Faculdade de Engenharia da Universidade do Porto

Mestrado Integrado em Engenharia Informática e Computação

Image Retrieval for E-Government Documents

MIEIC Project Report 2007/2008

Nuno Miguel das Neves Machado

Supervisor at FEUP: Prof. Ana Paula Rocha

Responsible at UCB-ISS: Prof. Dr. Norbert Kuhn

April of 2008

To my family and friends
for supporting me all the time

Abstract

This document describes the implemented techniques and procedures regarding the field of Content Based Image Retrieval on databases on a project within the FABEGG (Framework zur Abbildung und Beschleunigung von nationalen und internationalen E - Government – Genehmigungsverfahren) project at Fachhochschule Trier (FH Trier), Umwelt-Campus Birkenfeld - Institute for Software Systems (UCB-ISS), Germany, through the Project from Mestrado Integrado em Engenharia Informática e Computação (Integrated Master on Informatics and Computing Engineering), at Faculdade de Engenharia da Universidade do Porto (Engineering College of University of Porto).

It was observed during the past years an increase in the number of applicable methods regarding the field of Content-based Image Retrieval. The idea is always using the base concept of extracting image features as well as using this type of information to understand the contents of an image, relating such contents to large scale image databases. Generally these image features regard aspects of images such as colour, texture and structure.

The FABEGG system is a set of tools that provide accessibility to users while interacting with the German government's document management system. Such interaction is aided using applications developed in domains such as Speech Recognition or Image Retrieval. In this document it will be described how some of the procedures and techniques for Image Retrieval were implemented regarding the image comparison by using total or partial content of an image, by analysing colour and structure features in the context of document template retrieval at the FABEGG system.

In order to achieve this main goal, there are three important stages: preprocessing stage, image sampling and feature retrieval stage and comparison stage.

Because pre-processing is an important part of image retrieval, several well known pre-processing techniques within the image retrieval subject will be described and discussed.

Then the next step's goal is to get concrete information from the given visual content, and to make this retrieved information able to be compared on the last stage of development.

The last stage is a direct comparison of the retrieved feature lists, whose results are dependent on the combination of the last steps.

Finally test results will be presented with further discussion on the implemented procedures for all stages described above.

Acknowledgements

I would like to thank the Institute for Software Systems at Fachhochschule Trier, Umwelt-Campus Birkenfeld for participating in the Project/Dissertation program at Integrated Master in Informatics and Computing Engineering. I would like to express my most gratitude to Prof. Dr. Norbert Kuhn who was my supervisor during the project and gave me this opportunity. I would also like to thank Stefan Richter and Michael Schmidt who also gave me support on this project. I would like to thank Prof. Ana Paula Rocha from Faculdade de Engenharia da Universidade do Porto for supervising me and the work on this report document. Finally I would like to express my thanks to my family and friends for supporting me during this time.

Table of contents

List of figures

List of acronyms

FH Trier – Fachhochschule Trier

OCR – Optical Character Recognition

RGB – Red Green Blue

UCB-ISS – Umwelt-Campus Birkenfeld – Institute for Software System

1 Introduction

This chapter begins by introducing the FABEGG system and its context in Computer Vision subject, as well as the purpose of the implemented project within the FABEGG system, described on this report document. The last section presents how the contents are organized on the remainder of this report.

1.1 The FABEGG project

Computer Vision has been under significant research efforts during the past decades. Such efforts led to an increase on the number of application fields. Nowadays, Computer Vision has great impact on fields like medicine, industry, military and autonomous vehicles construction.

The FABBEG system is a set of hardware and software tools that is continuously being developed and improved at the Fachhochschule Trier (FH Trier), Umwelt Campus Birkenfeld - Institute for Software Systems (UCB-ISS).

Recently, some activities involving governmental entities can be done through the internet. However, printed documents still play a significant role on many other activities. The main objective in FABEGG system is to enhance the accessibility methods to governmental authorities, done by users with reading disabilities, providing them a much better access to activities that use paper documents [1]. This help is provided through a system based on many technologies including Computer Vision, more specifically, in the field of Image Retrieval.

Generally, activities that involve the government authorities use a basic workflow: the government users provide the documents in some way, then the citizen users would use documents, such as forms, to access government services. These document types require user interaction, in most cases, to fill the required form fields. Once this interaction is finished, the documents would be sent back to the government by the users.

The FABBEG system comes as a solution to enhance the interaction between visually impaired users and the government, aiding all the necessary operations throughout this workflow by implementing an enhanced human-computer interface for citizen users and a database-based system for document management.

Implementing the FABEGG system on this workflow would eventually change the behaviour of both citizen and government user types. Therefore, for each user type there is two different components with different functions that work together as a client/server application. The government user uses a component called FABBEG forms front end [1], that is used to manage a database and/or repository systems. The government users will use this application to load documents to the repository and provide metadata information that is associated to the respective document. Some of the information is also given by analysing the image based on OCR (Optical Character Recognition). This information at the database would help citizens to interact with the client application called FABEGG user front end [1]. An user captures the printed document with a proper camera that is designed for the purpose, and the application would recognize the document's content using OCR so that a query can be sent to the server in order to obtain the information related to the template that was inserted by the government user at the repository. This information is then used to give aid to users, (e.g. by providing tips on form fields or helping the reading of a document with speech synthesis). The user front end also provides other kinds of activity related to the document image using image manipulation such as foreground and background colour change and image enlargement.

The government activities workflow would have to be adapted so that supporting this conditions would be possible. Figure 1 shows the workflow that would support the FABEGG system and the government activities.

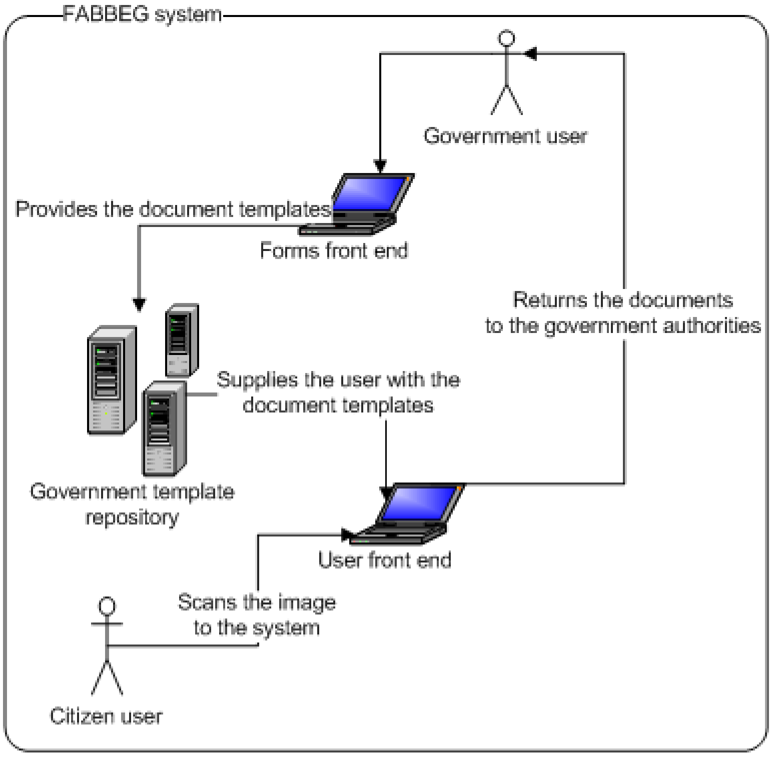


Figure 1: Workflow on FABEGG system

Image Retrieval plays a very important role in the workflow ilustrated on figure 1. However, most of the system's limitation is due to the use of a proper camera that captures the whole document at once and the image recognition being based on OCR. Recent implementations on the FABEGG project would allow other input devices such as scanners or common cameras, but because the document isn't entirely captured like in the proper camera, it wouldn't retrieve the information for the whole document, but only for the captured information. The second limitation is that OCR-based image recognition only recognizes text in documents, making that situations with image on documents (e.g. logos, diagrams, pictures), become invalid right from the start of the user front end workflow.

1.2 Image Retrieval for E-Government Documents and the FABEGG project

Because the current Image Retrieval system relies on external information to successfully extract the information on documents, it is necessary to create new methodologies to overcome this problem.

In order to enhance the Image Retrieval functions at the FABEGG system, it was necessary to make a deep research on the existing solutions and to make a system capable of performing comparison among images, analysing the images by content only, and therefore, evaluating similarities. In addition, it became necessary that images should be compared by analysing partial content. This means that, for example, an user front end image may be a part of a template stored at the repository.

Considering the introduced ideas for a new Image Retrieval system, the main goals proposed for this project are to implement an Image Retrieval system capable of:

- Retrieving images regardless of external informations such as metadata or OCR information, such as character font information;
- Recognize images with common content such as images with the same logo, regardless to the absolute position, orientation and size of this content in different images.

Achieving these allows the FABEGG system to be able to search other content besides text information and furthermore, allows that document information can be retrieved by giving the partial content of an image.

This project required other research efforts such as pre-processing techniques, image feature extraction and comparison of two images by considering their extracted features.

It was implemented under Microsoft Visual Studio 2005 Express Edition, using the Aforge.net framework for image processing by Andrew Killinov [URL02].

1.3 Organizations of subjects on this report document

This document consists of 6 chapters. Chapters 1 gives an introduction to the general subject and explains the context of the project. A closer look is taken on chapter 2, where all concepts are explained, required to understand the following chapters.

Chapter 3 discusses the pre-processing techniques used in this project. It also includes a discussion for each one where advantages and disadvantages are pointed out.

Chapter 4 and 5 explain proposed methods for colour and structure feature extraction. These chapters also explain how similarity retrieval is made for each case. Like in chapter 3, the results will be discussed for each one of the used methods.

Finally, chapter 6 concludes by reporting the final results obtained when using the implemented solutions described on chapters 3, 4 and 5. It also discusses future directions of work. In addiction, the project results can be seen on the annex section of this document.

2 General concepts and definitions

2.1 Introduction to Image Retrieval on Databases

Over the last decades significant research has been made on the field of digital image processing and analysis. As the usage of digital image database oriented systems grew, there was an increase in development of image processing and analysis procedures by the research community. But because there is a wide range of techniques and procedures for this purpose, as well as a significant number of existing Image Retrieval systems, there is a lack of consensus among the research community of which procedures are the best to be implemented.

Efforts made to enhance Image Retrieval research focused on three different approach types [19]:

- Metadata Image Retrieval systems;
- Semantic Image Retrieval systems;
- Content-based Image Retrieval systems.

Most traditional systems add metadata as information on an image for database query on Image retrieval. This method implies that the information is given in form of text. Can either be by manually describing an image, by giving text information or by automatic annotation where the image information is automatically classified by proper algorithms, mostly used for database sorting and organization.

Semantic Image Retrieval focus on higher levels of user inputs by using neural networks and visual content for image descriptors. This type of Image Retrieval is usually used in learning systems. It is the most recent attempted approach on this field of study.

Content-based Image Retrieval systems are the most widely used for implementing an Image Retrieval system. This type of retrieval focuses on using visual information present on images such as colour, texture, structure and spatial relations on the image. These content types are identified on images and therefore, are converted into simpler and easily comparable information that are named as feature vectors.

Within the image feature extraction techniques, there are several approaches. Therefore the representation of feature vectors isn't the same for each one of these approaches. However, in most systems this representation is done by calculating a set of values using such extraction techniques. What may vary in feature vector representation is the meaning of each number, range of values, size of vector, etc.

The implemented project within the FABEGG system uses content-based procedures only, so this technique will be emphasized here.

Figure 2 describes a typical architecture for a Content-based Image Retrieval system [13].

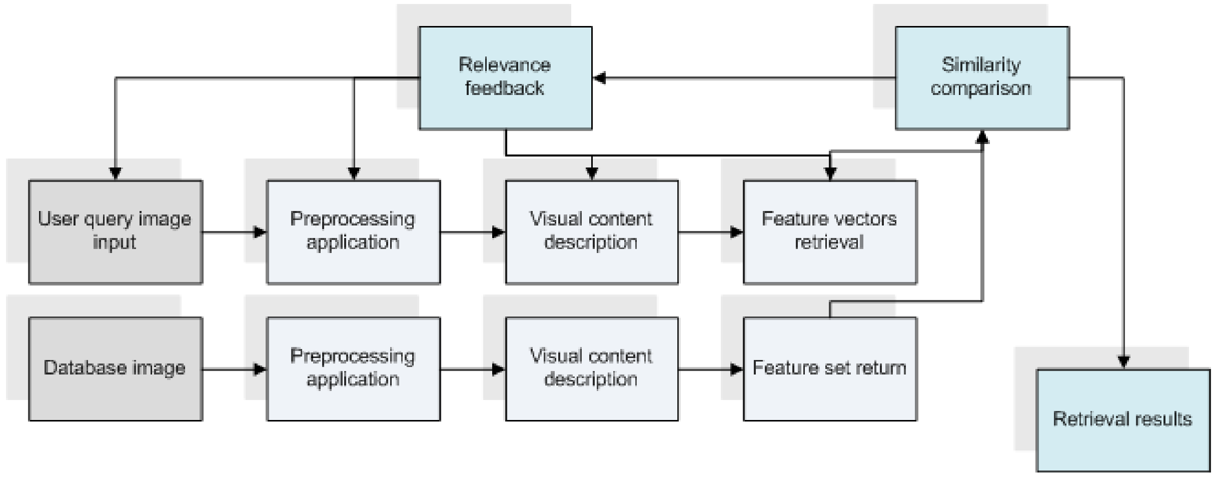


Figure 2: Typical Content-based Image Retrieval architecture

Similarity between two images will be acceptable if these images are considered to have similar content. The metric used should be defined on the beginning, as there are also several procedures for similarity evaluation.

In most systems, relevance feedback is given. The idea is to give report on the retrieval accuracy, generally specified by mathematical formulas.

2.2 Content-based Image Retrieval

Content-based Image Retrieval is a subject of Image Retrieval that aims to research and develop on image recognition through image features based on visual content.

Content-based Image Retrieval was originated in the 1990s, when attempts of automatic image retrieval from databases procedures were starting to appear. It was initially thought and used to overcome the problems previously found on systems created that used metadata information to describe images. Because metadata requires manual human description, it became unacceptable that such method could be implemented on large scale databases. On the other hand, textual information may be susceptible to human error [20].

There are two ways of retrieving information from an image through Content-based Image Retrieval: using visual information and high level information [7].

An ideal way to describe an image is said to be by using a high-level approach to retrieve the content on images, a case where content-based and semantic based systems are combined. On this scenario, the systems receives high level input such as “I want pictures of yellow cars”, that is then used to retrieve the visual information through analysing its semantic meaning. On this project, this type of retrieval isn't used, so it will not have further analysis.

Unlike previously implemented solutions based on information given by using text, Content-based Image Retrieval uses visual content to describe the image. Because the main concept of this retrieval type attracted more potential users of many more disciplines, many solutions were developed on several directions. These solutions approached other related fields such as statistics, pattern recognition, signal processing, and Computer Vision [20].

Despite of the existing different solutions, Content-based Image Retrieval systems focus on the same content types that are present on any image. These content types are also the existing feature types on this kind of retrieval, which can be enumerated as [2] [3] [4] [5] [7] [20]:

- Colour features: relates the image content to the Red-Green-Blue (RGB) space.
- Texture features: it is an important property of images. Features are retrieved by analysing image patterns either by statistic or by image structure.
- Structure features: relates to the shapes that can be recognized on an image.
- Spatial relation features: unlike other feature types, the spatial relation information relates to relative information of the image contents, and does not describe the contents itself.

Colour retrieval on images refers to computing colour histograms using image pixels values. These pixel values are visually shown as colour on the image.

Texture refers to recognition of patterns within an image. Segmentation is also applied to distinguish different texture sets. It implies a more complex set of methods to compute a texture than by just using RGB colour space. Furthermore, its representation is always challenging to perform, and not all image types contain well defined textures (e.g. documents).

Structure on images relates to shape identification. Approaches that use structure identification require the use of edge detection algorithms or segmentation. Because this type of pre-processing is still challenging to obtain good results that fit all cases, it requires human-made decisions to select the correct parameters.

Collecting the described features is done by view based analysis without any assumption. The success or insuccess factors are therefore given by how "intelligent" the implemented systems are. Development teams also prefer to adopt combinations of content type and its methods of extraction to enhance the chances of success. In fact, content can be better described if the features are considered depending on each other [2].

In the current project, for the purpose of document retrieval, it was assumed that printed documents have no texture because paper texture is not well defined enough that such information could suffice a reliable retrieval, and so only colour and structure are considered. The next sections of this chapter present some information on image processing and content description used for this project.

2.3 Image processing

Being one of the many subjects of signal processing, image processing is important to help to identify image features on Image Retrieval. Generally speaking, image processing can be understood from a higher level perspective as a set of operations on an image.

The most common image processing techniques are the following [14] [15] [18]:

- Geometric transformations: includes transformations such as rotation and resizing;
- Colour conversions: perform conversions on the image colour space (e.g. RGB to grayscale);
- Colour corrections: are all transformations for brightness and gamma correction;

- Noise removal: a specific part of image processing that removes unwanted elements of an image;
- Image segmentation: a technique that divides an image into regions or clusters.
- Edge detection: detects the pixel values where a sharp change on the image's brightness occurs.

Many of these image processing techniques can be found on most digital image editors and are therefore, widely used.

On this project, these pre-processing methods were used in order to help identify image features and as consequence, contributed to the success (or insuccess) of similarity results.

2.4 Image content descriptors and feature vectors

Considering the process that leads an input image to the retrieval of similar images on databases, on an Image Retrieval system, it is necessary to apply pre-processing to the input image and retrieve the visual content descriptors from the resultant image.

As mentioned before, these descriptors may result from different visual content types: colour, shape, texture and spatial relationship information. Feature vectors are a specific representation of these descriptors. Generally it is a kind of representation that is sorted in a list of numerical values that refer to some particularity on the image, resultant from the application of methods related to the image feature type. For example, a feature vector based on spatial relationship may be the distances among the objects belonging to a scene on the image. A feature vector can also have different types [2] [3]. For example, a feature vector can be retrieved from an image containing information about the amount of red pixels and the relative distances among these pixels. These values are represented as simple as possible in order to make it easy to compare different vectors. In addition, feature vectors' values should have a very specific meaning before being used. The meaning relates to the feature type and should have the same representation in all comparable images (because structure with colour cannot be compared, only structure with structure).

In any feature type the descriptors should all follow the same quality rules, in order to retrieve the ideal results:

- A good visual descriptor should be invariant to possible accidental variance, such as luminance and presence of noise;
- It should be invariant to certain image transformations such as rotation and scaling. It is important that image quality can be preserved under these circumstances;
- An image descriptor should be global or local:
    - Global descriptors refer to the global image content;
    - Local descriptors are related to specific elements on the image such as segmented regions or identified shapes.

Descriptors from visual content are retrieved using appropriate methods, and so the result depends entirely on it, as a metric is not globally defined by the content itself. Therefore, obtaining such conditions mentioned before also depends on the methods used.

However, quality of local descriptors depends almost exclusively on the success of the segmentation process. Because a good segmentation is still a challenge on Image Retrieval, good success cases are hard to obtain in general cases.

It is also important to set the difference between global and local descriptors. Because the goal of this project was to compare not only the whole image but to find parts of images within another image, both approaches were implemented for visual descriptors.

Further on chapters 4 and 5, the implemented methods for the purpose of partial image detection are discussed.

3 Image Pre-processing techniques

3.1 Introduction

This chapter describes the implemented pre-processing techniques and its results, advantages and disadvantages on the system and general usability [13] [14] [15] [16] [17] [18].

Pre-processing is one of the most important steps on the image recognition procedure that was implemented for this project. The following sections describe the algorithms used and furthermore, the results of its application are discussed. Some of these pre-processing methods were implemented using Aforge.net, with the exception of the noise removal procedures.

The pre-processing techniques described on this chapter are:

- Image resizing;
- RGB to Grayscale conversion:;
- Contrast correction;
- Binary conversion;
- Digital filtering;
- Mathematical Morphology operations;
- Canny Edge Detector;
- Independent noise removal;
- Noise removal by statistically defining noise;

3.2 Image resizing

Resizing became useful because it is a fast solution to problems where the absence of noise and quality of edges are relevant. There are several well known algorithms for this procedure.

Effect of resizing is a result of interpolation between pixels. The idea is to reach the best results for each pixel based on the adjacent pixels. It works on two directions (enlargement and reduction of image size). There are two types of interpolation algorithms: non-adaptive algorithms and adaptive algorithms [16].

Non-adaptive algorithms treat all pixels equally whereas adaptive methods consider some image detail like edges and texture.

Non-adaptive algorithms

As non-adaptive algorithms, these are the most widely used: nearest-neighbour, bilinear and bi-cubic algorithms.

The *nearest neighbourhood* is the simplest to be implemented as well as the fastest. However, it produces the worst results. It considers only the nearest pixel and as a result, it makes the effect of changing the pixel's size when the image is resized, meaning that no interpolation was applied.

The *Bilinear* algorithm considers the four neighbours on the diagonal directions, and the centre pixel is calculated by simply getting the average from these neighbours, in a situation like the one shown in figure 3.

The *Bicubic* algorithm is the most used even in some well known image editors. It has the best results by following general criteria. Therefore, this method was selected to be implemented in our project. It works similarly to bilinear but instead of 4 neighbours, it considers 4 groups of 4 neighbours in each diagonal direction.

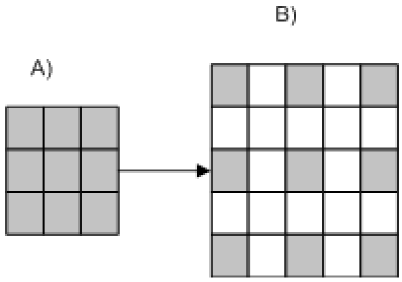


Figure 3: A) Original section of an image B) The section before pixel interpolation, the white spaces represents the unknown pixels

Figure 3 shows an example of a section of pixels being resized to bigger dimensions. The white squares represent the unknown pixels values for the final result. Depending on the algorithm chosen, these values will be smoother or rougher.

Adaptive algorithms

Adaptive algorithms are mostly proprietary algorithms and therefore are designed for unique purposes. Examples of adaptive algorithms are: Qimage, PhotoZoom Pro, Genuine Fractals.

Only the non-adaptive algorithms were able to be tested. The *Bicubic* interpolation was the one showing the best results. Despite of the slightly bigger time consumption when compared to the other algorithms, the quality in results is worth it. In addition, the usage of this procedure for resizing an image to smaller dimensions brings several advantages: because some of the detail is lost, intensive noise spots are eventually removed as well, furthermore, any further pre-processing is done much faster.

3.3 RGB to Grayscale conversion

An image represented in the RGB colour space from an user's point of view offers more detailed information, especially when colour feature extraction algorithms are applied. However, in most cases certain operations require simpler representation.

Internal representation of an image in the RGB space implies 3 channel usage. When imagining the pixel values represented on a matrix, it requires 3 values for each pixel, for one of the 3 colour channel components. Grayscale conversion transforms a given RGB image into another image at the grayscale space using 1 channel only. The first main advantage besides the simpler representation explained is that an operation on the grayscale space takes

3 times less than in RGB space. In addition, this step is necessary for structure feature extraction algorithms [14].

The most basic algorithm for grayscale conversion is done by simply calculating the average value of the 3 channels for each pixel. The resultant value is the grayscale value for the single-channel colour space.

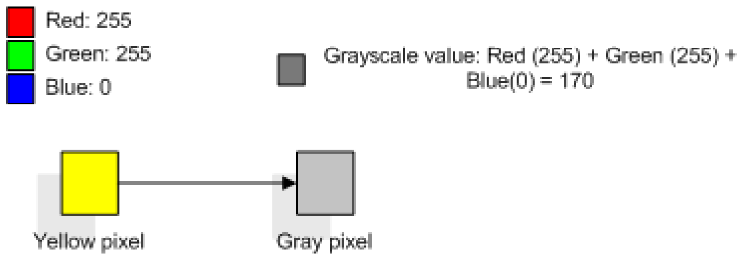


Figure 4: RGB to Grayscale conversion

Figure 4 shows an example of a basic RGB to grayscale conversion for one single pixel. It is to be noticed that both RGB and grayscale values are between 0 and 255.

Enhancements to this procedure have been made because of luminance issues, there are other implementations based on the base algorithm previously explained. By applying the base algorithm for grayscale conversion, equal weights to each RGB component are given. The result is that eventually some brightness details are lost. By giving different weights for each RGB component makes the resultant luminance more realistic.

An algorithm based on the BT709 norm, uses the following weights for the RGB space.

- Red: 0.2125;
- Green: 0.7154;
- Blue: 0.0721;

There are other algorithms similar to the algorithm based on the BT709, but with different coefficient values such as the R-Y and Y algorithms. When applying the BT709 norm to the example illustrated on figure 4, the result is obviously different:

Figure 5 shows some results of the application of the BT709 norm based algorithm.

Figure 5: RGB to grayscale conversion using BT709 norm.

3.4 Contrast correction

The chosen contrast correction processing operation can have a better understanding when thinking of the next steps of edge and shape recognition. Because the contrast correction enhances the sharp changes on brightness, its application prevents the loss of information.

As mentioned on the previous section, both colour or grayscale image can be seen as matrices of values on its internal representation. However these values are not mapped directly into visual content and the image has to be passed through a lookup table for mapping brightness values from the image's intensity values. Thus, the lookup table will define the intensity values on the image that are to be displayed on a monitor [13].

Brightness and contrast correction are achieved through lookup table manipulation, and are perhaps the most common purpose on image processing.

Lookup tables for brightness and contrast can be represented in functional form by a linear function such as: $g(u) = k\,u + b$.

Figure 6 illustrates a good approximation of the linear functions for brightness and colour lookup tables.

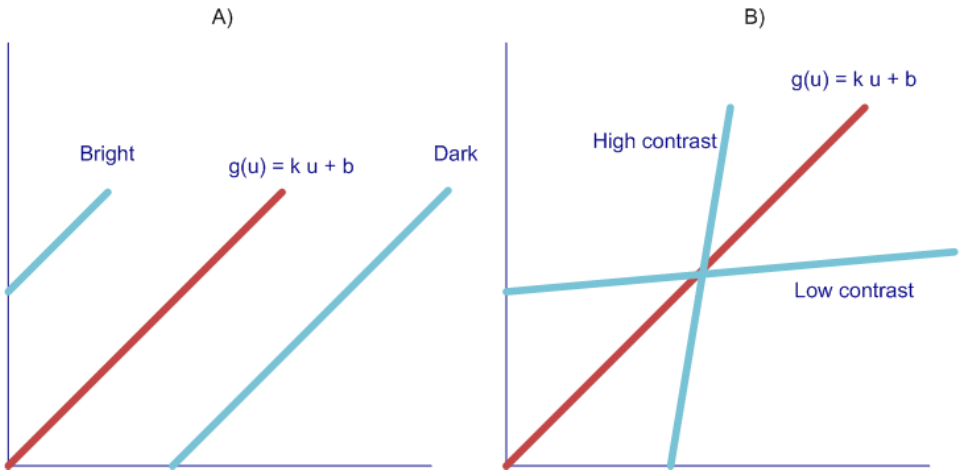


Figure 6: A) brightness lookup table linear function B) contrast linear function lookup table

Brightness can be increased and decreased by changing the parameter *b* on the lookup table function. By doing so, luminance values on the image are changed to a darker or brighter result. For the contrast it is also necessary to change the variable *u* to get higher and lower contrast intensity values. Some effects are to be notice on this case. If $b \rightarrow -\infty$ and $u \rightarrow \infty$, the contrast is theoretically the highest. However, drawing the function to the opposite direction makes the colour mapping to be inverted (e.g. white is drawn as black and black is drawn as white).

In this project only the contrast correction component was considered, it does not require any user's parameters or custom configuration, with the exception of the number of iterations. It is an important pre-processing technique that enhances the quality of the edge detection.

Figure 7 shows an example of the application of the contrast filter.

Figure 7: Example of one iteration using the contrast correction method. Contrast on the image on the right is higher than the one on the left

3.5 Binary conversion

Generally speaking, binary conversion is referred to a particular case of an image in the grayscale colour space. In fact, a binary image is a grayscale image with the maximum and minimum values only [13] [15].

Binary images are mostly used on shape recognition and the main objective of this procedure is to exclude the background, leaving only the regions of interest. Setting the difference between background and foreground regions is then, the main challenge present on this process.

The most basic algorithm for this procedure is to set a threshold value that will be used to set the white and black pixels on the image. The values below the threshold number will be equal to 0, and the values above it, will be increased to 255.

The basic algorithm for binary conversion implies a simple, yet challenging issue: the threshold value requires user intervention, and it is easy to realize that this may lead to problems on the next stages of image processing because the same threshold value isn't always the best in all cases. Thus, it becomes necessary to add some intelligence in order that the system can select the correct values by evaluating the case.

This is usually performed by calculating an adaptive threshold using statistic methods. Such methods may include analysing sections of pixels to retrieve the ideal grayscale value for the threshold.

Figure 8 illustrates some examples using different threshold values on the same situation.

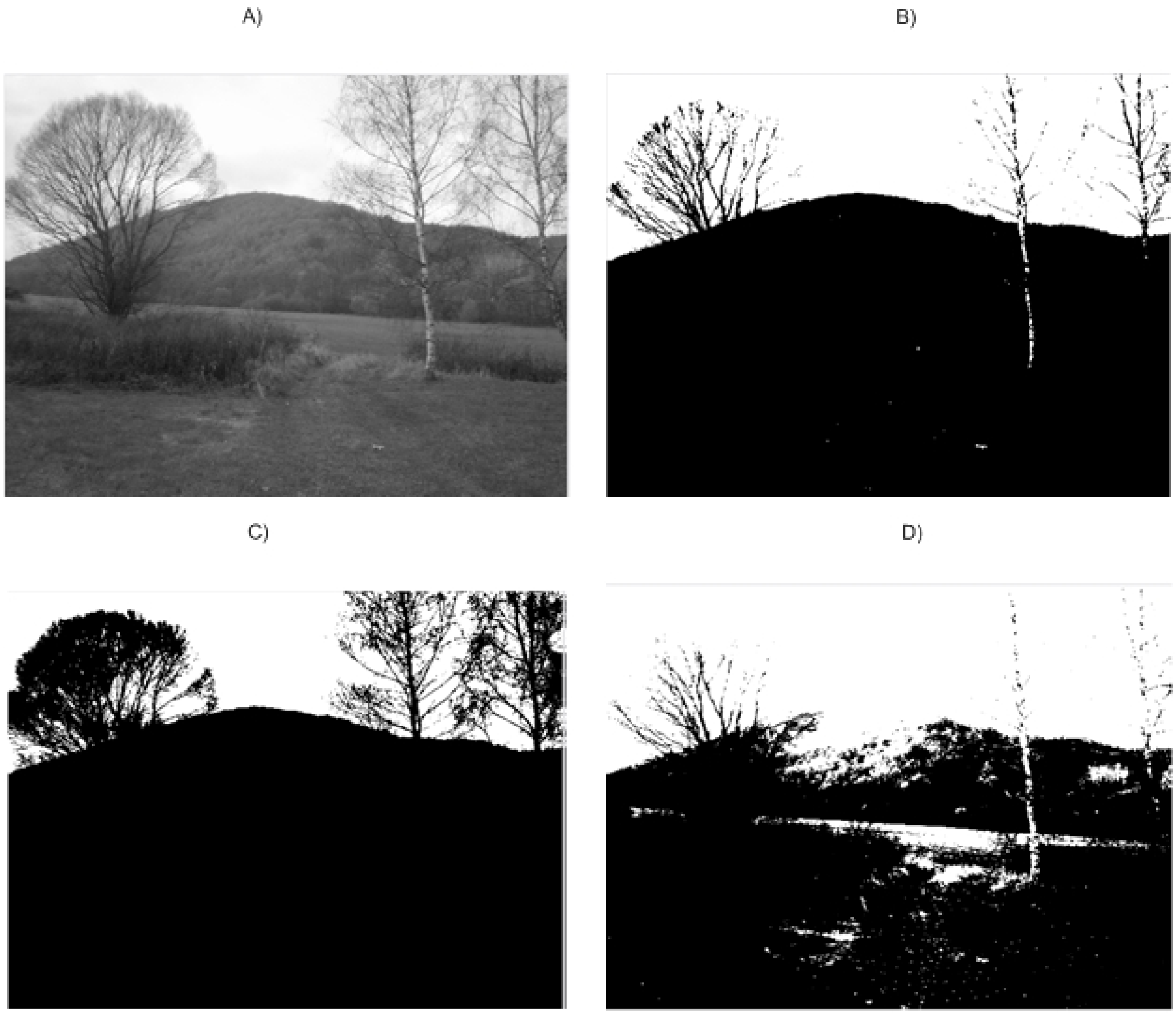


Figure 8: A) Original image in grayscale space B) Binary conversion using a threshold value of 128 C) Binary conversion using a threshold value of 200 D) binary image using a threshold of 100

On the examples illustrated above can be seen the effects of choosing the correct and incorrect threshold values. For this particular case the best results are achieved on sample C), however this value isn't always the best for all cases. An adaptive threshold finder could be implemented by analysing the luminance intensity values or RGB colour values, for instance.

For the document retrieval, it suffices to use the basic binary conversion method because the background and foreground are well defined. In most cases for documents, the threshold value of 128 is sufficient.

3.6 Digital filtering

Noise reduction is one of the biggest challenges on image processing and image recognition. Despite of the existing solutions it is still difficult to make computer systems understand the concept of noise. Digital filtering is perhaps one of the most common ways to remove noise without the need of using a method to detect noise. On the digital filtering field Mean, Median and Gaussian filters are the most used [15] [18].

It is referred as image noise to unwanted sections of pixels or colour details that are usually generated in the input device (e.g. Scanners, cameras) that are usually unavoidable.

Because the noise becomes a part of the image at the initial input, it is impossible to remove it without losing some of the information. Some of these removal techniques were already referred on this chapter such as the Mean, Median and Gaussian filtering, that are good at removing noise on RGB and grayscale colour spaces and preserving the image detail reasonably.

On this project, it was only implemented the Median and Mean filters. The Gaussian filter was not applied directly as a single pre-processing operation but it is used on edge detection described on the next section of this chapter.

Gaussian filter has the effect of smoothing an image. It is applied by using a 2-D isotropic convolution operator This filter is called as convolution operators because it is applied in the same method of standard convolution operations. For this purpose it is necessary to define a matrix of values named as kernel. The Gaussian filter can be defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where σ is the standard deviation and *(x,y)* defines the mean. Because the pixel values on images are given in discrete values, it is necessary to calculate a discrete representation for the Gaussian function as well, only then it is possible to perform a convolution. Furthermore, the discrete convolution kernel is represented as a matrix.

Smoothing an image with the Gaussian filter can remove both detail and noise. However, the dimensions of the effect still requires user's input for the standard deviation and mean values that define the kernel size. Higher values imply higher smoothing degrees.

The Mean and Median filters are simpler to be implemented.

The *Mean filter* consists on scanning the image pixel by pixel, from left to right and from top to bottom, considering the neighbourhood for each with a predefined size (usually the dimensions are 3x3). Mean filtering only uses the kernel to define the size of the neighbourhood.

For each neighbourhood, the average value of pixels is then calculated, including the main pixel, and the result is averaged to an integer value.

The image output will then have more homogeneous regions, clearing the rough sections of the image and therefore, removing noise.

Median filter is also a way to remove some of the noise that may present on the original image. It is an alternative to the Gaussian and Mean filters.

When applying the filter, the image is scanned, from left to right and top to bottom, pixel by pixel. For each pixel, the surrounding pixels are selected like in the Mean filter (still using the *NxN* dimensions for the kernel). For each group, the median value is selected and applied to the main pixel.

Figure 9 shows the application of the Mean and Median filter for one pixel by analysing the surrounding neighbourhood.

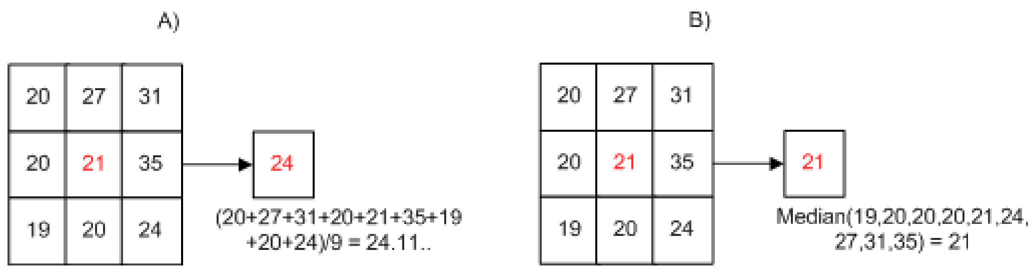


Figure 9: A) Application of the Mean filter with a kernel of 3x3 B) The same example but applying the Median filter.

Experiments while using these filters showed that it can be helpful when working with big and detailed images. Small or resized images may show its edges heavily distorted if they don't have enough detail because some of the detail is already been removed by the resizing process, and therefore the edge quality is more exposed.

On documents, the Median filter revealed more efficient than the Mean filter. The Gaussian filter is used along with edge detection, explained on the following section of this chapter

3.7 Mathematical Morphology operations

Mathematical morphology image processing techniques are operations based on simple mathematical concepts. It is widely used along with binary images, edge detection and shape recognition.

Such operations consist on image foreground and background transformations on grayscale images, but are mostly used on binary images, where generally black pixels represent the background and white pixels the foreground, although this depends on the implementations.

The most basic mathematical morphology operation examples are called Dilation and Erosion. All other implementations on mathematical morphology are based on these two operations [15].

Both Dilation and Erosion operators work similarly to each other. They also use a structure similar to the kernel used on digital filtering, however this structure, named structuring element, is used for different purpose.

Structuring element can be considered as an image sample with small dimensions (usually, the default dimensions are 3x3), and is used by considering each pixel of an input image. For each pixel the structuring element is superimposed to the main pixel. The origin of the structuring element is the pixel in the middle that coincides with the main pixel being analyzed. Depending on the algorithm, different operation are performed

On *Dilation* process, each background pixel is superimposed by the structuring element, as described before, and if at least one pixel on the structuring element coincides with the main pixel on the group, then this pixel is considered to be foreground pixel.

Erosion works similarly, but instead of scanning background pixels, foreground pixels are considered. If all pixels on the image that match the structuring element are foreground, then the input pixel is left at it is, otherwise, this main pixel on the group is considered as background.

Figure 10 shows examples of Dilation and Erosion applications.

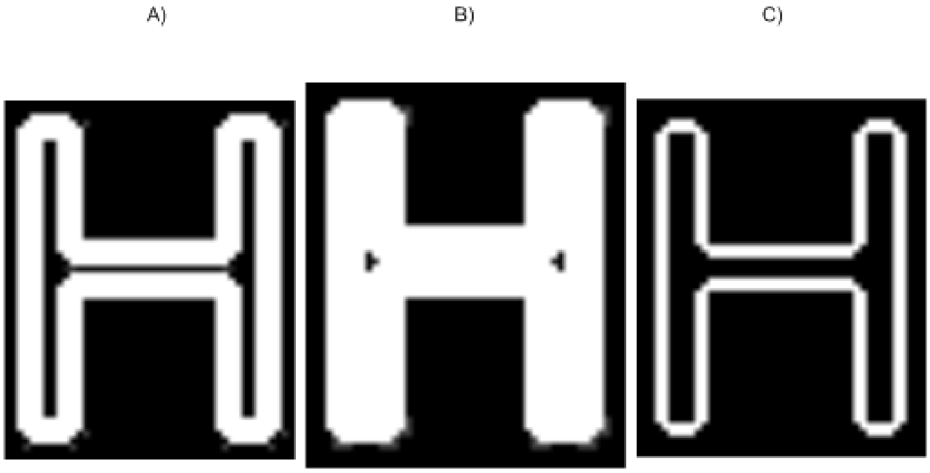


Figure 10: A) Original shape B) Shape after applying Dilation operation using a 3x3 structuring element C) The shape after using the Erosion operation with a 3x3 structuring element.

Other mathematical morphology operations are combinations of these two basic operations or extensions of them. For example, the Opening and Closing filter are sequences of Dilations and Erosions or vice-versa. Other operations like the Hit-and-Miss transformation or the Thinning operations are extensions of Dilation or Erosions.

Closing filter has proven useful on shape recognition since it is capable of closing opening gaps on shapes. The only problem associated with this operation is that it is a Dilation followed by an Erosion using the same structural element, meaning that the user (or the system) does not take control on which edges are to be closed. The closing filter is not controlled, and therefore, is not always the most convenient.

3.8 Canny Edge Detector

The problem of edge detection is, with no doubt, the one with the most importance on shape recognition and structure feature extraction. During the past decades, edge detection represented a huge challenge that even in present research days is difficult to overcome. This challenge starts at the point of defining edge. Such definitions have a variety number of answers. From the human perception it can be defined as something that separates different objects or parts of an image, that from the same point of view are identified by common sense. From Computer Vision's point of view it is generally defined as a strong contrast of intensity. In addition, it is also accepted that this intensity change has a direction on the image [6].

Many attempts have been made to create a good algorithm for edge detection. Among those attempts, some examples are the Sobel, Roberts-Cross and Canny.

Research done during the development of the Image Retrieval for E-Government Documents led to the selection of the Canny Edge Detector for edge detection because by using the same conditions on the referred algorithms, the Canny retrieves the best results according to general criteria for images types (e.g. man-made objects, landscapes, animals, etc.).

General implementations of this algorithm take a raw image in any colour space and the output is a grayscale image, where the lowest scale value (black colour) is not considered an edge. The values of the edges on the grayscale space depend on the variation of colour on the image. Considering this, the highest values are given when this variation is also high [6] [21].

This algorithm is applied to an image by the following steps:

- The first step is to apply to the raw image, a Gaussian filter in order to reduce the possible noise. The result is a blurred image without much significant noise;
- At this point, it is necessary to determine the direction of each edge. The directions' set should be limited to vertical, horizontal and diagonal, with the angles of 0, 45, 90 and 135 degrees. To get the directions for the edges, it is necessary to apply an existing edge detection operator that will give this information (e.g. Sobel Edge Detector).

$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

$$\mathbf{\Theta} = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

- Using the edge detector operator, there are two images given, Gx and Gy, that represent the result of convolution of the original image with the Gaussian filter on the horizontal (left to right) and vertical (top to bottom) directions, respectively. therefore, G represents the matrix of magnitude on the image and θ the angle function that determines the direction.
- The binary map from derivation example from the Sobel operator that shows the directions is shown in figure 11 B).
- At this stage of the algorithm, it is necessary to scan the result of deriving the image matrix with the edge detection operator and filter the pixels that are not on the ridge top. This is performed by comparing each point on the edge to the neighbors and the directions of the point, and those with the same direction and an intensity below the magnitude level are set to zero.
- As a final step, it is necessary to perform a threshold with hysteresis. This algorithm requires two threshold values, a low and a high. It considers that edges are continuous lines with high gradient values. The low threshold identifies the minimum acceptable value for the pixel to be considered a part of an edge. The second threshold helps to identify pixels that are surely a part of the edge. All pixels connected directly or indirectly to a pixel with value above the high threshold are considered edges if they are also above the low threshold.

Figure 11 shows a simple example of the Canny Edge Detection application.

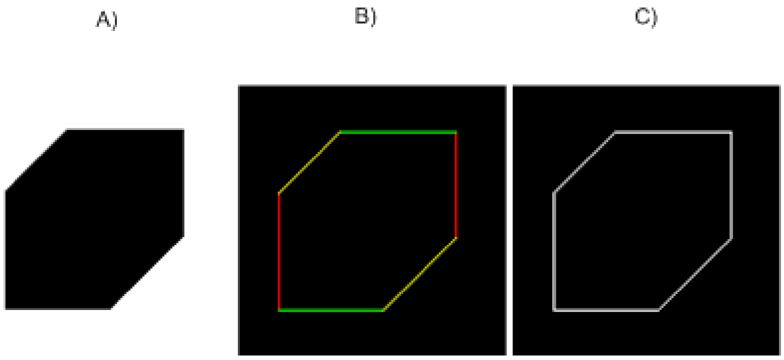


Figure 11: A) Original image B) edge operator direction mapping, resultant from the edge detection derivation C) final result with the default input values.

Unlike other pre-processing techniques presented in this report document, the Canny Edge Detector requires a set of parameters that are not easily manageable in order to get the desired results. Most common implementations of this edge detector method require as parameters, the definition of the Gaussian kernel size, the high and low threshold values.

For document retrieval it was found sufficient to fairly detect expected edges, the following parameter values:

- Gaussian kernel size: 5x5;
- High threshold value: 100
- Low threshold value: 20

Increasing the Gaussian kernel size, the image shows less susceptible to noise presence, but loses some detail at the same time, influencing the quality of edge detection and therefore, the final result.

Changing the low and high threshold values influences the edge pixels that are accepted as a final result. Setting both values to lower value, it makes that most edges are less likely to be included on the final set. On the other hand, setting higher values to this interval may include some undesirable pixels on the image.

3.9 Independent noise removal

On these next two sections independent noise removal will be discussed, and two solutions are proposed here in order to overcome this issue.

Independent noise (or random noise) is referred as noise that is generated by the input device that was used to make the digital image. It is mostly noticed on this system after the edge detection procedure, and it is considered that from the human perception as not being a part of the shapes that are wanted to be recognized by the system.

The method proposed on this section assumes that independent noise is a group of isolated pixels with lower density than the significant edges that the system must recognize on further steps.

From the assumption that this noise type appears in the image as separated sections of pixels, the algorithm to remove independent noise is based on scanning the image from left to right and top to bottom, and considering the pixel groups with the same size as the structuring element. In addition, to enhance this process, the image will only be scanned when necessary, meaning that the group of pixels will be scanned when at least one foreground pixel is found.

- For each foreground pixel found, its neighbourhood is analyzed as a group:
  - For each group of pixels it is considered a minimum number of pixels that an edge should have, in order to do not be considered as noise.

    If the number of pixels on that group is above the estimated value, the pixels are left as they were before;

    Otherwise, the pixels are set as background pixels.

There are two parameters to be considered here: the size of the group that will influence the number of pixels that are analyzed at a time, and the minimum number of pixels that a group needs to have in order to be accepted as not being noise.

Figure 12 shows an example of the independent noise removal algorithm usage:

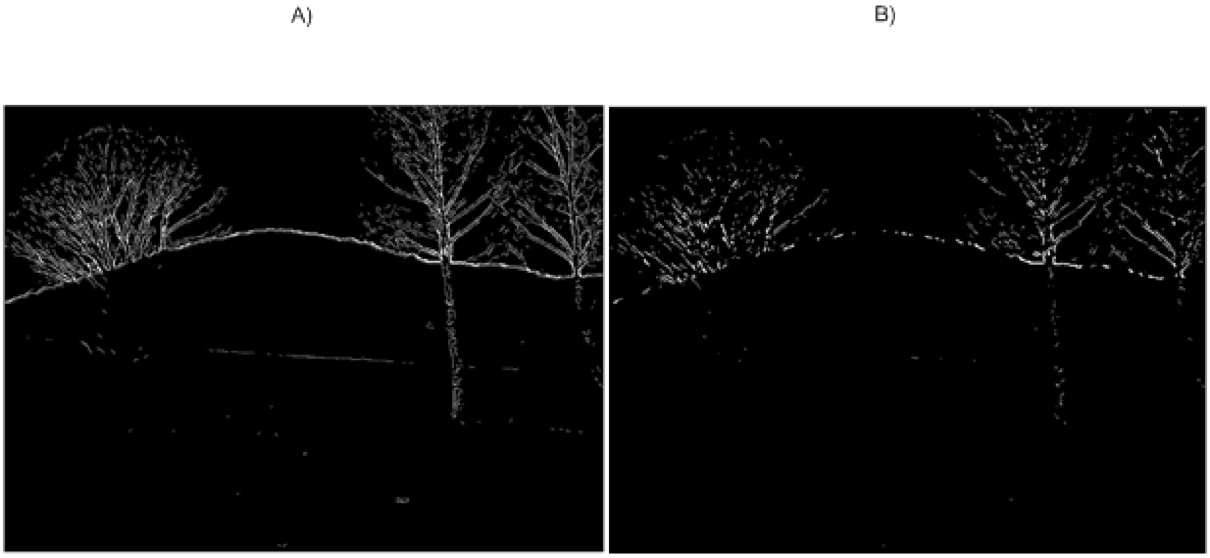


Figure 12: A) Image with the sampled edges B) The same image after an application of the noise reduction algorithm.

Because the application results depend much on the image size and detail of the retrieved edges, the expected quality may stay between insufficient and removal of significant parts of the image.

Changing the pixel group size and pixel number values may solve these problems but it will always end on human decisions to give the right direction to this process.

3.10 Noise removal by statistically defining noise

The previous procedure for independent noise removal has a clear disadvantage: that the definition of noise is an assumption from the user's input parameters into the system. On this section a more reliable algorithm for independent noise removal is proposed which idea is based on letting the system define noise by statistical measures.

Ideally, what seems to be noise on an image may be measured by the sections (on a binary image) that don't reassemble to nothing considered important on the retrieval context. From the Computer Vision perspective, describing noise may rely on mathematical procedures only. For this purpose, it becomes necessary that the system becomes able of defining noise by using an algorithm capable of judging noise by mathematically understanding the circumstances where the noise can be found.

When comparing to the independent noise removal algorithm, this differs on two concepts:

- As mentioned before, the noise is calculated instead of being assumed by the user, being no longer needed to specify parameters that would make the system define noise;
- On the other hand, it relies on the fact that the found noise is not erased, but labelled as being noise.

Labelling noise instead of removing it directly prevents the removal of wrong parts of an image assumed *a priori* that could eventually be a part of an edge. The noise is really removed later on shape detection by the following criteria:

- If labelled noise pixels are connected to a valid shape, then these pixels are considered a part of a shape and therefore, are not noise;
- If labelled pixels are isolated and no valid shape is associated to them, then these are considered as being independent noise and are discarded on that stage of processing.

This whole process starts by considering a binary image in its matrix representation which means that instead of working with a grayscale image, a two-dimensional matrix is used . It is assumed at this point that this binary image the edges are sampled already, where:

- 0's are background;
- 1's are edges and noise.

At this stage, it is necessary to measure the amount of possible noise by scanning the matrix in four directions (from left to right, top to bottom, right to left and bottom to top), and calculate how many edge pixels (positions with value 1 on the matrix), are sequential for each direction.

The results will be placed in four matrices, one for the respective direction. For example, considering the direction from left to right, there is the following sequence: {0,0,0,1,0,1,1,1,0}, the resultant matrix will then be {0,0,0,1,0,1,2,3,0}. This maintains an exact location where such sequences were found, as well as the number of sequential edge pixels that was found on the exact locations. On the example above, there is are two sequences of one pixel and three, these sequences are shown on the resultant matrix on the exact location where they were found.

With the current information, it is possible to know the density for the edges on each direction, now to label the noise pixels, it is necessary to scan the initial matrix once more, and for each position with value 1, the four matrices are combined in the same position, and if the four matrices' sequence number in that point is equal or less than the maximum sequence number of each, divided by 5, then the pixel is labeled as being noise.

The final result will be the changed image matrix, with 0's on black pixels, 1's on white pixels (that are definitely not noise), and 2's on pixels that are suspected to be noise.

Considering another example, the matrix from the direction left to right: {1,0,1,2,3,4,5,0}, retrieved from the image {1,0,1,1,1,1,1,0}, the resultant matrix would be {2,0,2,1,1,1,1,0} (considering only that the matrix is scanned in one direction).

Figure 13 shows an example of the application of this algorithm. The results can be compared with figure 12, as the sample image was given in the same conditions for both.

A) B)

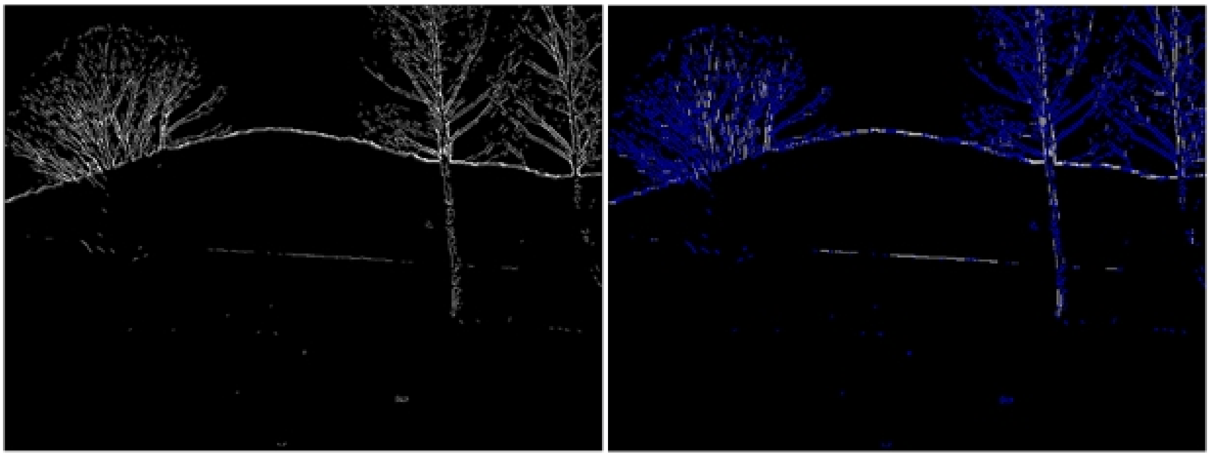


Figure 13: A) With the calculated edge detection B) Image with the pixels labeled as being noise at blue colour

The samples that result from this process only include labels where sets of pixels are supposed to be noise. With the pixels labelled, the noise can be identified and removed during the shape recognition process by isolating labelled pixels that are

3.11 Conclusions

In this chapter, many pre-processing techniques were proposed, and it is clear that one simple application of one single method will not suffice this pre-processing stage Image Retrieval. In fact, it is necessary to use a combination of these techniques. However, many questions: arise: when a pre-processing technique should be applied? Which circumstances are favourable for each method? An activity diagram is shown on Figure 14, which shows all possible combinations of the presented methods.

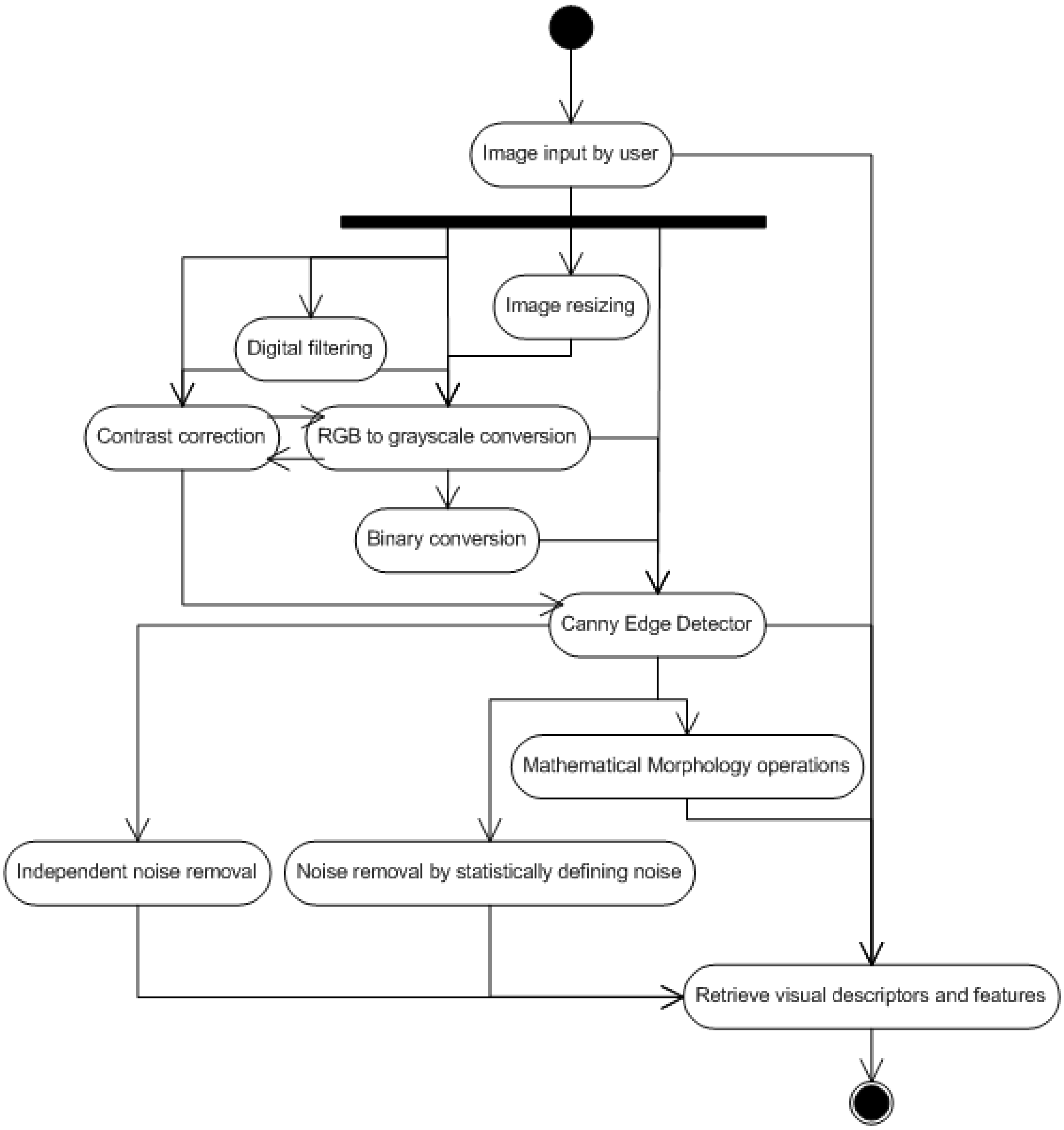


Figure 14: Activity diagram for the possible pre-processing sequence on a raw image.

It is not implied that the pre-processing have that order in general situations, but it is important to follow the sequence illustrated in figure 14, in order to create conditions for the next stages of image feature extraction that the implemented system requires.

Some pre-processing techniques have their influence on effectiveness of resource and time consumption management or quality of image so that a good visual content and feature extraction can be performed.

For example, image resize influences the speed of image processing because it was mostly used to get the given image in smaller dimensions, at a cost of image quality. All other pre-processing methods that are executed before edge detection have the common goal to enhance the quality of the edge detection itself. Noise reduction and Mathematical Morphology operations are used to increase the quality of shape detection.

It is still difficult to determine which subsequence is the best for all cases: some document templates require all; some require no pre-processing at all. In addition, the type of feature extraction influences this selection.

The pre-processing step is still a part of the user perception because it doesn't concern the final image to be used for feature extraction, but what is necessary to be extracted.

The next two chapters present two different approaches for feature extraction and similarity comparison, and for each one of them, different combinations of pre-processing are required as well.

4 Colour feature extraction and similarity evaluation

4.1 Introduction

This chapter presents the implemented method for extracting colour features by analysing image content based on the RGB colour space.

Representation of image features on colour domain is a subject with most use among image retrieval method types. The main reasons point to its simplicity of representation and being less involving than other feature types. In addition, it is a type of representation usually less susceptible to noise presence and invariant to image orientation.

Like most of the implemented systems, this method uses histogram-based retrievals that allow construction of feature vectors by using statistical information that can be retrieved on images.

Furthermore, this method is combined with an image segmentation method in order to overcome some problem of high costs of processing that image retrieval based on histograms cause.

In addition, human perception only considers relevant representation of an image, where the finest details are not involved. Therefore, this method will simplify the feature extraction by considering dominant colours from a small set instead of the complete RGB colour space [5].

As mentioned before, colour feature extraction has the advantage of retrieving information from visual content and being invariant to some challenging problems like image orientation and noise. This method was developed in order to enhance this invariance and also to input invariance to scale differences. This achievement is done along with the segmentation, where the representation of features is done by considering the histogram colours in percentage instead of maintaining the full information of a pixel.

The following sections of this chapter describe the proposed method in more detail.

4.2 Colour descriptors and colour features

The stage of visual content extraction is sequent to the pre-processing stage, although, this method doesn't require any pre-processing at all. In fact, in order to obtain the best results it is required that the input image has the best colour representation as possible, and most of the techniques for pre-processing that are used would modify the colour structure of images.

The method here described allows colour features to be extracted from an image by averaging its colour space into a small and predefined domain set of colours. The idea is to create segments on the image that correspond to the same colour average present on that set, eventually removing noise and fine details that histogram-based algorithms are affected by [5].

The algorithm works around these definitions:

- Colour set;
- Local colour descriptor;
- Global colour descriptor.

Colour sets are user predefined, and should contain well defined and distinct colours only. Each element on the set is composed by the colour name and the respective RGB values for each. For example, in this project, the colour set was composed by black, white, red, green, blue, yellow, magenta and cyan.

To each segment on the image it would be corresponded one colour of the set that is stored on the respective colour descriptor. As mentioned before, this value is defined by retrieving the average colour on the segment. The colour descriptor for each segment is then, a local feature because it represents a description of a section of an image. The segmentation is made by dividing the image in clusters and the dimension for each must be predefined by the user.

Global colour descriptor is defined as being a representation of global colour features on images. Therefore, this colour descriptor type contains information about the colour in the set, which is calculated by the amount of the same local colour descriptors in the image, in percentage.

The whole process can be summarized as follows:

- An image is received from the pre-processing stage (pre-processed or not);
- The image is then scanned from left to right, from top to bottom by analysing the pixel in clusters. For each cluster:
    - The local colour descriptor is retrieved. This is done by analysing each pixel on the cluster and calculating the average colour on the set;
    - When all pixels are averaged to a value, the local colour descriptor for the current cluster is the same as its dominant colour (which the colour on the set that has more presence on the cluster);

    Figure 15 illustrates this process for one cluster.

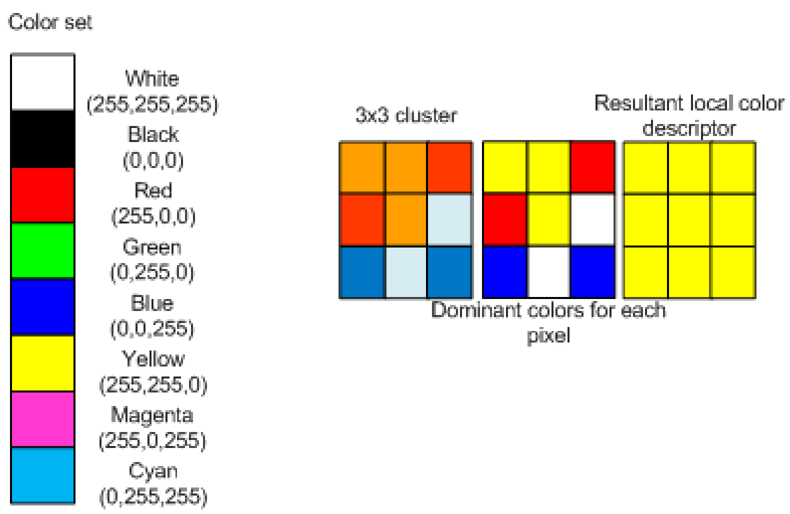


Figure 15: Process for selecting a local colour descriptor from a cluster, considering the given set on the left.

- When all clusters' dominant colours are retrieved, the system would have to calculate the global colour descriptors. The process is done similarly to the calculation of local descriptors on a single cluster: the selection of the global dominant colours is made from the amount of presence of the local descriptors.
- The final feature vector is then retrieved as a list of dominant colours and their respective percentage sorted from the major presence to the minor. For instance, {(white, 60), (black, 15), (yellow, 10), (blue, 10), (green, 5), (red, 0), (magenta, 0), (cyan, 0)}.

Any feature vector retrieved using this method has at least one element non-zero value and at most ten elements, making it easy to manage and understand.

Figure 16 shows an output example of the application of this method.

Figure 16: A) A raw image (without any pre-processing applied) B) The colour descriptor's generated map for this image with clusters with 2x2 of size C) The output log with the colour features retrieved.

Unfortunately, the results that can be seen when applying this method do not suffice the identification of documents, because the general set of templates has elements with black or white as dominant colours, most of the other colours set are inexistent on this image type. Although this method can be an useful approach when considering single elements of a document (e.g. logos and images). Furthermore, another possible application is using the resultant feature vectors to index the images on a database by considering the colour with the most presence, then the second dominant colour, etc.

4.3 Similarity evaluation using colour features

As mentioned before, a vector of features is, in general cases, a brief representation of the details on the visual content domains (colour, texture, shape and spatial relationship), and comparison between feature vectors is made is accordance with the metric used.

The method for colour retrieval on documents presented on the previous section uses a very specific and uncommon metric for its description, since in general implementations of feature extraction there aren't theoretically limits for feature values. On this particular case, the description is related to the colour presence on an image, and therefore, it resides in a range between values 0% and 100%, making the possible distance of similarity also limited.

It also differs from other common methods by the fact that the user is responsible for the generic colour set input. The quality of this set selection will have its influence on the final result. Generally, the output should be based on visual contents only.

Considering these characteristics about the feature vectors, the comparison method is also adapted to the given circumstances. A comparison of feature vectors is generally done by calculating the distance value between the vectors' values directly, but because these values are within a range, and furthermore, are associated to a certain colour names, the comparison cannot be done so directly (e.g. the amount of blue presence cannot be compared by the amount of presence of white colour).

The distance is then calculated by the given steps:

- Two colour feature vectors are supposedly given as an input, the first is retrieved from the query image while the second is already stored at the database;
- Considering each element on the first vector by the colour name, it is necessary to find the respective colour on the second vector in order to make the comparison of presence on each images;
- For each colour, a value of difference is stored. Let feature *f1* representing white colour on an image have 25% of presence on the query image, and feature *f2* that relates to the same colour on an image on the database with 24% of presence. In this case, the distance is 1% because that is the difference of presence that lacks in one of the images.
- Once all values are computed for all colours, the total similarity is given by *100%-total difference*. The total difference is a sum of all differences computed for each colour.

It is to be noticed that in some cases, the total difference value would eventually be higher than the maximum possible value of 100%, which will lead to negative values of similarity. For this case, the similarity value is set to its minimum of 0%.

Some application results of similarity retrieval can be seen on figure 17.

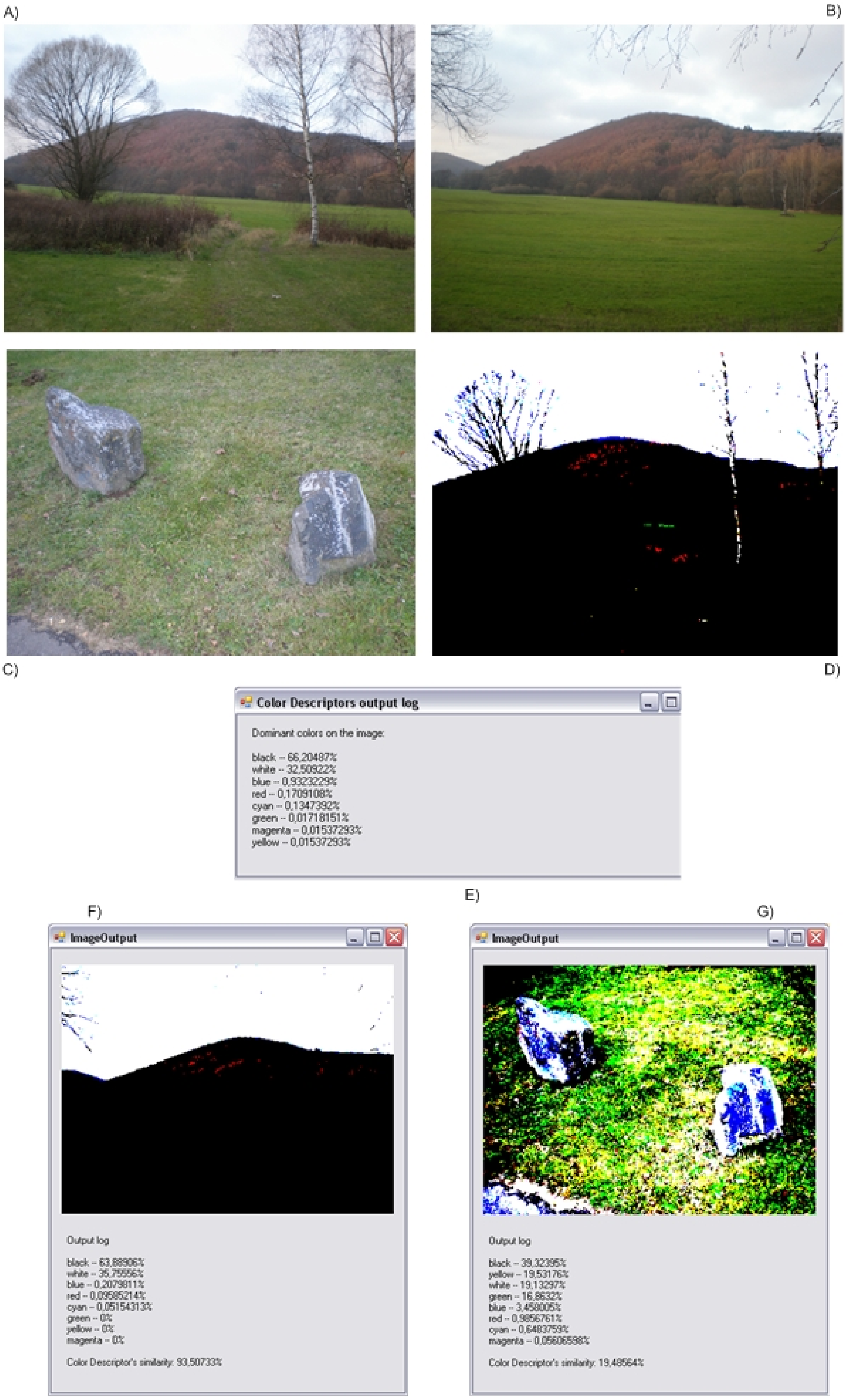


Figure 17: A) Query image B) Image from database C) Another image from database D) Colour descriptors' map from query image E) Colour features vector output from query image F) Similarity comparison results between query and the image on B) G) Similarity comparison results between query and the image on C)

4.4 Conclusions

The described method is a simple way of retrieving colour feature vectors that are easily understandable. However, when analysing some results, it is clear that its usage needs some attention due to some issues.

First, this method is more dependent on luminance variations than in colour, this effect can be seen on the retrieval example shown on figure 17, where both samples on the database have similar colours, but its brightness difference has heavy influence on the colour features computing.

In addition, because this method is totally independent from any semantic or structural information, it may lead to false assumptions of similarity.

Besides the problems encountered, this method may provide good support of Image Retrieval procedures but it is required to be combined with other procedures of image recognition.

It was also possible to verify the invariance to rotation, scaling as well. Furthermore, the presence of noise has little significance on the results. It is also invariant to translation operation if independent objects are considered instead of full image content.

The next chapter describes a more complex method for feature extraction based on structure content on an image.

5 Structure feature extraction and similarity evaluation

5.1 Introduction

In this chapter it will be introduced the method for structure feature extraction based on shape recognition techniques, used on this project.

Shape is one of the most important and challenging features on Image Retrieval field of study. During the last years, research attempts on this area have found most of the issues on structure feature extraction focused on shape representation, similarity measure and shape indexing.

Shape representation is generally considered the most important issue on structure feature extraction. It is described differently between human and Computer Vision, however the approach to its description is done similarly. Human perception tends to automatically divide a scene on an image into objects. Such objects are represented by edges and shapes, in a low level analysis. In Computer Vision systems, the process to identify low level features from high level inputs is done with the same approach, but to retrieve such information different methods are necessary because the understanding of these low level components is also different.

The step to process an image for its edge representation is done when applying pre-processing techniques for that effect. On a specific use case, the quality of this stage is therefore dependent on the pre-processing methods that were previously applied to the image. Thus, it becomes complex and involving the detection of the same objects that human perception would allow to detect.

Structure features are then extracted through the following steps:

- Object and shape recognition: It identifies and labels the objects that are present on a given image by identifying significant pixel groups that are connected edges. From each object, the shape's basic information is extracted;
- Shape description and feature extraction: from the available shapes, it is necessary to represent the shape using a reliable describing method. From this result the feature vector extraction is possible.

Like in colour descriptors, similarity retrieval is performed depending on the metric used and the form which the feature vectors are presented in this process.

The next section on this chapter introduces the definition of objects on an image and the methods used to describe the information about a shape.

5.2 Object and shape recognition procedures

As mentioned before, there is a difference between perception of objects and shapes in human and Computer Vision. Computer Vision retrieves an object based on the edge representation instead of the human sense. This section introduces a proposal of a method for object and shape perception and labelling based on the edge detection technique described on chapter 3.

To understand this structure analysis method, the definitions below are required to be introduced:

- Image object: It is what in human perception is considered as being an "object" in Computer Vision point of view. It is composed by a set of edge pixels linked together.
- Shape: It is defined as being a group of sequential pixels that form the outer boundary of an image object. A shape is then, a specific property of an image object, as it can be seen on figure 18.

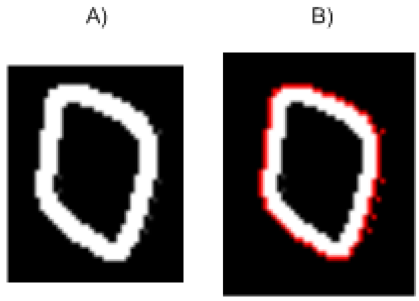


Figure 18: A) Image object without a shape defined B) Image object with shape defined by red pixels

Unlike in the colour features retrieval method, the quality of image retrieval isn't regardless to the pre-processing applied. In fact, a good pre-processing results leads to good quality on edge detection, and therefore, a good quality on object detection. For example, on the example illustrated on figure 18, a wrong choice of pre-processing methods could make the object completely different.

The introduced concepts are two methods for pixel labelling that actually indicates what could be an object from the human perception. Pixel labelling methods result on converting the visual content into basic information about the image objects (e.g. its position on the image, the position and value of the pixels that belong to the object or its region of interest).

Reasons that lead to a choice of pixel labelling methods regard the feature extraction method. On this project it was used the Fourier Descriptor method that uses a shape (as it is shown on figure 18). On some other feature extraction methods, it would suffice just to retrieve information about the image objects.

To describe a shape, it is then necessary to detect the corresponding image object first. The implemented process that finds all image objects on an image is as follows:

- The image is scanned from left to right and from top to bottom until an edge pixel is found;
- Starting from the pixel found, the connected pixels are searched in all directions. For each connected pixel found, the same procedure is repeated until no more pixels are directly or indirectly connected to the first pixel, leaving as result the information of the whole group of pixels;
- One thing to be noticed while the image object is retrieved is that the edges may have voted pixels as being noise from the previous pre-processing techniques. On this event, the voted pixels are accepted as objects if they are directly or indirectly connected to a non-noise pixel.

- With the information just retrieved, the considered pixels are then marked as "visited" and the procedure starts all over again. Marking pixels in such way will avoid that the same image object is detected a second time.

The next steps from this point on will not use the visual content on the image, as all needed information about image objects will suffice the requirements.

Using the previously described process may eventually detect noise as an image object, making this procedure very dependent on a fair choice of noise reduction. On such cases, it is impossible to have a good base of judgement of which image objects were noise. On the specific case of document processing, it can be judged by basic characteristics such as the small dimensions that noise eventually have (unless it is dependent noise).

Using the information from an image object, a shape is retrieved using a more complex procedure described as follows:

- For each image object:
  - Considering the region of interest of the image object, take the first pixel found while scanning from left to right and top to bottom. The first pixel belonging to the image object is more likely to be a pixel on the boundary.
  - With the first pixel, it is necessary to form the shape. It is a complex procedure that requires a more specific search. The main difference between retrieving a shape and an image object is that a shape doesn't accept all pixels in the image object region. For this purpose the image object is scanned from the first pixel found as a moving cursor around the shape.
  - Each pixel in the boundary is visited by the moving cursor that performs its trajectory considering the counter-clockwise direction with a view range of 180º and giving priority to the pixels on the further right on the view. The process is finished when the first pixel is found again so that the result is a closed curve.
  - Such process can be compared to a situation of a car cornering a lake, where the car is the moving cursor on the image and the lake is the image object. Figure 19 illustrates this process.

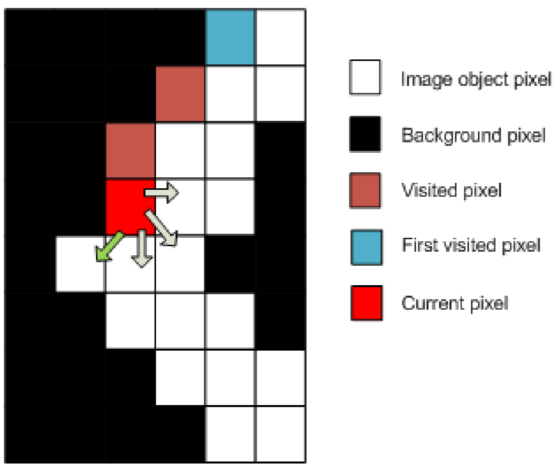


Figure 19: A section of an image object and the process of shape retrieval with a moving cursor

- The cursor is implemented in a way that it can adapt itself to autonomously change the direction to corner the boundary. This is done by checking the new direction possible each time a new pixel is visited.

A perfect shape is, as mentioned before, a sequence of pixels from the outer boundary of an image object. Finding this solution is complicated due to problems such as existent disconnectivity on image objects' boundaries or presence of dependent noise that may result on discontinuous sections of the shape. Solving such problems is complicated and generally takes much more processing time. The problems with this nature can be corrected by first identify the discontinuous pixel sequence that originated the issue. The sequence is then corrected by rolling back either to a few visited pixels back or to the very first pixel, and once the discontinuous branch is found a second time, the cursor moves to the next acceptable pixel that is the closest to the boundary as possible. This process is similar to a search in a tree structure.

At this stage in the process, the image object is no longer needed and the features can be extracted using the shape information only. The following section describes the methods to extract features from shape structures.

5.3 Structure feature extraction using Fourier Descriptor method

With the current shape representation achieved, there are some well known techniques that allow the description of this type of feature. Some of these techniques involve different type of information that can be derived from image object or shape representation such as image region representation which is technically used as a gray level measure on specific regions (e.g. the definition of image object is one type of region representation); or contour representation for the outer borders of edges (e.g. the defined shape representation is a contour representation).

Region representation description methods tend to use global information of an image object. As consequence, the information on the boundary is most likely to lose relevance on the similarity comparison stage. This representation type also works with image structure properties that are representation of the probability function along the image object. These properties are often named as Image Moments. Derived from this definition, there are many techniques based on this image moments, the most used among them is the Hu's Set of Invariant Moments [8].

Contour based approaches require more complex solutions because this representation type uses more specific information on an image object. But despite this fact there are some well known and robust techniques based on structure descriptors such as the Wavelet Descriptors and the Fourier Descriptors that are derived from the spectral transforms, Wavelet Transform and Fourier Transform, respectively [12].

Wavelet Descriptor based methods provide better results because it achieves finer details that usually Fourier Descriptor based methods don't consider on either spatial or spectral domains. On the other hand, Fourier Descriptor methods provide better access to basic information with much less processing effort. In addition, because these methods don't consider the finest details on the shape, dependent noise is often easy to overcome. Considering these features, a method based on Fourier Descriptors was chosen for this project.

Several methods have been proposed on Fourier Descriptors that focus on different representation approach such as the pixel coordinates, the angular function or the curvature representations. Every method has also different results because they have different shape representations as well. [7] [9] [10] [11] [12]

Generally speaking, the Fourier Descriptors are retrieved through a system that receives a given shape representation and the descriptors are retrieved through Fourier Transform.

The implemented method proposes a Fourier Descriptor based method that uses pixel distance to the shape's centroid in coordinate representation from the shape representation type previously mentioned. This method also provides a shape description that is invariant to translation, rotation and scaling of image objects' shape. In another words, the final feature vector resultant from this process must be comparable independently of the circumstances which the image can be found.

Such process was then adapted due to the specific subject of document retrieval. It is an extensive procedure that requires other methods in order to achieve invariance to the explained conditions. It is described as follows:

- At the first stage of this process, the image should have its pre-processing done and containing the set of shape represented by a list of sequential pixels that form a boundary for each image object.
- Unlike colour features recognition, each image object will have its own vector of features. Thus there is no global features' vector for the image. Therefore, all features are local features and are independently used from the image's circumstances.
- For each shape on the image:
  - Because objects will eventually be in different sizes, the shape must be normalized. This means that only a set of pixels will be strategically selected. From the principle that candidates should be equally spaced, the pixels selected have the same arc length between neighbours. Let *P* be the perimeter of the shape and *K* the number of candidates, the ideal arc length is then given by *P/K*.
  - A normalized shape illustration can be seen on figure 20.

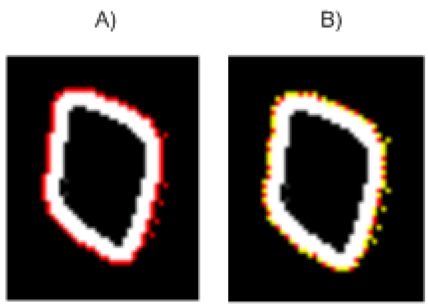


Figure 20: A) Shape represented by red colour on an image object B) Normalized shape to 64 pixels marked at yellow.

  - From the normalized shape. It is now necessary to perform a Forward Fourier Transform that will result in a vector of Fourier Descriptors, so it is also needed to convert the given coordinates in complex numbers in order to apply the Fourier Transform definition..

- To achieve translation invariance, the given input should be the distance from each pixel to the shape's centroid. The centroid *(Xc, Yc)* is given by:

$$x_c = \frac{1}{L}\sum_{t=0}^{L-1} x(t),\ y_c = \frac{1}{L}\sum_{t=0}^{L-1} y(t)$$

- Where *L* is the total number of pixels, *x(t)* and *y(t)* are the functions for the coordinates of those pixels.
- The distance to the centroid for the normalized set of pixels is then given by the following formula:

$$r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{1/2}$$

- Where *r(t)* is the result set of distances to the shape's centroid
- Because *r(t)* is given in numbers from the real set, the resultant coordinate set *s(t)* is given by *s(t) = r(t) +0i*, making the imaginary part of each number excluded from the translation invariance calculation.
- The discrete Fourier transform goes as follows:

$$u_n = \frac{1}{N}\sum_{t=0}^{N-1} s(t)\exp\left(\frac{-j2\pi nt}{N}\right),\ n = 0, 1, \ldots, N\text{-}1$$

- Invariance to rotation can now be analyzed. Considering the phase and magnitude that can be calculated from the transformed set, rotation invariance can be achieved by ignoring the phase component. This means that the final set of Fourier Descriptors will be given by the magnitudes of the transformed set.
- Invariance to scale is granted when the feature vector is calculated from the fourier Descriptors vector. It is done by dividing each Fourier Descriptor, from position 1 to *N*, by the value on the position 0.

$$\mathbf{f} = \frac{|FD_1|}{|FD_0|}, \frac{|FD_2|}{|FD_0|}, \ldots, \frac{|FD_{N/2}|}{|FD_0|}$$

- Where each *FD* is a Fourier Descriptor and *f* is the final features set that will be used to similarity retrieval. Because there are only N/2 values of different frequencies, there is only needed to be calculated half of the FD vector.

Procedures for Fourier Descriptors retrieval usually involve many more functions that are variations besides the basic transformation step as shown in the previously procedure steps.

The interpretation of the values that result as being the elements of the feature vector relate to the level of detail that the object is described. The lower frequencies of the Fourier Transform describe the general parts of a shape while the higher values describe the finer details. This effect can be verified to the resultant vector, a reverse Fourier Transform is performed. The shape that is then supposedly recovered will look less detailed. This also serves as an explanation about why the first index of the descriptor list isn't considered to describe the shape. It is because the first value relates to the position of the shape only.

The next section describes how the similarity is retrieved from a comparison between two vectors of this type.

5.4 Similarity retrieval using structure features

Unlike the method used for colour retrieval, the features described on the last section are more generic feature representation, which means that they also use a more generic metric and similarity retrieval method.

From the principle that the feature values have no maximum value, then the vector distances will not have a range set of values as well. The base of judgment of which shapes can be considered similar among themselves is then different, and usually only after some experiments with the metrics it is when the similarity can be seen.

The most widely used methods for feature vector comparison are based on regular vector distance based formulas such as the Minkowski Form Distance, Quadratic Form Distance, Makalanobis Distance and Kullback-Leibler Divergence and Jeffrey-Divergence. These mentioned methods work on distance metrics based on the Euclidean Distance, which is the most common form of similarity retrieval from this vector type [7] [11].

This method consists on simply calculating the distance between two feature vectors using the Euclidean Distance's 2nd norm.

For example, let the following vectors be two vectors of features retrieved from two different images. So the distance between these two vectors is given by:

$$\vec{x}_1 = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \end{bmatrix}^T \quad \text{i} \quad \vec{x}_2 = \begin{bmatrix} x_{21} & x_{22} & \dots & x_{2n} \end{bmatrix}^T.$$

$$||\vec{x}_1 - \vec{x}_2|| = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

Expectations for the results depend on the content of each vector. The closer it is to 0, the more similar these two vectors are.

Figure 21 illustrates two examples with objects given in different conditions

A) B)

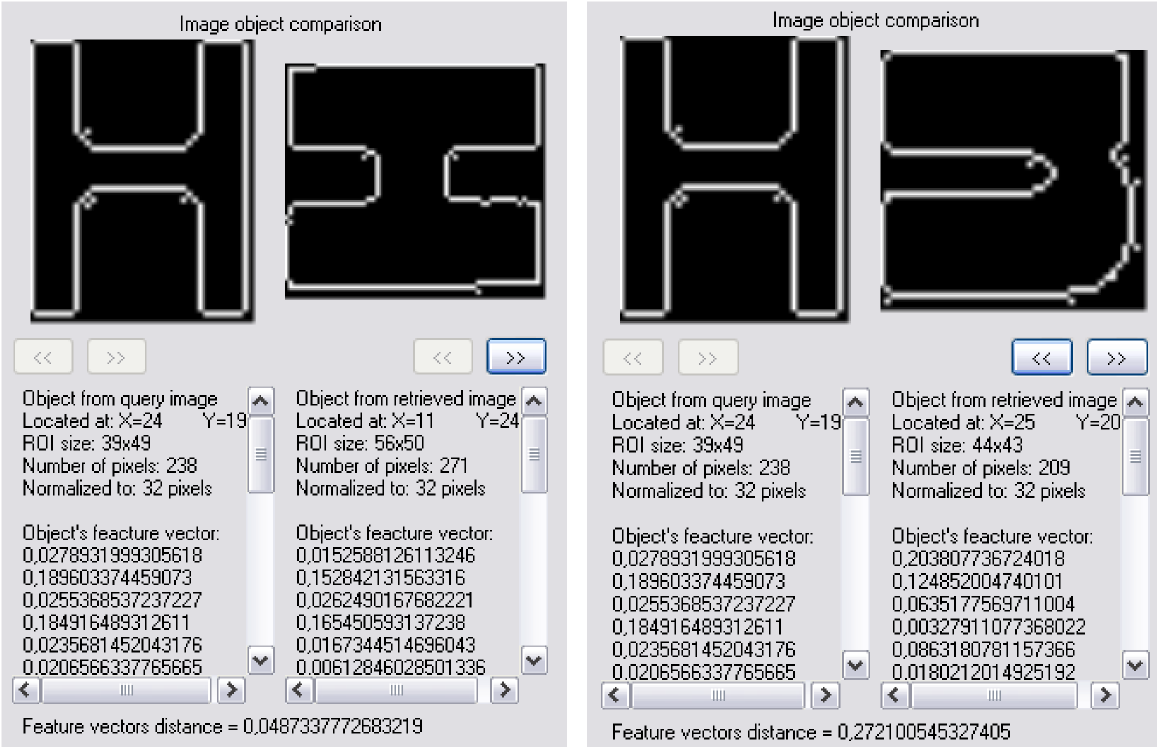


Figure 21: A) Two similar objects B) Two different objects.

Due to the normalization process, this method becomes robust when dealing with dependent noise because in most of the times it is not taken into account to the Fourier Descriptor process. It is also possible to see which distance values can be considered a safe statement of similarity even without a maximum value.

5.5 Conclusions

The methodologies presented on this chapter showed one of the many possible ways to retrieve structure features. The specific solution was implemented regarding the problem of single object detection among images on databases. Most of this project's efforts were made related to structure feature retrieval because it is more relevant on document template retrieval than recognition made by colour.

Most of the efforts on the implementation of this solution were to enhance the ability to overcome the many challenges that still remain on the shape representation process. Such problems usually start as a consequence of the pre-processing stage that is working on most of the methods regardless to automated decisions and sensitivity to image characteristics such as colour and luminance variance.

Furthermore, the shape recognition is made regardless to any orientation or existing information, thus it is difficult to control presence of undesired shape sections that are generally known as dependent noise. The reason for that is the difficulty in Computer Vision to create such sensitivity on Content-based Image Retrieval applications that allow machines to have critical sense.

Although, adding the normalization method for shape representation was a robust way of ignoring the noise. But even a normalized shape may become an issue if a wrong number of candidate pixels are chosen. Increasing this value will make more Fourier Descriptors would be included and consequently, the finer details of the object could be included, which would eventually increase the amount of error on the shape representation. On the other hand, decreasing this value would hide undesirable details but if this value is to low when compared to the amount of pixels on the shape, it would certainly create a situation where shape descriptions are ambiguous to this type of similarity retrieval.

Most of the hard work on this procedure from the user's point of view is then finding a solution that the shapes can be described as best as possible. Figure 22 illustrates a case scenario applied to the main problem where the shapes are retrieved without noise or other signal problems.

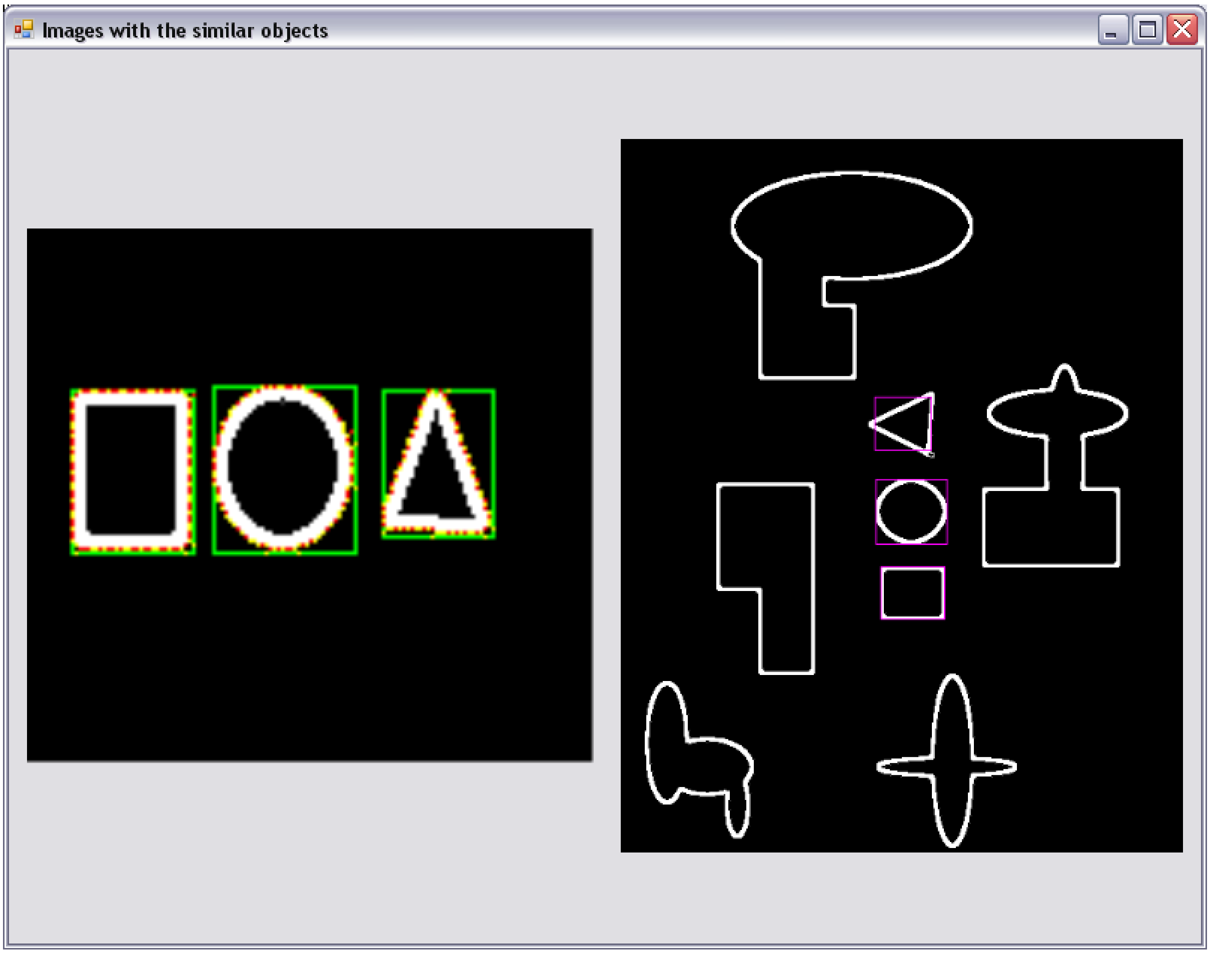


Figure 22: Visual output of similarity retrieval where the set present in the left image is found on the right image on the database among other image objects.

However even with the shapes well represented there may be problems due to ambiguous information. It is a result of this retrieval being based on the shape information only, and sometimes different information may have similar shape representation. Such situation is illustrated on figure 23,

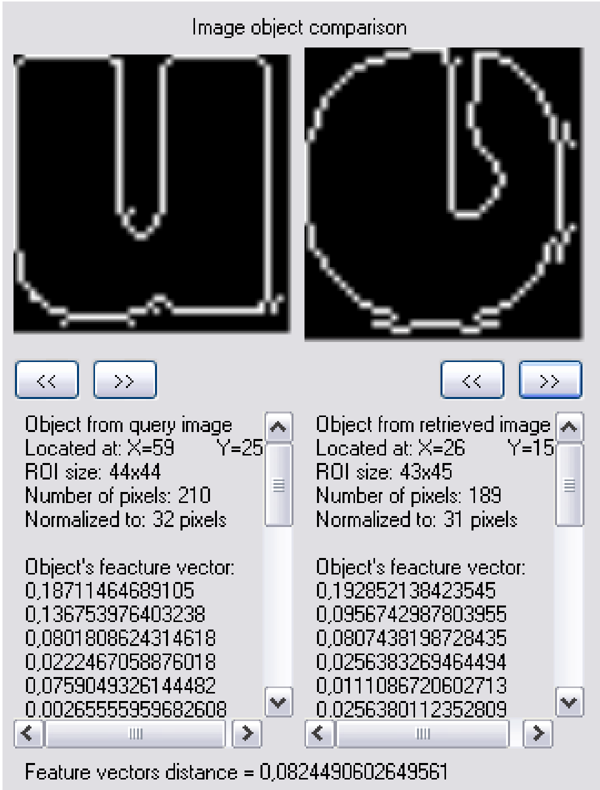


Figure 23: An example of a wrong judgement of similarity

The example above shows that because of the shapes are similar, it concludes by using the Fourier Descriptor retrieval method that both images are similar. However from the human perception, it is known that it is not true because the object can actually be recognized as being letters of the alphabet.

With this case it can be verified that even despite of being a robust solution is not reliable just by itself. However such case could be fairly avoided if it was combined with a method that approaches another feature such as colour, texture or spatial relationship.

When considering both objects on figure 23 on true colours, from the conclusions of the colour descriptors, the dominant colour retrieval could be a consistent resolution because the feature vectors there would be different. However it is also true that structure feature extraction methods were created to retrieve information from an image regardless to colour variance.

Although the most reliable combination could be the combination of this method with a semantic-based method for Image Retrieval. It could not work for any arbitrary image content, but documents are mostly composed by characters that are well known by systems such as the OCR (Optical Character Recognition) based systems. As for document images such as logos or diagrams the problem is still complex and challenging to be used along with digital image processing, however, most methods including the presented procedure on this chapter are reliable enough with sketched images interpretation due to the absence of problems caused by digital input such as noise.

6 Conclusion

6.1 Implementation results and achievement analysis

In this report document it was presented a set of implemented methodologies related to Image Retrieval on the context of Image Retrieval for E-Government Documents. With the efforts on study and development of this project, not only was possible to evaluate the state of art of this field of study, but it also allowed to verify on which conditions the proposed objective can be achieved.

In fact, it was possible to verify that despite of the many solutions available, there are many more obstacles that still remain not avoidable on this and many implemented systems. However, the wide range of usability in Computer Vision makes it worth the efforts. In addition, because developments in this area are generally for specific purposes, some systems can be successfully implemented due to the target area having restrictions on its requirements, meaning that less effort in research is required.

As mentioned before, the system implemented during this project was also for a specific purpose: to work with German government document templates based on images' visual content and regardless to other external information such as metadata or manual user inputs. More specifically, the system would be able to recognize specific objects on a scene given by a query image input. Like in the specific-based purpose systems mentioned before, this system have also limitation requirements that makes it less sensitive to problems regarding image recognition. Limitations focus on specific query construction, pre-processing techniques applied and how the feature extraction is done.

When analysing through the stages of Image Retrieval from this system, it is clear that decisions that were made initially, as in on the first stages of image query specification and preprocessing, have much more impact on the result than in later stages. In fact, successful retrieval is more likely to occur if the query is visually well constructed. For example, drawn images provide better results than photos because they are not affected by digital signal problems such as noise. In addition, queries that are provided by digital cameras are affected by several other factors like colour or luminance. Digital images are also dependent on the environment where they are created.

Image processing techniques were applied to solve such problems. For this system, user made images does not require any pre-processing at all because the conditions can be managed by the user. On the other hand, image processing was developed under a great effort to enhance digital image recognition quality. It is true that the further the issues are carried on, unsolved through the recognition process, the harder it is to solve them on the final stages, and because it is not always possible, the pre-processing structure must be robust enough to control such problems. On this system, colour based problems are more easily handled than structure based because shape recognition is still a hardcore problem.

Feature extraction process is the least manageable of all on this system because only two approaches exist. And because documents don't have well defined colours and are textureless, the structure extraction method based on Fourier Descriptors has more success despite of the problems on shape representation.

Nevertheless the implemented system is capable of recognizing well represented objects on a scene and its methods are invariant to rotation, translation and scaling, which allow most of the circumstances found on document recognition to be identified. However, as explained on this chapter, the query input is restricted to certain quality parameters that provide the best results as possible.

6.2 Further work

Despite of the complex subjects related to the work done, the developed project represents a small part of the FABEGG system. The results described and illustrated along this document were an outcome of an application that was developed independently from the existing system, thus it uses none of the existing methodologies and restrictions.

Therefore it is necessary as a first step, to integrate the implemented solution in the FABEGG system, eventually adapting developed application to the existing Image Retrieval system's workflow. The integration would allow that the methodologies that were developed could be a part of the existing human-computer interaction enhancement and eventually using the existing methodologies on Image Retrieval for the FABEGG system.

Finally, the result system requires a dedicated database system at the government authorities. Because the FABEGG system works as a real time client/server application set, it is necessary that the same time consumption at the client side does not happen at the database server. It is required that operations such as pre-processing, colour descriptors' retrieval or shape retrieval are previously done and stored at the same database.

References

[1] KUHN, Norbert; RICHTER, Stefan; NAUMANN Stefan – Improving accessibility to business processes for disabled people by document tagging, Institute for Software Systems in Business, Environment and Administration, University of Applied Sciences Trier, Location Birkenfeld, 2007

[2] IQBAL, Quasim; AGGARWAL, J. K. - Feature Integration, Multi-Image Queries and Relevance Feedback in Image Retrieval, [Online document], Computer and Vision Research Center Department of Electrical and Computer Engineering, The University of Texas, Austin, 2003. Available at http://amazon.ece.utexas.edu/~qasim/papers/iqbal_visual03_HDR.pdf

[3] KHOTANZAD, Alireza; HERNANDEZ, Orlando – Color Image Retrieval Using Multispectral Random Field Texture Models, [Online document], Department of Electrical Engineering, Southern Methodist University, Dallas, 2008. Available at http://spib.rice.edu/DSP2000/submission/DSP/papers/paper076/paper076.pdf

[4] VAN DE WEIJER, Joost – Color Features and Local Structure in Images, [Online document], University of Amesterdan, 2004. Available at http://lear.inrialpes.fr/people/vandeweijer/papers/thesis.pdf

[5] VALOVA, Irena; RACHEV, Boris – Retrieval by Color Features in Image Databases, [Online document], University of Rousse, Department of Computer Systems and Technologies and Technical University of Varna, Department of Computer Systems and Technologies, 2008. Available at http://www.sztaki.hu/conferences/ADBIS/17-Valova.pdf

[6] BRUNNER, Gerd – Structure Features for Content-Based Image Retrieval and Classification Problems, [Online document], Albert-Ludwigs-Universit¨at Freiburg im Breisgau, 2006. Available at http://deposit.d-nb.de/cgi-bin/dokserv?idn=983387052&dok_var=d1&dok_ext=pdf&filename=983387052.pdf

[7] LONG, Fuhui; HONGIJANG, Zhang; FENG, David – Fundamentals of Content-Based Image Retrieval, [Online document], pp 1-26, Microsoft Research Asia, 2008. Available at http://research.microsoft.com/asia/dload_files/group/mcomputing/2003P/ch01_Long_v40-proof.pdf

[8] RIZON, Mohamed; YAZID, Haniza; SAAD, Puteh; SHAKAFF, Ali; SAAD, Abdul; MAMAT, Mohd; YAACOB, Sazili; DESA, Hazri; KARTHIGAYAN, M. - Object Detection Using Geometric Invariant Moments, [Online document], American Journal of Applied Sciences, Science Publications, 2006. Available at http://www.scipub.org/fulltext/ajas/ajas361876-1878.pdf

[9] RANGAYYAN, Rangaraj; GULIATO, Denise; DE CARVALHO, Juliano; SANTIAGO, Sérgio – Feature Extraction From the Turning Angle Function for the Classification of Contours of Breast Tumors, [Online document], 2008. Available at http://medlab.cs.uoi.gr/itab2006/proceedings/Mammography/56.pdf

[10]FOLKERS, Andre; SAMET, Hanan – Content-based Image Retrieval Using Fourier Descriptors on a Logo Database, [Online document], Institute for Signal Processing, Medical University of Lubeck and Computer Science Department, University of

Maryland at College Park.2002. Available at
http://www.cs.umd.edu/~hjs/pubs/icpr02-folkers-paper.pdf

[11]PETKOVIÉ, Tomislav; KRAPAE, Josip – Technical Report Shape description with Fourier descriptors, [Online document], 2002. Available at http://ipg.zesoi.fer.hr/petkovic/publications/Shape%20Description%20with%20Fourier%20Descriptors.pdf

[12]ZHANG, Dengsheng; LU, Goujun – Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures, [Online document], Gippsland School of Computing and Information Technology, Monash University, 2008. Available at http://knight.cis.temple.edu/~lakaemper/courses/cis595_2004/papers/fourierShape.pdf

[13]FARID, Hany – Fundamentals of Image Processing, [Online document], 2008. Available at http://www.cs.dartmouth.edu/farid/tutorials/fip.pdf

[14]BUNKS, Carey – Grokking the GIMP, [Online Document], Sams, 2000. Available at http://gimp-savvy.com/BOOK/index.html?node54.html

[15]HIPR2 – Image Processing Learning Resources, [Online]. Available at http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm

[16]Cambridge in Color – Digital Image Interpolation, [Online]. Available at http://www.cambridgeincolour.com/tutorials/image-interpolation.htm

[17]CoderSource.net – Image Processing XOR & XNOR, [Online]. Available at http://www.codersource.net/csharp_image_processing_xor.aspx

[18]Efg's Computer Lab – Image Processing Algorithms, [Online], 2003. Available at http://www.efg2.com/Lab/Library/ImageProcessing/Algorithms.htm

[19]Wikipedia – Image Retrieval, [Online], 2008. Available at http://en.wikipedia.org/wiki/Image_retrieval

[20]Wikipedia – Content-Based Image Retrieval, [Online], 2008. Available at http://en.wikipedia.org/wiki/Content-based_image_retrieval

[21]Wikipedia – Canny Edge Detector, [Online], 2008. Available at http://en.wikipedia.org/wiki/Canny_edge_detector

[URL01] http://iss.umwelt-campus.de/iss/index.php

[URL02] http://code.google.com/p/aforge/

ANNEX A: Performed tests and results

This annex presents the application of the methods presented on this report document and their results.

General pre-processing results

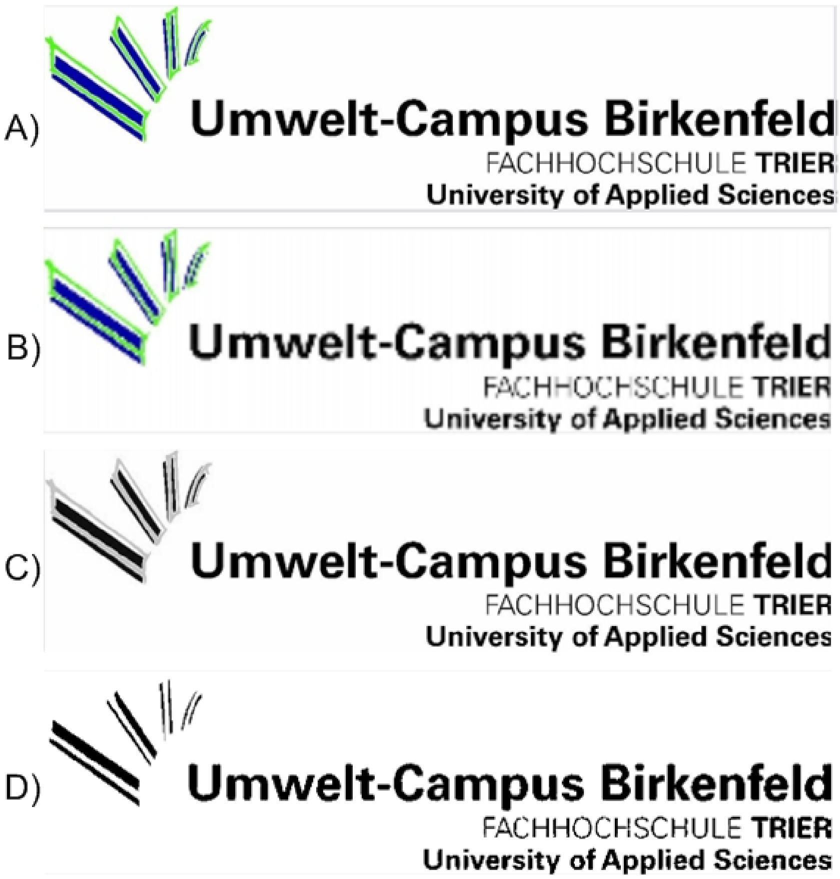


Figure 24: A) Original image B) Image resized to ¼ of its original size C) Original image converted to grayscale colour space D) Binary conversion using a threshold value of 128

A) B) C) D)

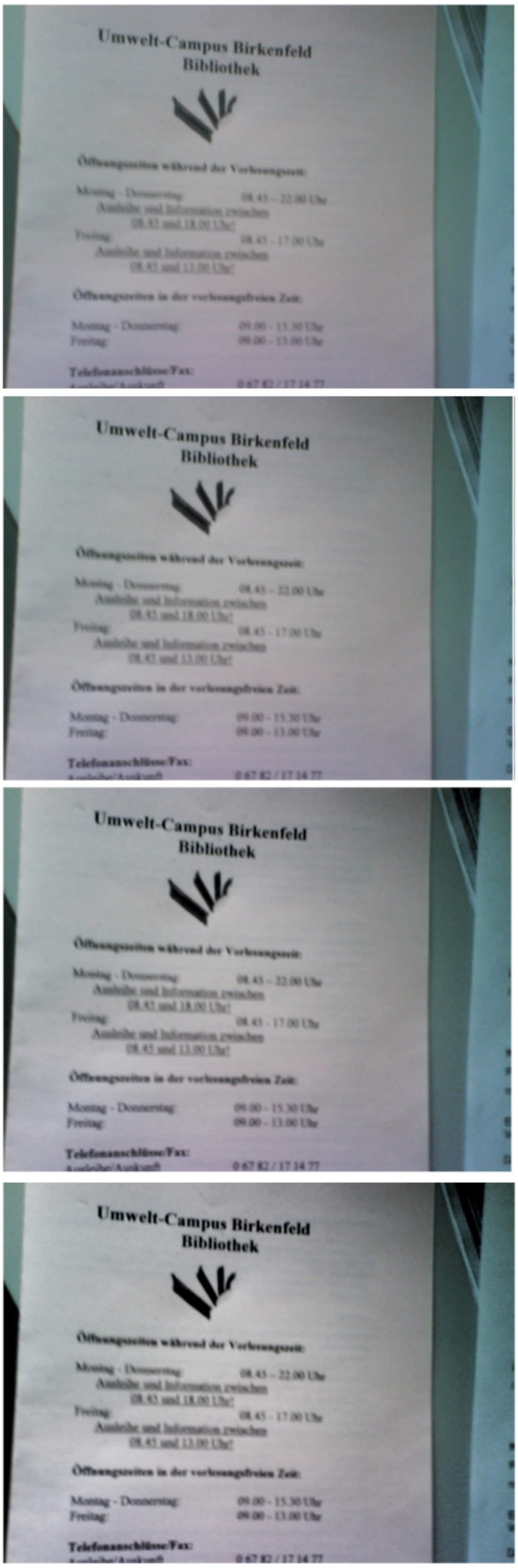
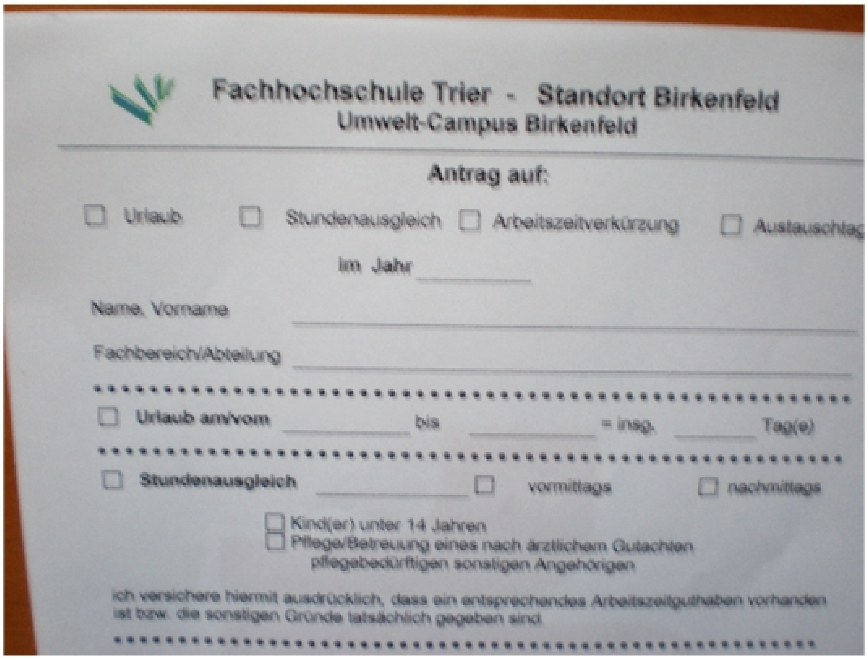


Figure 25: A) Original image B) The image after 1 iteration of contrast correction C) The image after 2 iterations of contrast correction D) The image after 3 iterations of the contrast correction

A)



Fachhochschule Trier - Standort Birkenfeld
Umwelt-Campus Birkenfeld

Antrag auf:

☐ Urlaub ☐ Stundenausgleich ☐ Arbeitszeitverkürzung ☐ Austauschtag

im Jahr ________

Name, Vorname ________

Fachbereich/Abteilung ________

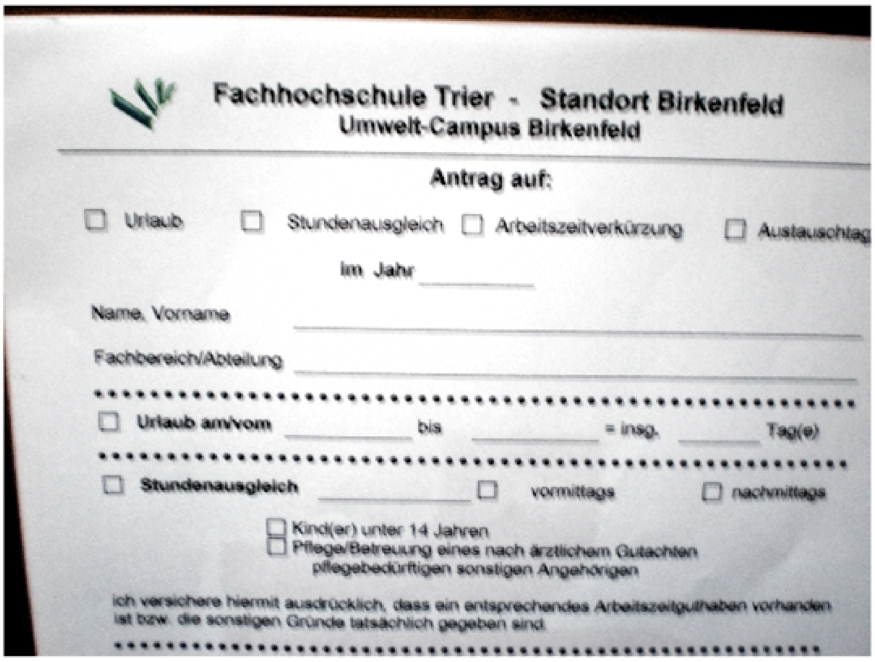
☐ Urlaub am/vom ________ bis ________ = insg. ________ Tag(e)

☐ Stundenausgleich ________ ☐ vormittags ☐ nachmittags

☐ Kind(er) unter 14 Jahren
☐ Pflege/Betreuung eines nach ärztlichem Gutachten pflegebedürftigen sonstigen Angehörigen

Ich versichere hiermit ausdrücklich, dass ein entsprechendes Arbeitszeitguthaben vorhanden ist bzw. die sonstigen Gründe tatsächlich gegeben sind.

B)



Fachhochschule Trier - Standort Birkenfeld
Umwelt-Campus Birkenfeld

Antrag auf:

☐ Urlaub ☐ Stundenausgleich ☐ Arbeitszeitverkürzung ☐ Austauschtag

im Jahr ________

Name, Vorname ________

Fachbereich/Abteilung ________

☐ Urlaub am/vom ________ bis ________ = insg. ________ Tag(e)

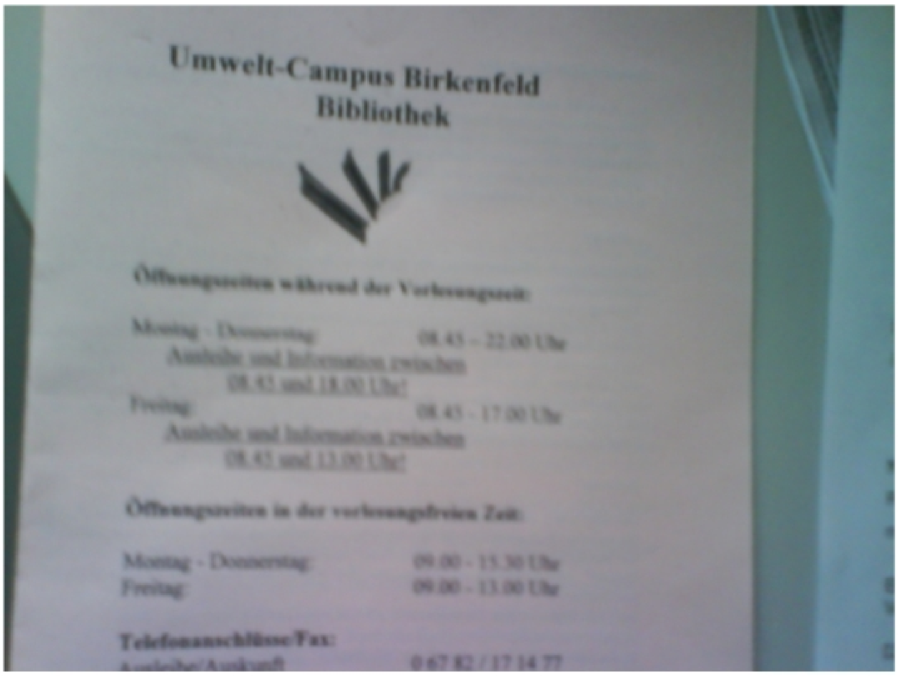
☐ Stundenausgleich ________ ☐ vormittags ☐ nachmittags

☐ Kind(er) unter 14 Jahren
☐ Pflege/Betreuung eines nach ärztlichem Gutachten pflegebedürftigen sonstigen Angehörigen

Ich versichere hiermit ausdrücklich, dass ein entsprechendes Arbeitszeitguthaben vorhanden ist bzw. die sonstigen Gründe tatsächlich gegeben sind.

Figure 26: A) Original image B) Image after 4 iterations of contrast correction

A)



B)

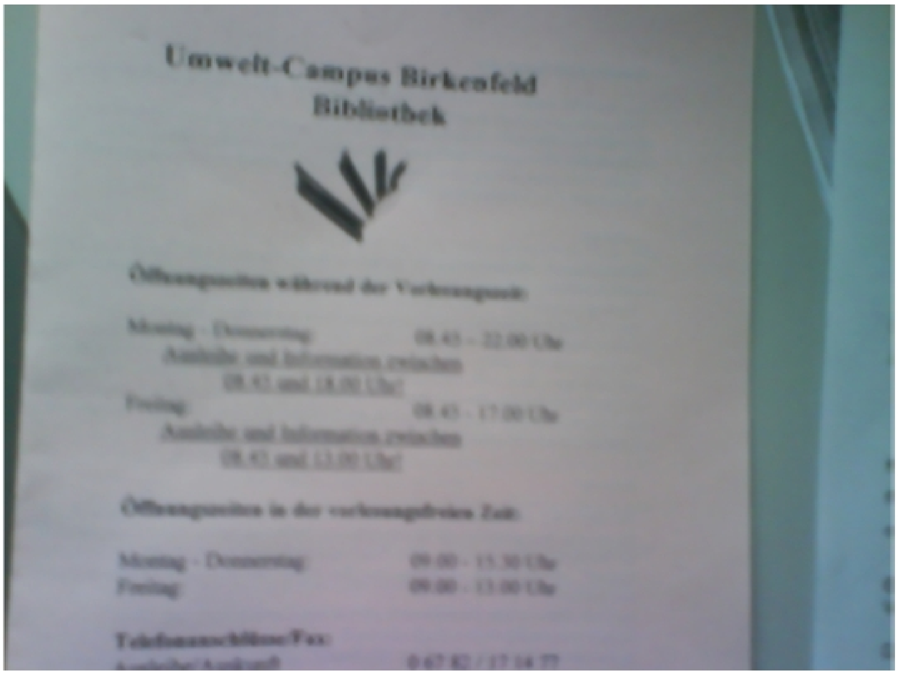


Figure 27: A) Original image B) Image after median filtering application

Colour feature extraction and similarity retrieval results

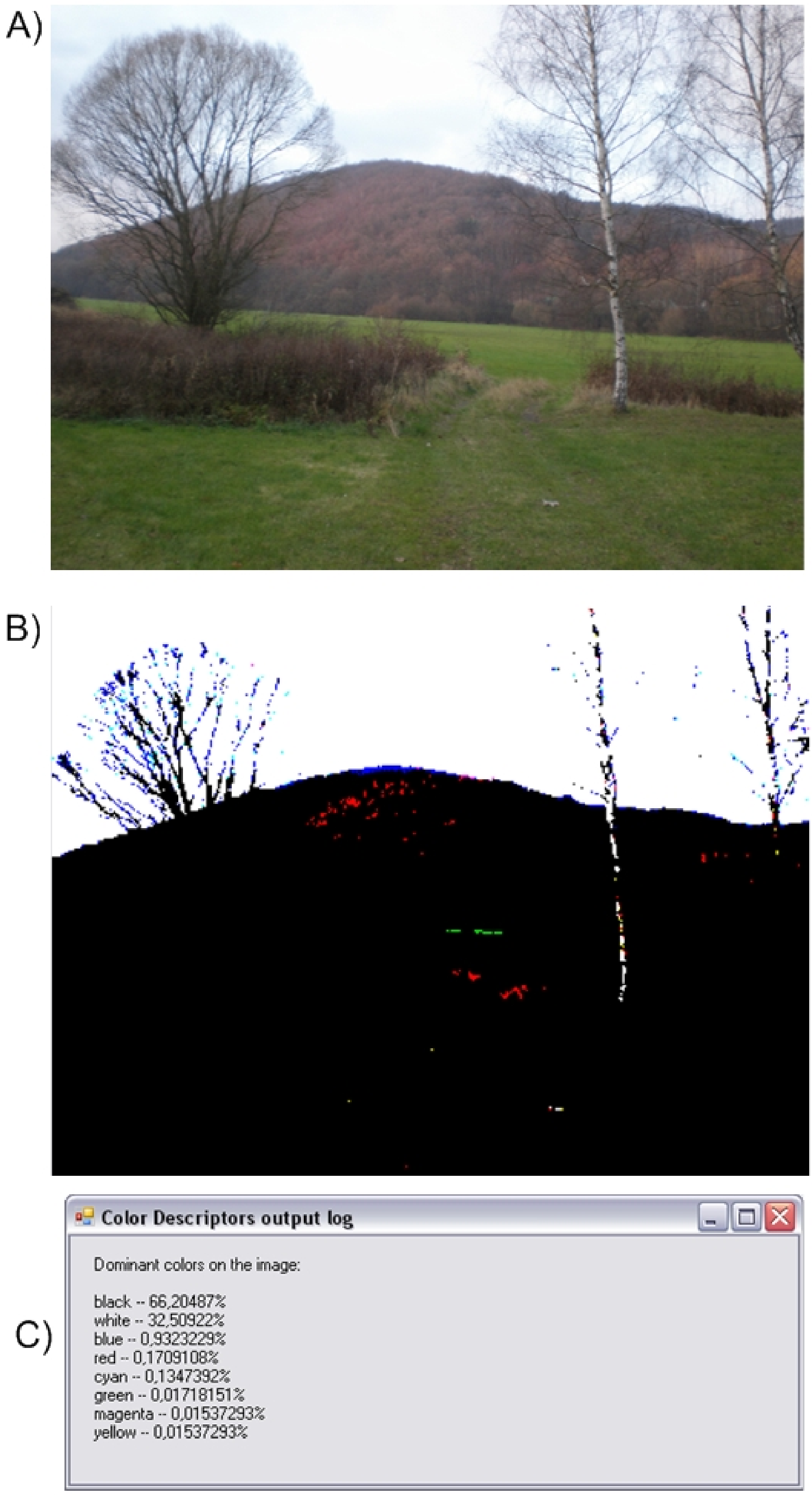


Figure 28: A) Image resized to ¼ of its original size B) Colour descriptors' mapping on the image with 2x2 cluster size C) Output log for the colour feature vectors of the image

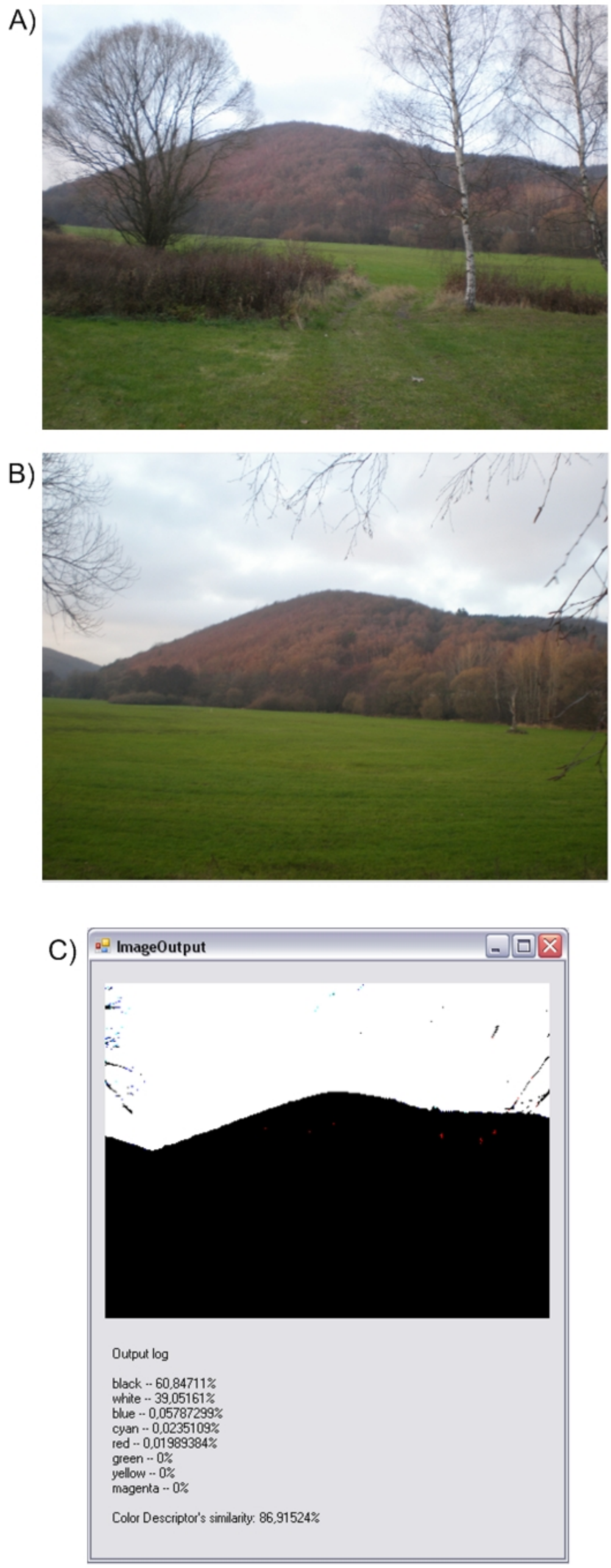


Figure 29: A) Query image resized to ¼ of its original size B) Database image resized to ¼ of its original size C) colour feature vector and similarity results from the comparison between the database image and the query image

Canny Edge Detector results

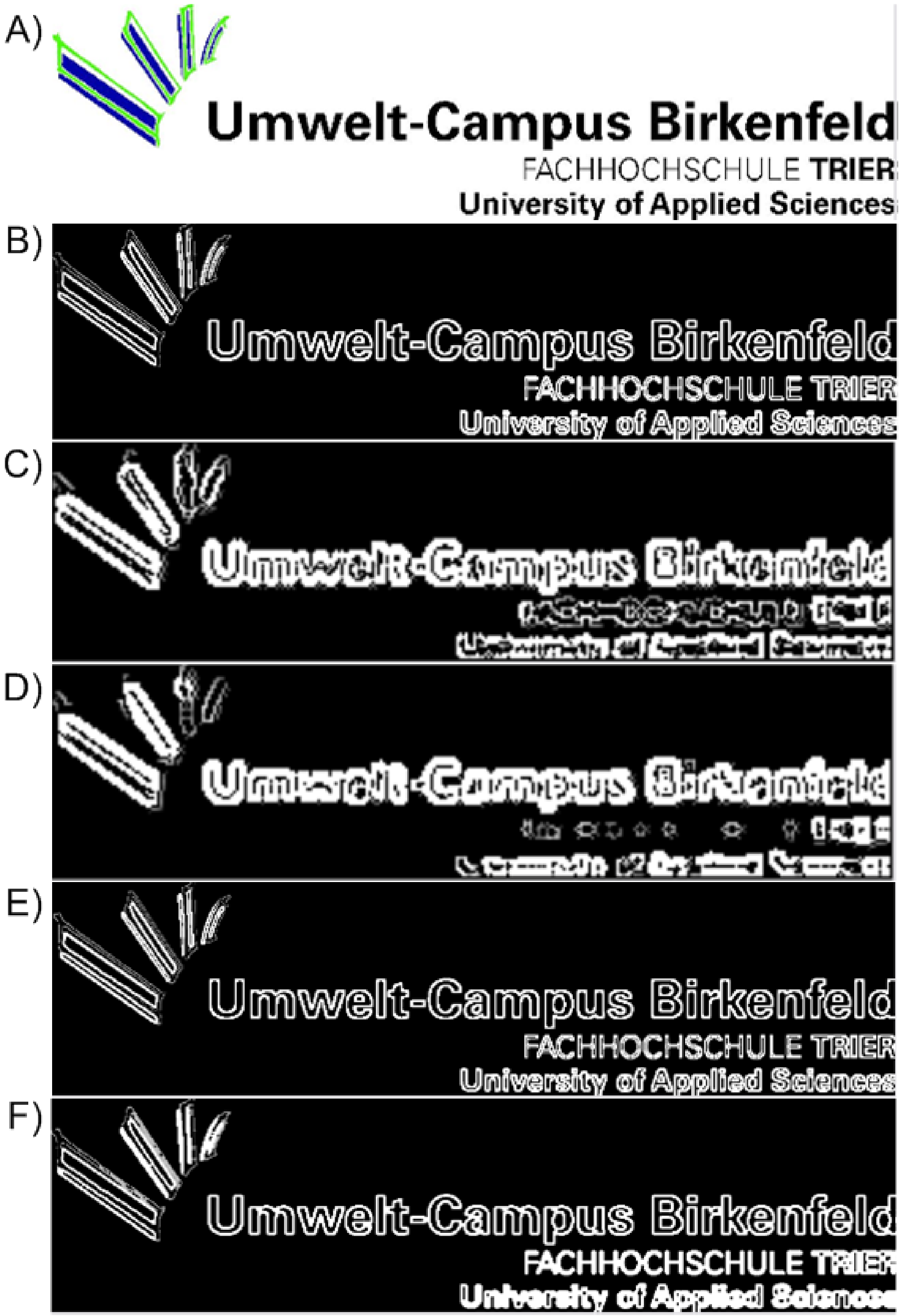


Figure 30: A) Original image B) Image after applying the Canny Edge Detector without any pre-processing C) Image after being resized to ¼ of its original size and then having the Canny Edge Detector applied D) Canny Edge Detector applied after resizing and applying the median filter E) Canny Edge Detector applied after applying 2 iterations of contrast correction to the image F) Closing filter applied after the edge detection

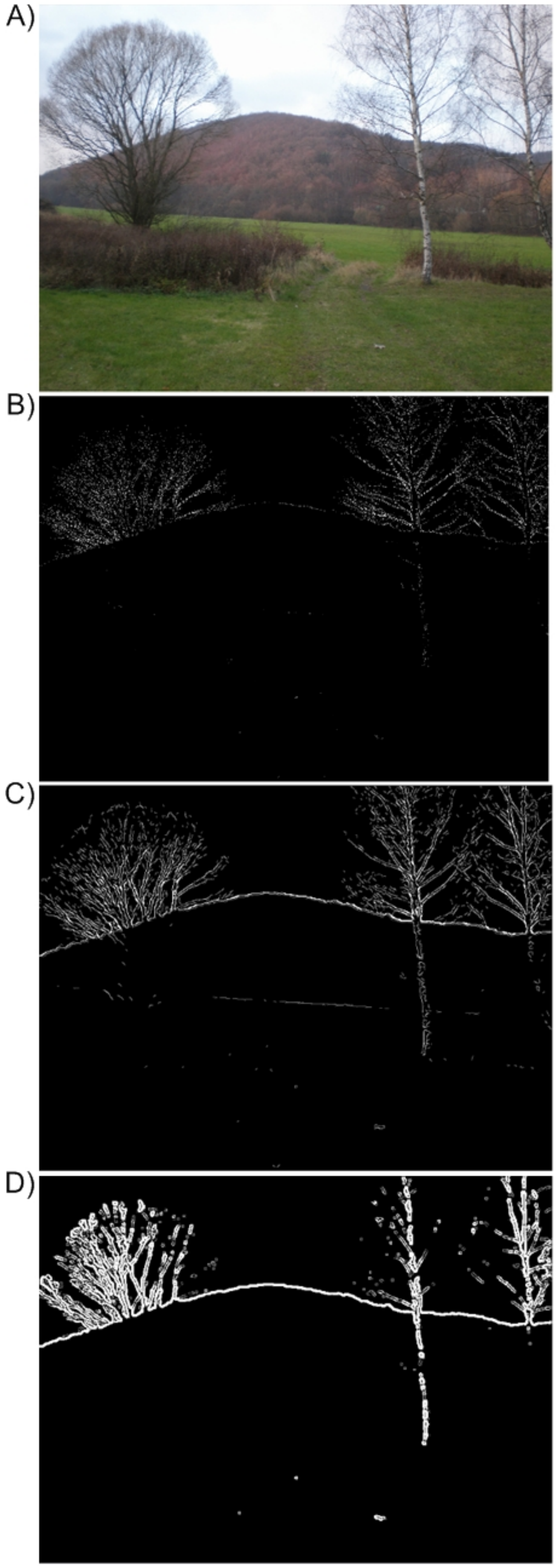


Figure 31: A) Original image B) Image after performing the Canny Edge Detector without any pre-processing C) Image after resizing to ¼ of its original size and applying the Canny Edge Detector D) Image after applying the grayscale and binary conversions with threshold of 128 amd the Canny Edge Detector

Noise removal results

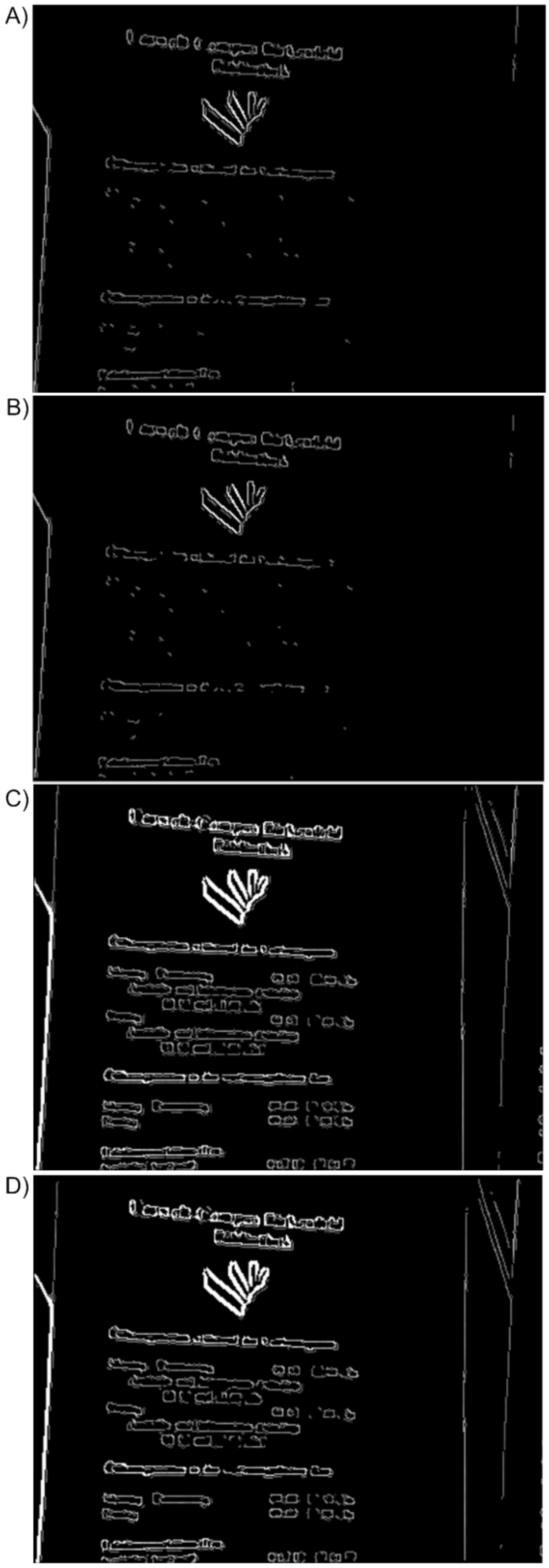


Figure 32: A) Image with edge detection performed and noise presence B) Image after independent noise removal with a 3x3 window C) Image after 3 iterations of contrast correction and edge detection D) The image in C) after independent noise removal with a 3x3 window

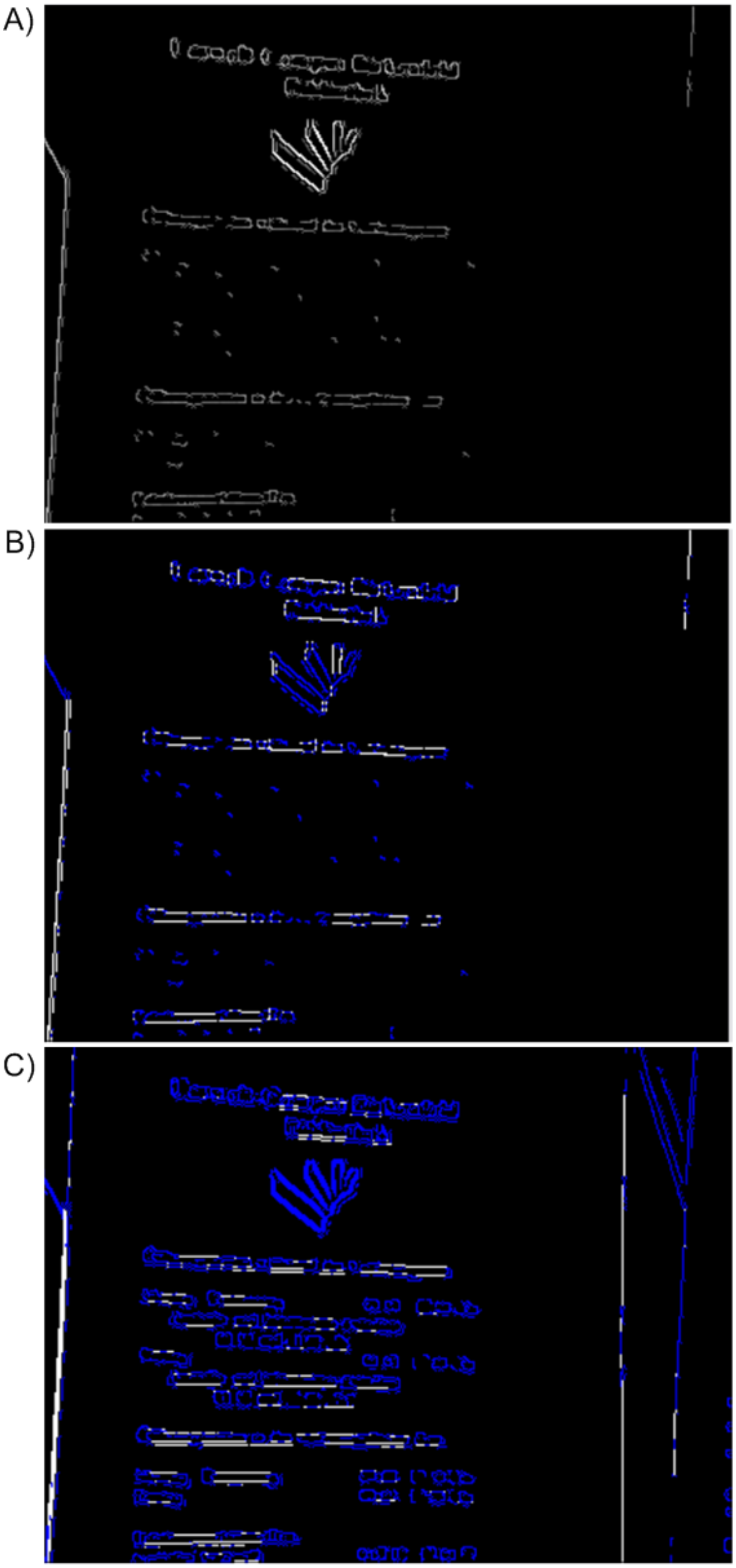


Figure 33: A) Image with noise presence after edge detection B) Image after statistically defining noise with marked pixels on blue C) Image after 3 iterations of contrast correction, edge detection and noise statistically defined with noise pixels marked with blue

Image object and shape recognition results

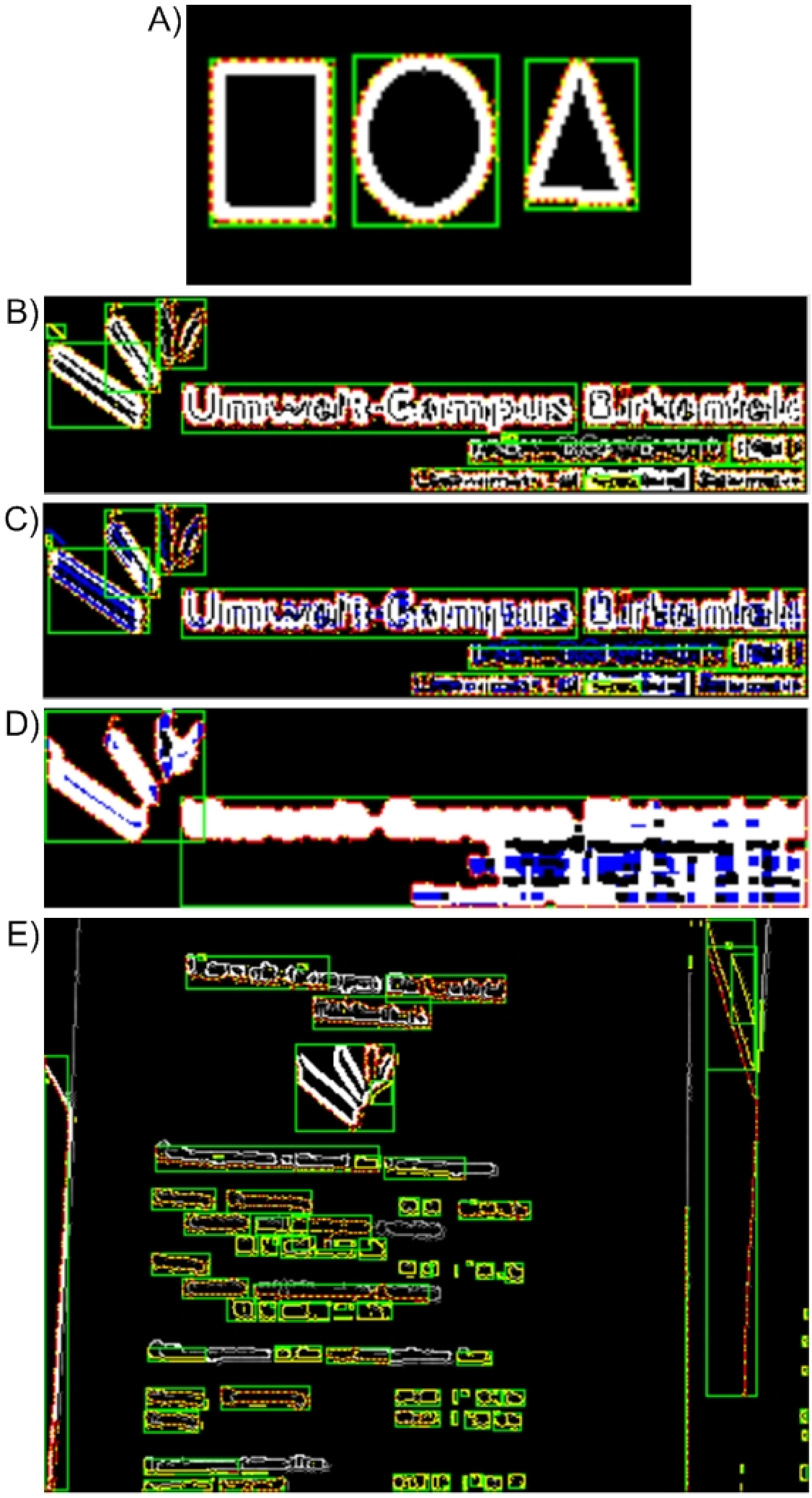


Figure 34: A) A simple example, 3 objects on the same image with no pre-processing applied B) 17 objects on an image with ¼ of its original size C) The same image in B) but with the noise defined statistically and 11 objects detected D) The same image in the same circumstances as in C) but with closing filter applied and 2 objects detected E) Another example with 3 iterations of the contrast correction

Similarity retrieval using structure features results

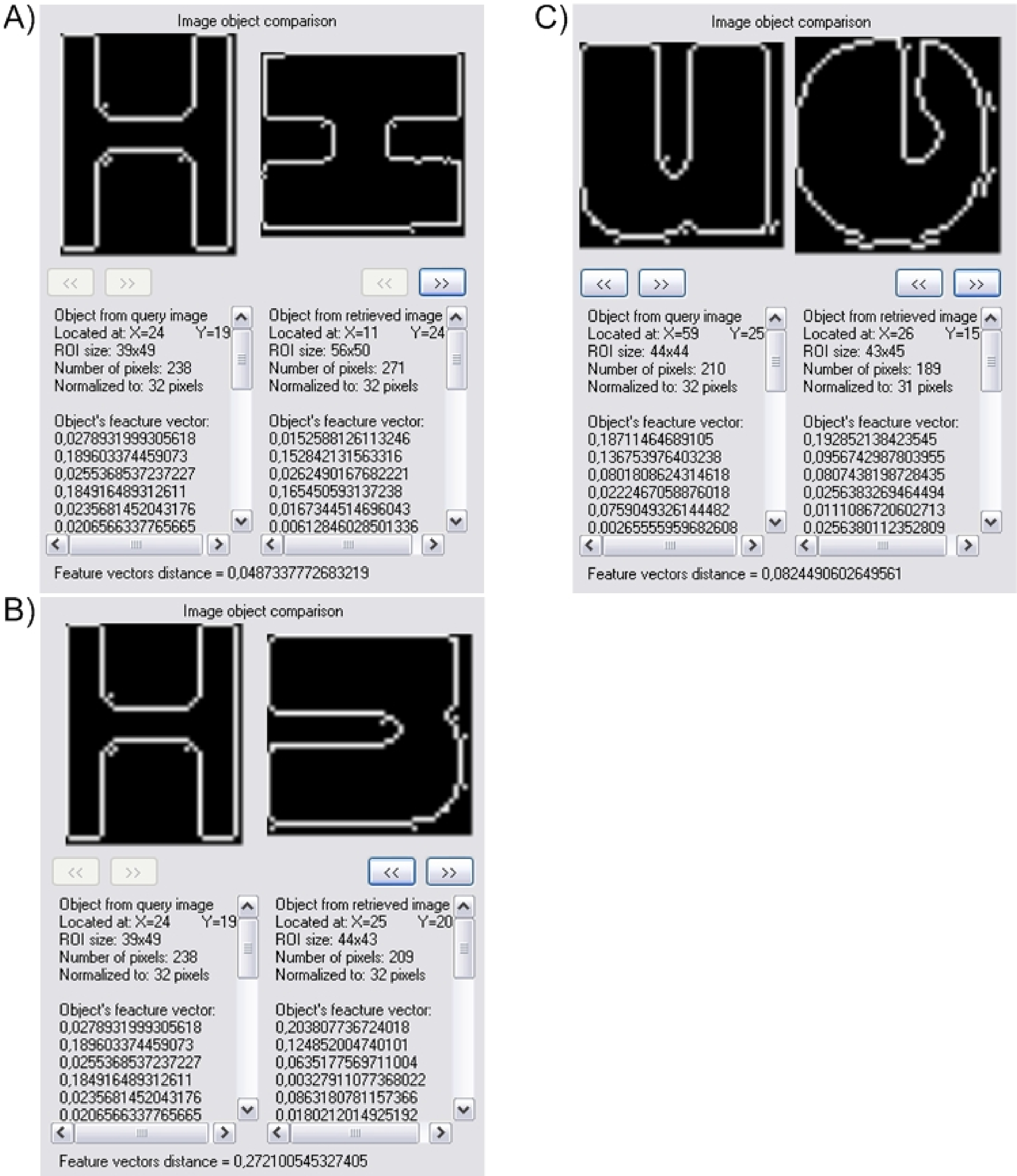


Figure 35: A) Comparison between two similar objects B) Comparison between two different objects C) Wrong measurement of similarity due to ambiguity of shape representation

Results on images with similar objects comparison

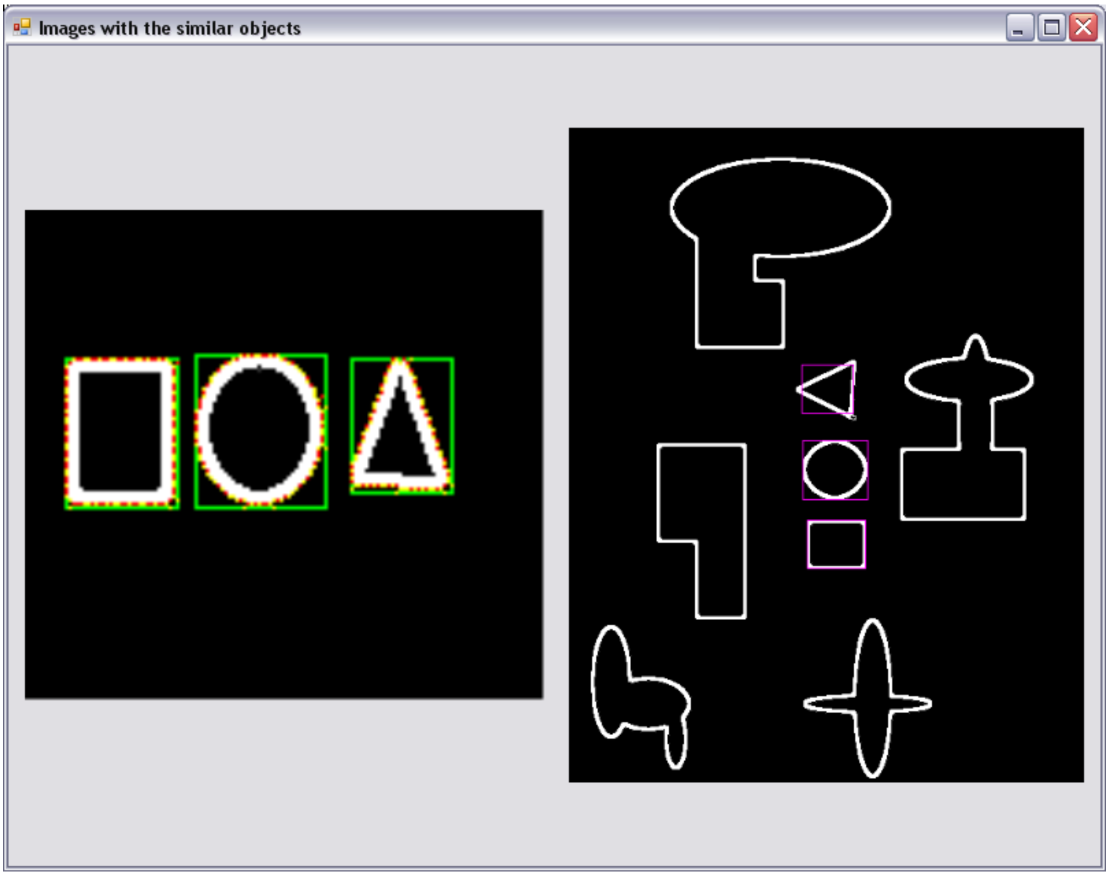


Figure 36: Query image and database with image objects and shape processed and normalized without any pre-processing. The image objects on the query are successfully identified among the objects the right

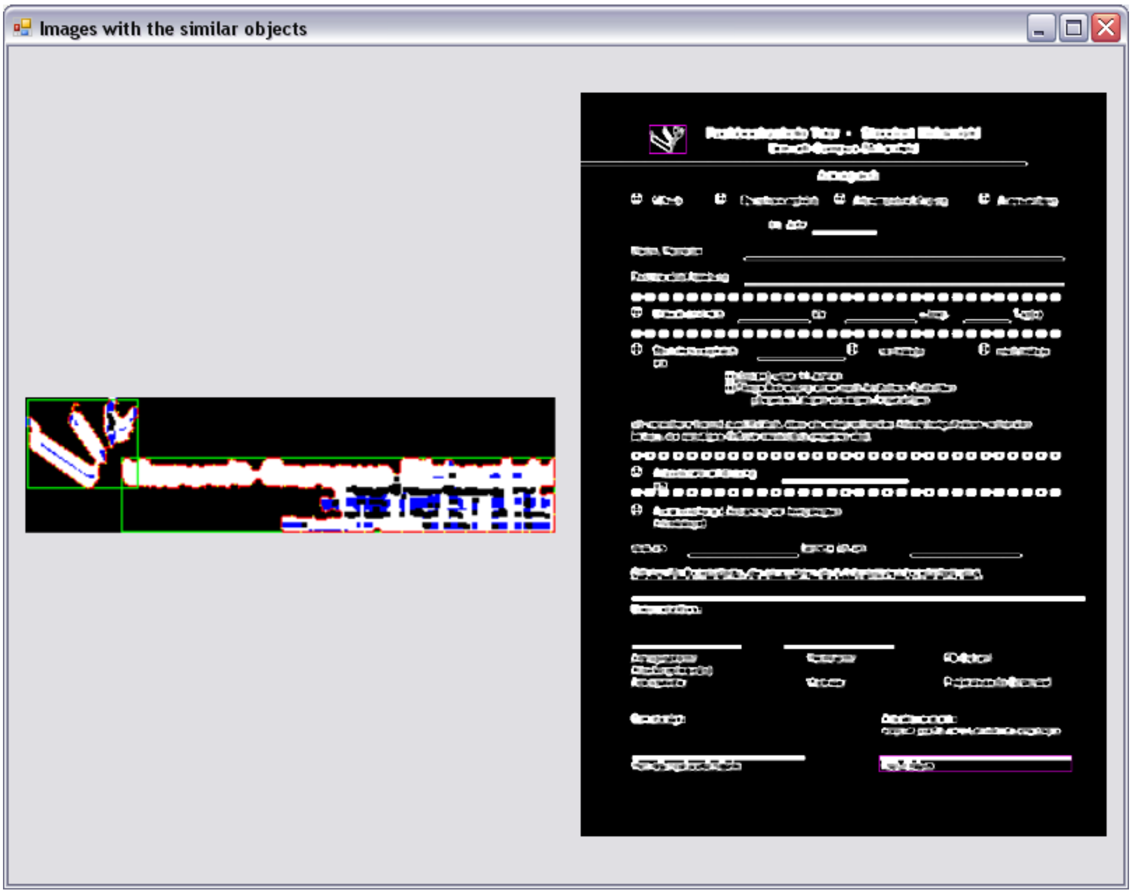


Figure 37: An attempt to try to find a logo within a real template. The logo is composed by 2 objects, a drawing and text. The drawing is found successfully but the text object matched another similar shape. The pre-processing sequence used for both was resize to ¼, edge detection, closing filter, noise removal by statistically defining noise and shape detection