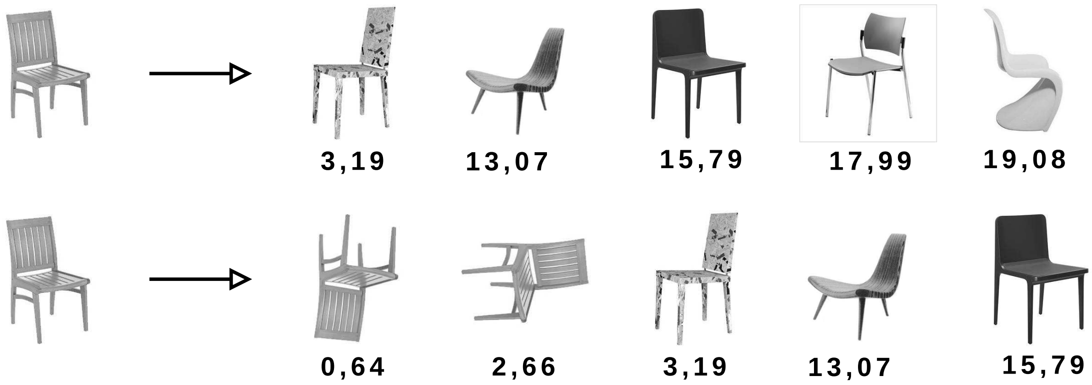
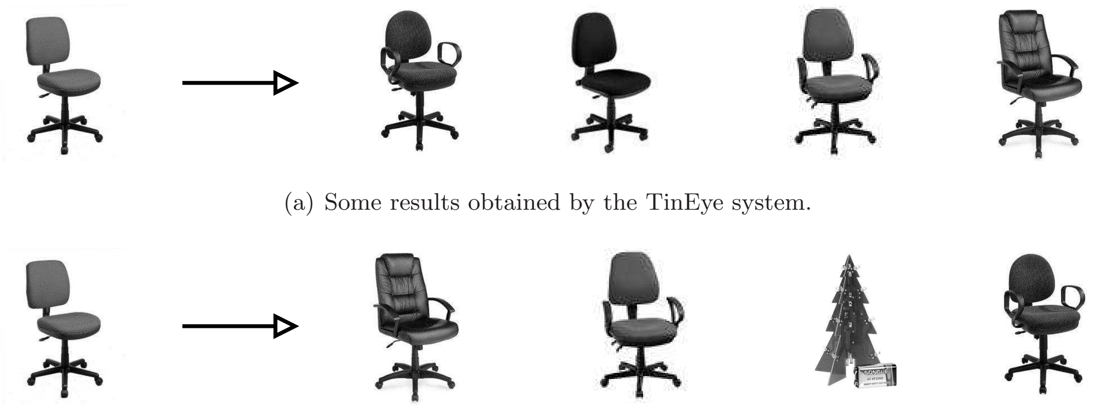


Figure 4.5: Results proving the system is insensitive to image transformation.

4.4 Comparison with existing systems

We used TinEye Reverse Image Search Engine (`http://tineye.com/`) to search for images similar to the image on the left of the Fig. 4.6(a). The system produced as a result the images of this figure. We introduced those results in our system, as well as images of different objects, namely trees, cars and apples. The Fig. 4.6(b), shows the four most similar images to the same one introduced in the TinEye. Our system produced almost the same results. The only difference was the appearance of an image of a tree which is an example of a *false positive* result: the system produced a result that was previously known that it would be wrong. This proves the importance of having well tagged images in the database so it will be easier for a CBIR system to accomplish its tasks. An efficient annotation of all the images is the best starting-point for our system.



(a) Some results obtained by the TinEye system.

(b) Some results obtained by the proposed method.

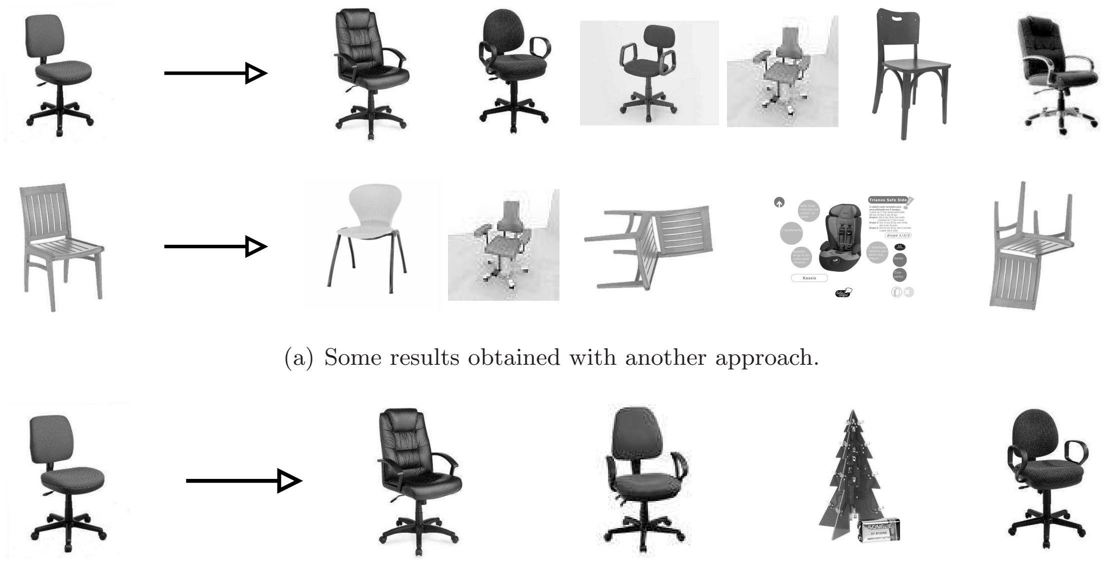
Figure 4.6: Comparison of results with another system.

Another existing system is ALIPR (`http://alipr.com/`). We introduced the same input image in this system. Then it asked us to choose some tags like *indoor*, *man-made*, *animal*, *thing*, *tool*, *photo* and *decoration* from a list. Those tags would help to find similar images but from the results produced, we concluded that it works with a too little database of images or it is not working for this type of image. The resulting images that were suppose to be similar to the chair were almost all pictures of people without any relation to a chair. As we could investigate this system analyze the image to automatically generate a list of tags and then search for images with similar tags.

4.5 Comparison with another approach

Another approach of the problem is to divide each image in 9 same-sized pieces. For each piece a vector of three features would be created. All the nine vectors resulting from one image would composed its *signature*. The process of comparing image *signatures* would be the same as it was presented: by euclidean distance. The features used were the entropy, the absolute central moment and the mean sample value of each piece histogram.

The Fig. 4.7(a) shows on the top, two results from this approach. With the left images as input, the system finds the rest of the rows as the most similar images to the one inputed. The system we proposed finds the results shown in Fig. 4.7(b). It can be observed that this new approach produces better results than the one presented previously. The only problem is that if



(a) Some results obtained with another approach.

(b) Some results obtained by the proposed method.

Figure 4.7: Comparison of results with another approach.

the same object is presented in two different images placed in two different positions, it do not guarantee to find those images as the most similar to each other. As it can be seen in Fig. 4.7(a), the row of the bottom presents an image in the left that was used as input to the new system. The rest of the row was obtained as results. The third and fifth resulting images were suppose to be the ones most similar to the input.

Chapter 5

Conclusions and future work

In this work we proposed a CBIR system based on keyword-searches that starts with an image chosen by the user and is then able to find similar images and present them ordered by similarity.

The system leads to satisfactory results for many user-supplied images, and therefore (at least in those cases) it gives very satisfactory results. However, for certain other images, it may perform sub-optimally. In the cases where less than amazing results are obtained, we tried to find explanations and ideas that may help to improve the system, should someone be willing to invest some effort in this in the future.

The preliminary analysis of the image quality has core importance for any CBIR system. In our approach, we also ignore any image that seems to have some content in the borders. This also implies a built-in limitation to approximately constant backgrounds. Instances of an object in dynamic, vibrant backgrounds must be taken into consideration too, and may expose a limitation of our approach. We also admit that it could lead to images being wrongly discarded as not having enough quality to be considered in further processing. These are issues to be solved.

As we work with gray-level images, we only segmented images to detect edges with the Canny detector. However, objects are recognized by humans also by the contrast of colors in the image. Regarding this, working with color images and with a segmentation process like the *JSEG segmentation* [18] may help in improving our system.

The feature-extraction process deals with really simple statistical features that may not be the best ones to represent any object in the world. A lot of research cited in the Chapter 1 considered this matter, and suggested certain feature combinations as the best ones to characterize an object. Obviously, a lot of confusion exists in trying to solve this issue because, as we have said before, there is a *semantic gap* between how to computationally represent images with low level features in order to achieve the efficiency with which humans form semantic concepts to recognize objects. In some of the previous works, the main goal was to obtain a *signature* for each object that would represent it in any image by learning the centroid of all the feature vectors processed when searching for it over all the images of the database. We assume that objects may change color, texture and to a certain extent shape, and still be easily recognized by humans as the same object, despite all the variations. A strategy that pretends to represent an object with a signature, which amounts to a definition, is doomed to fail — because two objects with different low-level features can be in both cases be perfect representations of the same object.

The first future step to try to improve our system may depend on different features, like MPEG-7 shape and position descriptors, which can be used to compose the feature space used to create the image signature. With "histogram analysis, information about spatial configuration is ignored". A possibility is to automatically detect which features would better distinguish it from other images when analyzing an image. It would be assumed that similar images would have the same differential combination of features and the similarity measure process would take it into consideration to obtain better results.

A curious idea, shared by [9] and [5], consists in rejecting systems that do not consider, in their similarity measuring process, the similarities among the target images. Moreover, in [9], relationships between target images that are considered similar are built. This is an interesting idea for future improvements of our own system.

Actually, there are two big systems that, recently, started to yield the possibility of finding similar images to the ones obtained from image retrieval, which is what we are trying to achieve. Those systems are Google Similar Images (`http://similar-images.googlelabs.com/`) from Google and Bing(`http://www.bing.com/`) from Microsoft. Obviously, these systems benefit from their large database of well tagged images and have the management and processing of those tags as the main tasks to obtain good results. They are still not perfect but the results can be impressive depending on the image that the user is searching for. This varying performance (depending on the given image and the richness of the database) is familiar to us.

These systems have a well elaborated graphical user interface that helps reaching better visual approval from the users. Our system does not have any interface developed, yet. A nice future work would be to develop a visually attractive interface for our system, so that the results achieved can be easily demonstrated. As soon as the GUI is stable, an innovation can be introduced: after obtaining the list of images similar to the one picked up initially, the user would be allowed to transform the conditions of the image selected. For example, if the user chose an image of a desk chair with a fully white background, it would be possible for the user to rotate, scale or move the chair within the image. This could be based on guesses inspired by the images previously considered as similar.

Thinking in very high level terms, this work, as all similar works published until now, are in some way, contributing to different aspects of the problem of recognizing objects with computers.

Objects are present everywhere and we recognize them almost unconsciously, with seemingly little effort. Objects could have different properties like color, shape and texture. Objects appear in different places, in different sizes and in different view points (from the front, from the back, from the side). Objects may appear partially obstructed, blurred, translated and/or rotated. Moreover, it is unlikely for us to see the object in the same conditions twice in our lives because every time our eyes move, the pattern of neural activity changes. Still, we are able to recognize them, associating each one with a name that we learned to identify it. Recognizing objects by computers is a major field of study in Computer Vision and the authors of this work are hopping to contribute some day with the knowledge acquired from its development.

Bibliography

[1] Y. Zheng, J. Yang, and Y. Zhou. Unsupervised segmentation on image with jseg using soft class map. *Intelligent Data Engineering and Automated Learning IDEAL 2004*, Volume 3177/2004, Pages 197-202, October 2004.

[2] Y. Rui, T. Huang, and S. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, Volume 10, Issue 1, Pages 39-62, March 1999.

[3] Y. Chen and J. Z. Wang. A region-based fuzzy feature matching approach to content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 24, Issue 9, Pages 1252-1267, September 2002.

[4] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, Volume 5, Pages 913-939, December 2004.

[5] Y. Chen, J. Z. Wang, and R. Krovetz. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE Transactions on Image Processing*, Volume 14, Issue 8, Pages 1187-1201, August 2005.

[6] R. Datta, W. Ge, J. Li, and J. Z. Wang. Toward bridging the annotion-retrieval gap in image search. *Proceedings of the 14th annual ACM international conference on Multimedia*, Pages 977-986, 2006.

[7] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, Volume 40 , Issue 2, Article 5, April 2008.

[8] J. Laaksonen, M. Koskela, and E. Oja. Picsomself-organizing image retrieval with mpeg-7 content descriptors. *IEEE Transactions on Neural Networks*, Volume: 13, Issue 4, Pages 841-853, July 2002.

[9] K. U. Barthel. Improved image retrieval using automatic image sorting and semi-automatic generation of image semantics. Technical report, FHTW Berlin, Germany, 2008.

[10] K. U. Barthel, S. Richter, A. Goyal, and A. Follmann. Improved image retrieval using visual sorting and semi-automatic semantic categorization of images. Technical report, FHTW Berlin, Germany, 2008.

[11] K. U. Barthel, S. Mammani, and N. Wyatt. Automatic image sorting using mpeg-7 descriptors. Technical report, FHTW Berlin, Germany, 2008.

[12] K. Tieu and P. Viola. Boosting image retrieval. *International Journal of Computer Vision*, Volume 56, Issue 1, Pages 17-36, 2004.

[13] E. Izquierdo and D. Djordjevic. Using relevance feedback to bridge the semantic gap. *Journal Desconhecido*, Volume 3877/2006, Pages 19-34, February 2006.

[14] A. Marakakis, N. Galatsanos, A. Likas, and A. Stafylopatis. A relevance feedback approach for content based image retrieval using gaussian mixture models. *IET Image Processing*, Volume 4132/2006, Pages 84-93, September 2006.

[15] M.M. Rahman, B.C. Desai, and P. Bhattacharya. An interactive and dynamic fusion-based image retrieval approach by cindi. *Advances in Multilingual and Multimodal Information Retrieval*, Volume 5152/2008, Pages 657-664, September 2008.

[16] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis. Region-based image retrieval using an object ontology and relevance feedback. *EURASIP Journal on Applied Signal Processing*, Volume 2004, Issue 6, Pages 886-901, June 2004.

[17] Y. Liu, D. Zhang, and G. Lu. Region-based image retrieval with high-level semantics using decision tree learning. *Pattern Recognition - The Journal of the pattern recognition society*, Volume 41, Issue 8, Pages 2554-2570, August 2007.

[18] Y. Liu, D. Zhang, G. Lu, and W. Ma. A survey of content-based image retrieval with high-level semantics. *The Journal of Pattern Recognition Society*, Volume 40, Issue 1, Pages 262-282, January 2007.

[19] E. Spyrou, H. Le Borgne, T. los Mailis, E. Cooke, Y. Avrithis, and N. O'Connor. Fusing mpeg-7 visual descriptors for image classification. Technical report, National Technical University of Athens, Greeece and Dublin City University, Ireland, 2005.

[20] L. Baboulaz and P. L. Dragotti. Exact feature extraction using finite rate of innovation principles with an application to image super-resolution. *IEEE Transactions on Image Processing*, Volume 18, Issue 2, Pages 281-298, February 2009.

[21] Mukul V. Shirvaikar. An optimal measure for camera focus and exposure. *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory*, Pages 472-475, 2004.

[22] N. Nourani-Vatani and J. Roberts. Automatic camera exposure control. Technical report, CSIRO ICT Centre, Australia, 2008.

[23] Michael Reichmann. The luminous landscape. 2007.

[24] David W. Jacobs, Peter N. Belhumeur, and Ronen Basri. Comparing images under variable illumination. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Pages 610-617, 1998.

[25] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Prentice Hall, 2nd edition, 2002.

[26] S. E. Umbaugh. *Computer Vision and Image Processing.* Prentice Hall, 1999.

[27] M. Nixon and A. Aguado. *Feature Extraction and Image Processing.* Elsevier Linacre House, 2008.

[28] R. Gonzalez, R. Woods, and S. Eddins. *Digital Image Processing Using Matlab.* Addison-Wesley Publishing Company, 2004.

[29] T. Kohonen. *Self-Organizating Maps.* New York : Springer, 1997.