TAMPERE UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

STEFAN UHLMANN

# Region-based Multimedia Indexing and Retrieval Framework

MASTER OF SCIENCE THESIS

Subject approved in the Department Council meeting on 11 September 2006

Examiners: Prof. Moncef Gabbouj

Dr. Serkan Kiranyaz

# Preface

This Thesis has been carried out in the Institute of Signal Processing, Tampere University of Technology, Finland as part of the MUVIS project

First and the foremost I wish to express my deepest gratitude to my supervisors Professor Moncef Gabbouj and Dr. Serkan Kiranyaz for providing me with the opportunity to work on this project and for their constant guidance, support, and fruitful discussions for the duration of this thesis project. My general thanks are also given to the whole MUVIS Team and the people of the Signal Processing Institute for the always friendly and productive working environment. I would also like to thank the Institute of Signal Processing of Tampere University of Technology for their financial support.

This work is the conclusion of a valuable learning stage, which started at my home university, FH Brandenburg. At this point I would like to acknowledge their contribution to my personal development in the scientific field.

A special thank you goes out to my friends in Finland, these last years have been great in your company, as well as to my friends back home in Germany who were always a magnificent source of motivation and inspiration. Thank you for being part of my life.

Finally, I want to thank my dear family, especially my parents and brother, for their endless love, care and support during my studies where ever I have been. I love you.

Tampere, May 2007.

Stefan Uhlmann

Insinöörinkatu 60 A 18

33720 Tampere, FINLAND

# ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Department of Information Technology

Institute of Signal Processing

**Uhlmann, Stefan:** Region-based Multimedia Indexing and Retrieval Framework

Master of Science Thesis, 103 pages

Examiner: Prof. Moncef Gabbouj and Dr. Serkan Kiranyaz

Funding: Academy of Finland Centre of Excellence in Signal Processing, Tampere University of Technology

June 2007

Keywords: image indexing and retrieval, CBIR, MPEG-7, MUVIS, region-based description, spatial information, feature extraction, segmentation.

Many systems have been proposed for automatic description and indexing of digital data, for posterior retrieval. One of such content-based indexing-and-retrieval systems, and the one used as a framework in this thesis, is the MUVIS system, which was developed at Tampere University of Technology, in Finland. Moreover, Content-based Image Retrieval (CBIR) utilising frame-based and region-based features has been a dynamic research area in the past years. Several systems have been developed using their specific segmentation, feature extraction, and retrieval methods.

In this thesis, a framework to model a regionalised CBIR framework is presented. The framework does not specify or fix the segmentation and local feature extraction methods, which are instead considered as "black-boxes" so as to allow the application of any segmentation method and visual descriptor. The proposed framework adopts a grouping approach in order to correct possible over- segmentation faults and a spatial feature called region proximity is introduced to describe regions topology in a visual scene by a block-based approach.

Using the MUVIS system, a prototype system of the proposed framework is implemented as a region-based feature extraction module, which integrates simple colour segmentation and region-based feature description based on colour and texture. The spatial region proximity feature represents regions and describes their topology by a novel metric proposed in this thesis based on the block-based approach and average distance calculation.

After the region-based feature extraction step, a feature vector is formed which holds information about all image regions with their local low-level and spatial properties. During the retrieval process, those feature vectors are used for computing the (dis-) similarity distances between two images, taking into account each of their individual components. In this case a many-to-one matching scheme between regions characterised by a similarity maximisation approach is integrated into a query-by-example scheme.

Retrieval performance is evaluated between frame-based feature combination and the proposed framework with two different grouping approaches. Experiments are carried out on synthetic and natural image databases and the results indicate that a promising retrieval performance can be obtained as long as a reasonable segmentation quality is obtained. The integration of the region proximity feature further improves the retrieval performance especially for divisible, object-based image content.

Finally, frame-based and region-based texture extraction schemes are compared to evaluate the effect of a region on the texture description and retrieval performance utilising the proposed framework. Results show that significant degradations over the retrieval performance occur on region-based texture descriptors compared with the frame-based approaches.

# Table of Contents

# List of Tables

# List of Figures

# Abbreviations and Acronyms

| | |
|---|---|
| **2D** | 2 Dimensional |
| **3D** | 3 Dimensional |
| **AFeX** | Audio Feature Extraction |
| **API** | Application Programmer Interface |
| **CBIR** | Content-based Image Retrieval |
| **CBMR** | Content-based Multimedia Retrieval |
| **CCV** | Colour Coherent Vector |
| **CoM** | Centre of Mass |
| **CPU** | Central Processing Unit |
| **D** | Descriptor |
| **DC** | Dominant Colour |
| **DLL** | Dynamic Link Library |
| **DS** | Description Scheme |
| **DWT** | Discrete Wavelet Transform |
| **FeX** | Feature Extraction |
| **FV** | Feature Vector |
| **GLCM** | Grey-Level Co-occurrence Matrix |
| **HVP** | Human Visual Perception |
| **HVS** | Human Visual System |
| **IEC** | International Electrotechnical Commission |
| **ISO** | International Organization for Standardization |
| **JSEG** | Unsupervised color-texture segmentation; Vision Research Lab, UCSB |
| **KMCC** | K-Means with Connectivity Constraint algorithm |
| **kNN** | k-Nearest Neighbour |
| **LBP** | Local Binary Pattern |
| **MAM** | Metric Access Method |
| **MM** | Multimedia |
| **MPEG** | Moving Picture Experts Group |
| **NQ** | Normal Query |
| **P** | Precision |
| **PQ** | Progressive Query |

| | |
|---|---|
| **QbE** | Query By Example |
| **QbR** | Query By Region |
| **QbS** | Query By Sketch |
| **R** | Recall |
| **RAF** | Region Area Factor |
| **ROI** | Region-Of-Interest |
| **SAM** | Spatial Access Method |
| **SD** | Similarity Distance |
| **SPMG** | Split and Merge segmentation |
| **mSPMG** | modified Split and Merge segmentation |
| **SQ** | Selective Query |

# 1 Introduction

Archival storage (filing) and finding (retrieving) of information (data) exists for several thousand years and both go hand in hand as a fundamental part in human nature. Over the years, the amount of information grew continuously. Thus, in the earlier years, looking for some specific information implied searching manually through data even though indexing technique already existed. This means laborious and time-consuming work already for small data collection, not to mention larger ones. Besides searching by oneself, one could also look for or ask someone who knows this particular field to assist and find the desired information faster.

In the modern age, computers and electronic storage media helped to record data and information more efficiently such as in database systems. Furthermore, this allowed replacing the manual approach by an automatic search (retrieval). However, the automatic search is in need of a suitable retrieval interface. Although the manual search is slow, it has the advantage that a human is conducting it and vice versa for the automatic search, i.e., it is fast but lacks semantic knowledge, which only humans can provide and understand. Two challenges come with the user interaction of the electronic retrieval system compared to searching conventional databases. The first may be described as fuzziness because the user does not precisely know how to express the information he/she is looking for. Hence, queries include vague conditions. The second may be circumscribed as uncertainty where the system does not have the knowledge about interpreting the content of the data. Thus, this leads to inaccurate and missing results. Therefore, the retrieval system has to provide a user-friendly interface to support the user in its needs for an efficient retrieval.

With increasing computerisation, more and more data are available and stored digital form, which made it necessary to search huge databases and data collections such as the WWW more efficiently and effectively. At the beginning of the WWW, there were no rules present for the definition of content and how to handle it. Over the years, the WWW changed to an information medium and searching those information, especially text, became easier due to search engines such as Yahoo, MSN, and Google.

However, information retrieval by text is not as easy as one might think because it is more than just matching words, phrases, or sentences. The issue here is that words can have

different meanings in different correlations such as homograph (word written in the same way with different meaning) and synonym (words have the same meaning but written differently). How often does it happen that one searches for something specific in the WWW by using one of those search engines but does not get any relevant results, only the words seem to match but not their actual meaning of context one had in mind. This is due to the aforementioned fuzziness and uncertainty because, generally, text based retrieval is not semantic which means that a search engine, a system, or a machine does not know what these symbols, character or numbers mean. Therefore, it is mainly just a symbol matching and hardly a semantic content matching by those search engines even though retrieval and its results have improved over the past years but are still far from optimal.

Besides text, huge interest grows in searching multimedia content such as images and videos due to increasing digitalisation based on the soaring number of digital devices (mobile phone, camera, etc.), which capture and store personal multimedia content. Moreover, the analogue audiovisual content from earlier ages is also converted into digital form. Hence, there is a demand to effectively store and organise such digital collections to support efficient queries and access schemes during retrieval, no matter if these collections are private or public such as Flickr [25] or YouTube [91].

However, this type of search has the same challenge as the text retrieval, i.e. lack of semantics or more precisely how to understand and describe the semantic content of a digital item? In order to accomplish this, two approaches can be considered: manual and automatic. For the manual case, the content description is performed entirely by humans. Here, the content is described by so-called *Tags*, which are set by user(s) who provide(s) the item (e.g. images for Flickr, video for YouTube). Its advantage is that the image or video is provided with a semantic content description due the human interaction. But this also bears a couple of risks. Firstly, to find an image or video, it has to have *Tags* and, secondly, these *Tags* have to provide a relevant and meaningful description. On the other side, the automatic case normally tries to describe the image content without any human interaction or intervention. Note further that these two approaches are not strictly separated and might be used simultaneously.

The field of Content-Based Multimedia Retrieval (CBMR) with its branch Content-Based Image Retrieval (CBIR) aims to provide options and possibilities to search and retrieve digital multimedia data in such an automatic way. The main idea is the following; based on a given query and its content find items representing the same or similar content as

efficient and effective as possible. Several questions may be raised such as “What is the content?”, “What are relevant parts or objects?”, and “How to describe those objects?”. All such simple questions can be answered with ease by humans thanks to our intelligence, learning capabilities, and life-long experience about objects and their logical relations among each other. The challenge in CBMR is to gain semantic (high level) content description based on the low-level representation, digital signal in bits. Just based on those “bits”, it is difficult -if not infeasible- at the current state to extract any semantic information of the content. Furthermore, the human interpretation and utilisation of the semantic content in audio-visual data for the content-based retrieval purposes are closely related. Based on this fact, humans tend to search in a pragmatic manner where parts or objects are usually more important then the entire audio-visual signal. Therefore, instead achieving such level of semantic description, CBMR and CBIR employ rather low-level approaches for content description focusing on the low-level visual cues such as colour, shape, texture, motion, etc.

By examining the scenery of Figure 1.1, what would be the best approach to describe the image content based on low-level properties? Would it be globally, describing the content as a mixture of all information, or locally, describing the content by individual parts with their specific low-level properties. Using a local content description over a global one might be preferred since it is suitable for an enhanced modelling of human perception especially if accurate and meaningful regions (objects) may be provided by pre-processing steps such as segmentation. Moreover, besides describing the actual regions better based on their local properties (features), this further allows the integration of region relationships into the content description. However, note that this may provide a better content description but will still lack the semantic meaning of the regions.

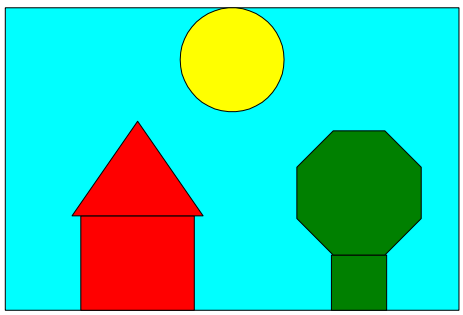

**Figure 1.1 - A pictorial image**

Over the years, many CBIR systems such as QBIC [21], Mars [71], PicToSeek [29], VisualSeek [82], Blobword [9], Netra [56], Windsurf [4], and SIMPLicity [88] have been developed, which employ such global and local content description. Moreover, there are systems such as MUVIS [67], which provides a framework for indexing and retrieval that is easily extendable by any third party using their on description modules.

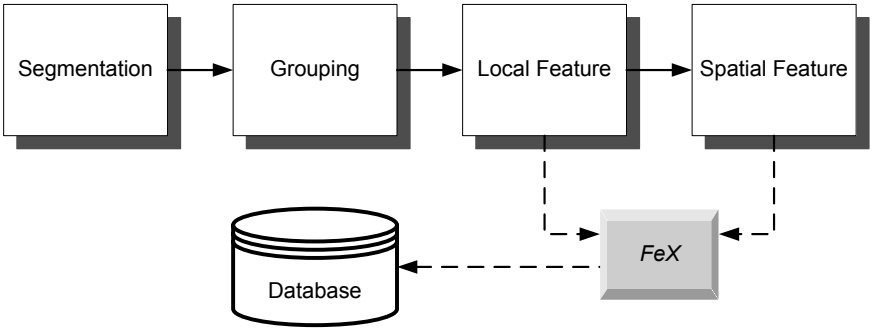


**Figure 1.2 - Basic schematic of the proposed framework**

In this thesis a local content description approach using image segmentation and regionalised description is employed. Furthermore, a spatial region description is integrated. These parts are combined into an adaptable regionalised CBIR framework. This framework employs four major parts for the content description namely, *segmentation*, region *grouping*, *local* and *spatial feature* extraction as illustrated in Figure 2.1. The framework structure allows the integration of *segmentation* and *local feature* extraction parts as black-boxes, so that any segmentation algorithm and low-level feature might be performed, extracted, and tested. Region *grouping* intents to correct the over-segmentation faults possibly encountered from the underlying segmentation method and *spatial feature* is used for describing the spatial layout of regions via introducing a block-based proximity computation. In order to overcome the limitations of the Hausdorff distance [38], a novel and robust approach is applied over the block-based computation. By including a matching scheme incorporating all regions into the retrieval process, the similarity between two images is based on a region similarity maximisation approach. Here, region-similarity might come from their local region properties, where two regions directly match by their low-level features (i.e. white regions in images (a) and (b) of Figure 1.3) or from their spatial region surroundings. For the surrounding similarity, two regions may not match in their local properties; however, their surrounding regions might be similar. In these cases the surrounded regions are then considered similar due to similar

region proximity compositions (i.e. mismatching white and black regions with similar neighbour regions in images (b) and (c) of Figure 1.3).


**Figure 1.3 - Example region similar by local and spatial features**

A prototype implementation integrates colour segmentation as well as colour and texture descriptors as region features. For extracting the region-based features, a back-projection approach is employed, which extracts features first frame-wise and then back-projects them onto the regions. Moreover, a simple texture indicator [43] is applied and adjusted for enhanced noise resistance and region application. It is used to check if texture extraction over a region is necessary and further to weight the influence of texture during retrieval. Because, if there is no texture in the region then there is no need to compute similarity from the texture feature while the region similarity should only depend on colour properties.

Integrating the entire framework into MUVIS due to its straightforward extendable framework structure allows indexing and retrieval in an efficient manner, which further allows the explicit testing and performance evaluation of the proposed approach. This is also the major reason for using the MUVIS framework as the chosen system to carry out this work. The proposed framework is then further used as a test-bed platform to perform texture-based image retrieval experiments via implementing three texture descriptors in order to study the effects of texture over the entire image and (texture) region(s). Retrieval performance is evaluated using ANMRR similarity metric both over frame- and region-based retrieval performance for synthetic and natural databases.

The rest of the thesis is organised as follows: In Chapter 2 an introduction for the current status of CBMR field is presented. MUVIS and other CBIR systems are presented in Chapter 3. The proposed framework and a prototype implementation are explained in Chapter 4 and Chapter 5 discusses experimental results obtained from various image databases. We will make the conclusive remarks and address some future work in Chapter 6.

# 2 CBMR - Content-Based Multimedia Retrieval

Content-based Multimedia Retrieval (CBMR) tackles the aforementioned challenges and became a large research field over the past years. The main goal is searching and finding similar MM items based on their content. In order to accomplish this, the content should first be described in an efficient way, e.g. the so-called indexing or feature extraction. Then fast and accurate retrievals among MM collections can be performed based on the content description. Figure 2.1 illustrates a general overview of a CBMR system where "offline" indexing phase is displayed in the bottom part and the "online" content-based retrieval is displayed in the upper part. Both phases interact with a collection of multimedia items (images, videos, etc.) from a multimedia database. During the retrieval, the user provides a query, which can be an example item, region, sketch, humming, or text.

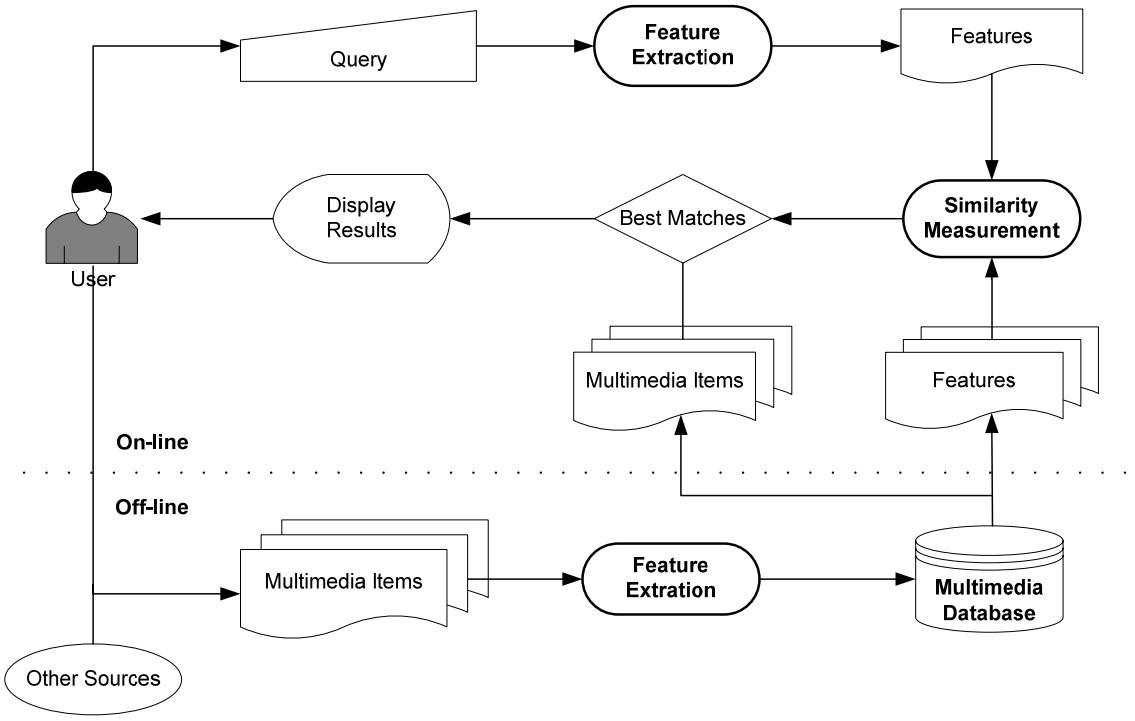


**Figure 2.1 - Overview of general CBMR structure**

In this chapter, a general introduction of CBMR is presented whilst the focus is particularly drawn on CBIR. The following subsection provides an overview of low-level visual descriptors and feature extraction schemes.

## 2.1 Visual Descriptors and Feature Extraction

One of the major challenges in CBIR is to describe semantic content (automatically) since "content" is a profound word in relationship with computers and data processing. Describing the content directly in a semantic (human like) way is complex and difficult due to the lack of knowledge and a sufficient level of intelligence. Therefore, CBMR for images and video mainly exploits the properties of human visual perception (HVP) via content descriptors, which are also called visual features. These features can be divided into two general types, low-level and high-level. The basic low-level visual features are colour, texture, and shape. With these, the content of a region of interest (ROI), which can be an image part or the entire image, is described based on pixel information using some basic properties. On the other hand, high-level features (concepts) provide a description of the content in a semantic manner presenting true information about objects and their relationship among each other. As a result, between low- and high-level features there exists the so-called *semantic gap*, which can be defined by "the lack of coincidence between the information that one can extract from the visual (multimedia) data and the interpretation that the same data have for a user in a given situation" as stated in [80]. Here, the lack of describing high-level concepts by low-level features was and still is the main challenge to master since nowadays CBMR systems need to map those low-level features to high-level concepts in order to improve the content-based retrieval accuracy.

The extraction of low-level visual features is an essential part in CBIR systems. Traditional approaches such as [21], [29], [71], [73], and [87] compute global (frame-based) features of images. Here, the low-level image content is described as a mixture of different colours, textures, and edge directions. In contrast to those traditional approaches, region-based methods such as [4], [9], [22], [56], [68], and [88] extract features over segmented regions. The main intention of using region-based features is that user's perception of image content can be better modelled and, therefore, may be used to reduce the gap between low-level features and high-level concepts.

The notion of feature extraction (FeX) in a multimedia image database is a part of the indexing phase in CBMR. The general methodology for indexing is to represent database items in a multi-dimensional feature vector format, which might contain several feature types such as visual, aural, motion-based, etc., depending on the database items. Content-based similarity between two database items can naively be assumed to correspond to the (dis-) similarity distance of their feature vectors. Hence, the retrieval of a similar database

item with respect to a given query (item) can be transformed into the problem of seeking those database items that are represented by feature vectors, which are close to the query feature vector.
In the following subsections a summary will be provided for image segmentation, the most basic low-level features, their extraction and retrieval focused on CBIR.

### 2.1.1 Image Segmentation

Image segmentation is an essential preliminary step in most automatic image recognition and scene analysis operations. The main goal for segmentation is to partition an image into homogeneous, constituent, non-intersecting regions so that the union of two adjacent regions will not meet a homogeneity criterion. Here, homogeneity is achieved with respect to some pre-defined properties such as colour and/or texture. In [80] the concepts of strong and weak segmentation have been presented. Strong segmentation is defined as the ideal object segmentation where each region would correspond to one single semantic object. Weak segmentation is present whenever strong segmentation cannot be achieved. Here, homogeneous regions are obtained, which do not necessarily describe objects. Weak segmentation may also result into two erroneous outcomes: under- and over segmentation. Under segmentation means two regions with different properties have been merged. Over-segmentation means that a larger homogeneous region is partitioned into more than one segment. So far there is no segmentation method suitable for all image types such as general purpose images, aerial or satellite images, and medical images. Moreover, each method is not applicable to one particular image type. Hence, segmentation is mostly application driven and mainly depends on what the user wants to achieve.

The weak segmentation scheme is utilised in most image segmentation ideas and various methods exist in the literature applying this approach. One of the most classical approaches is the so-called *pixel based* segmentation. It is also known as segmentation by threshold or mode method. Different threshold detection methods are described in [60] and [3]. This image segmentation approach is mainly applied on grey level images using the image histogram representation [5] and segmentation is, therefore, achieved by partitioning this histogram using some thresholds. Therefore, regions should be characterised by peaks in the histogram separated by valleys.

Another approach is *edge-based* segmentation, such as the work presented by Iannizzotto and Vita [4]. It usually consists of two main steps. The first one is to detect the edges

within the image and the second one is to link those edges to continuous contours. As shown in [48], this approach may lead to a useful object extraction if the major object contours can be revealed by the initial edge detection algorithm.

In *region-based* approaches, segmentation is achieved by grouping similar pixels together. The process starts with the so-called seed regions and based on those seeds the regions are further developed. There are two main approaches: region-growing [1] and region-splitting [35]. Region-growing starts with a small set of seeds and then seed regions grow by including homogeneous neighbours until all pixels are processed. In region-splitting the first seed is the entire image. If it is inhomogeneous with respect to certain criterion, it is split into sub-regions and then the operation proceeds with each sub-region as a seed until all sub-regions meet the defined homogeneity criterion. After splitting, a merging phase is usually needed to prevent over segmentation. Some serious drawbacks, however, occur such as the need of suitable seed regions for region-growing and also the effects of block-based region boundaries over region-splitting.

A common image segmentation method is the *morphological watershed* [55]. Here, a grey tone image can be seen as topographic surface. This surface is then either flooded from the bottom or rain falls on it. The water will eventually fill the valleys (catchment basins) in the surface. At a certain water level, a dam is built to prevent the flooded basins to merge together. When the complete surface is flooded, only the dams (watershed lines) are left and provide the segmentation result. There exist different variations of the original method such as [44] , [62], and [75].

Others, such as [66], [78], and [23], employ graph or tree-based approaches where the image is represented as a graph (tree) and, segmentation is achieved by splitting the graph (tree) so that each final sub-graph (sub-tree) represents an image region.

Since modelling human perception is the ultimate goal in computer vision, strong segmentation would be desirable by any application. So far a generic object-driven segmentation does not exist. Thus, Zhou et. al [93] investigated if strong segmentation is possible from weak segmentation perspective. They introduce a segmentation entropy curve as a model representation to provide a simpler object description. Based on their results they believe that strong segmentation is feasible. This can be emphasised by recent methods where various researchers have tried to integrate image understanding and

semantics into the segmentation process, which improves segmentation quality as can be seen in [18], [20], [78], and [89].

### 2.1.2 Colour Descriptors in CBIR

Colour plays an important role in HVP, and it is probably the most dominant cue to be recognised in a picture (image). Therefore, colour also plays a significant role as a property (feature) in CBIR as various existing systems ([4], [9], [21], [29], [71], [73], and [88]) use colour for content retrieval. Colour can be represented in various domains [26] such as HSV, RGB, YUV, CIE-Lab, CIE-Luv, etc., each of which has its advantages and disadvantages for certain applications. Since in CBMR human visual perception plays an eminent role, a perceptually uniform colour space may prove useful. Accordingly, CIE-Lab and CIE-Luv are commonly applied. HSV space, which is quite straightforward for human interpretation, has the drawback of discontinuity whereas RGB space is not perceptually uniform at all.

Probably the simplest colour descriptor is the *colour average* as used in [88]. It describes a ROI by a single colour, which limits its utilisation mostly for regions because the average colour of an entire image would not be provide sufficient and meaningful description. However, besides its simplicity, it provides a small descriptor size.

One of the most popular colour descriptor in CBMR is the *colour histogram* and its variants as in [9], [21], [30], [67] [82], and [87]. Here, the image histogram for each colour space component is used and, further, quantised into bins to reduce the amount of colour levels from millions to thousands or even to just hundreds and less. This static bin quantisation serves as a compact representation of the colour content in a ROI and it is robust against translation and rotation of an image. Moreover, different scales and changes in the point of view only slightly degrade the histogram representation. A drawback of colour histograms, however, static quantisation scheme is usually applied where the colour bin boundaries are determined empirically. This means that different colours will be clustered into the same bin for coarse quantisation (few number of bins), which results in a poor description and limited discrimination. On the other hand, the more bins (fine quantisation) a colour histogram contains, the better colour clustering and discrimination power may be achieved. However, a histogram with a fine quantisation will cluster similar colours into different bins and a large number of bins will also increase the computational cost for indexing and retrieval.

Another colour descriptor in CBIR is *colour moments*. The basic assumption is that the distribution of colour in a ROI can be interpreted as a probability distribution, which can be described by mean, standard deviation, and skewness representing the first three moments, respectively. It is been successfully used in [21] especially if only an object is represented in the image. Furthermore, colour moments have been efficiently and effectively used in colour distribution representation as proven by [83].

*Colour codebooks* [56] are another type of descriptors where a certain number of colours is either manually selected [79] (e.g. the ones most distinguishable for humans) or determined by dynamic quantisation [63] to limit the amount of colour information. This approach brings colour description closer to human perception by specifying important colours. However, the manual method needs to define a certain amount of colours to avoid erroneous colour mapping. Moreover, unique colours with small proportion are considered in the description even though there are not relevant in human colour perception [64]. For the dynamic case, the number of colours is important where larger numbers will increase the computational complexity.

A compact descriptor is the so-called *colour set* also known as dominant colour descriptor (DCD) employed in [17], [22], [58], and [64]. The main idea behind this is to describe the prominent colours in the ROI where colours are dynamically clustered (i.e. by colour distortion and area until a certain number of clusters is reached). Moreover, it is further consistent to HVP as HVS mainly perceives dominant colours and discards the rest [64]. Due to this fact, it is sufficient to represent the colour content of an ROI by the few DCs present in the visual scenery.

In order to reduce the semantic gap allowing retrieval based on semantic colour names, appropriated colour names [54] can be used as a colour descriptor by mapping semantic colour names to their colour space values.

A way of incorporating spatial information into colour histograms was introduced by Pass [72]. They proposed the colour coherence vector (*CCV*) where each histogram bin is partitioned into two types: coherent if it belongs to a uniformly-coloured region or incoherent otherwise. A *CCV* for an image *I* can be defined as

$$CCV(I) = \langle (\alpha_1, \beta_1), (\alpha_2, \beta_2), ..., (\alpha_N, \beta_N) \rangle \tag{1}$$

where $\alpha_i$ and $\beta_i$ denote the number of coherent and incoherent pixels for the $i^{\text{th}}$ colour bin and $N$ is the number of histogram bins.

A colour descriptor utilising spatial information of image pixels is the colour correlogram [37]. Besides describing the colour distribution in an image, it also characterises the spatial correlation of colour pairs. A colour correlogram is represented by a table where the table entries are indexed by colour pairs. The $k^{th}$ entry for a pair (i,j) specifies the probability of finding a colour pixel j at a distance k from a colour pixel i in the image. Considering combinations of distances and colour pairs, the descriptor size will be huge. Therefore, a simplified version is usually used instead called colour auto-correlogram, which only considers spatial correlation between identical colours.

### 2.1.3 Texture Descriptors in CBIR

Besides colour, texture is another significant cue for the description of visual sceneries. Thus, it is also a popular and important feature in CBIR to describe content of general purpose images. So far, there is no unique definition for texture; however, an encapsulating scientific definition as given in [40] can be stated as, "Texture is an attribute representing the spatial arrangement of the grey levels of the pixels in a region or image". Utilising those grey level arrangements, several methods have been proposed to extract texture information mainly from the luminance part of a ROI. The common known texture descriptors are Wavelet Transform [81] and Gabor-filter [59], (as signal processing approaches), Local Binary Pattern [70] and co-occurrence matrices [33] (as statistical approaches), and Tamura features [85] (psychological).

Tamura proposed a texture description by developing features close to the human visual system. In his work, he defined six features, namely *coarseness, contrast, directionality, line-likeness, regularity,* and *roughness*, compared to psychological measurements for human subjects where the first three features result in the best performance observed in [36]. The most fundamental attribute used by Tamura was *coarseness***,** which represents the relation between scale and repetition rates. Thus, coarseness tries to recognise the largest texture scale that exists in an image by measuring notable variations of grey-levels in non-overlapping windows of different sizes (i.e. $2^k$ with $k \in [0,5]$). *Contrast* measures the variation of grey levels in an image and describes to what extend their distribution is biased to black and white. It is modelled by the ratio of standard deviation and kurtosis (the $4^{th}$-order moment) over the image grey levels. The main idea behind *directionality* is to capture distribution of oriented local edges against their directional angles. In order to accomplish this, edge detection with simple masks (e.g. Sobel-operator) is applied where

edge orientation (angle) and edge strength (magnitude) are calculated for each pixel. Then, a histogram is generated by thresholding magnitude and further quantising it by edge angles. The histogram will reflect information about the degree of directionality. Due to its psychological approach, Tamura properties represent texture close to HVP but extraction of those features might be computationally complex.

An earlier statistical approach was the grey-level co-occurrence matrix (GLCM) by Harlick. Here, the texture descriptor is gained from the $2^{nd}$ order statistic. The GLCM computed within the ROI provides information about angular and directional pixel relationships and further defines how often two grey level values $g_i$ and $g_j$ are separated by a certain distance vector. Hence, the combination of different distance vectors captures different existing texture properties. Based on the GLCM, features can be computed such as energy, entropy, contrast, and homogeneity, all of which describe the underlying texture properties. The feature vector size, therefore, depends on the range of distance vectors and the amount of properties calculated from the co-occurrence matrix. The texture description power of GLCM depends on the combination of selected distance vectors where too few will provide a rather poor description and too many will increase the computational costs during feature extraction.

Yet another approach is Local Binary Pattern (LBP). It works directly on pixels and their neighbourhood as shown in Figure 2.2 (a). The neighbouring pixels are then thresholded by the current centre pixel, as shown in Figure 2.2 (b) and binomial factors (e.g. see Figure 2.2 (c)) are multiplied to the neighbouring positions greater than or equal to the centre pixel as in Figure 2.2 (d). Finally, the sum of the binomial factors yields in the LBP value being assigned to the centre pixel. This procedure is applied on each pixel in the image and the final descriptor is represented by a 256-bin histogram. A possible addition to this approach is combining LBP with a contrast measure, which is described by the difference between the average grey-level of those pixels that have value 1 and those which have value 0 (Figure 2.2 (b)). One of the advantages of LBP is its simple design that still provides a powerful descriptor. But a drawback is its 256 bin histogram representation, which is not a compact descriptor due to its large storage requirements especially for region-based description. Quantisation may be applied to the histogram but this might decrease the description power.

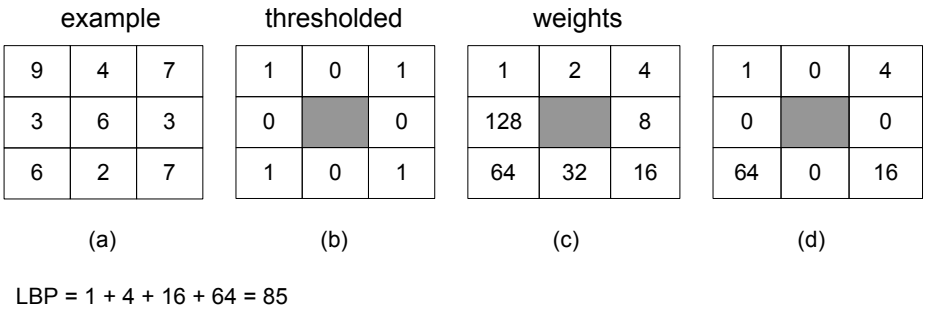


**Figure 2.2 - Computation of LBP value using 3x3 neighbourhood**

There are two major approaches from the signal processing prospective to describe textural features. The first one is a popular and powerful texture descriptor called Gabor filter presenting a multi-resolution approach. The main idea is to process an image by a bank of filters at different scales and orientations (multi-channel) as illustrated in Figure 2.4. Filtering can be applied in either spatial or frequency domain. According to [59], an image I(x,y) filtered with Gabor filter $g_{mn}$ results in its Gabor wavelet transform $W_{mn}$, which captures different frequency and orientation information about texture, can be formulated as,

$$W_{mn}(x,y) = \int I(x_1,y_1)\, g^*_{mn}(x - x_1, y - y_1)\, dx_1 dy_1 \qquad (2)$$

For each scale and orientation the magnitude response $|W_{mn}|$ is calculated as an output from which the first and second order moments (mean and standard deviation) are computed as the texture features. Thus, the feature vector is rather small and is formed per scale and orientation. However, a significant drawback is the computation of the filter coefficients, which is a complex process especially when a higher number of scales and orientations is applied. The second approach is the Wavelet-Transform (DWT), which decomposes a signal into a set of Basis Functions and Wavelet Functions. The wavelet transform computation of a two-dimensional signal (image) is also a multi-resolution (hierarchical) approach, which applies recursive filtering and sub-sampling. At each level (scale), the image is decomposed into four frequency sub-bands, LL, LH, HL, and HH where L denotes low frequency and H denotes high frequency as shown in Figure 2.3, which pictures a 3-scale DWT with ten sub-bands. Here, possible features are energy, mean, variance, and 2$^{nd}$-order statistics of each sub-band where mean and variance are commonly used. Computation and extracting those features may be more efficient compared to Gabor filter. However, describing features based on the three main

orientations (horizontal, vertical, diagonal) Wavelet performance might be degraded for random textures, which cannot be efficiently represented by such primary orientations.

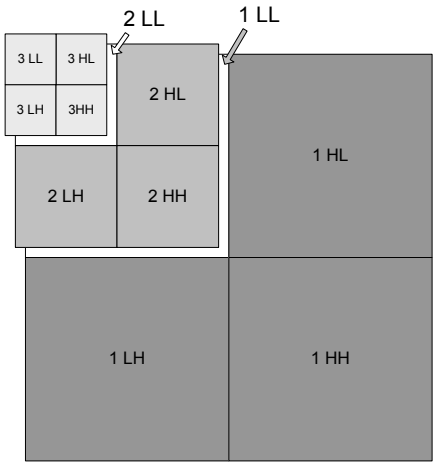


**Figure 2.3 - 3- scaled DWT sub-band decomposition**

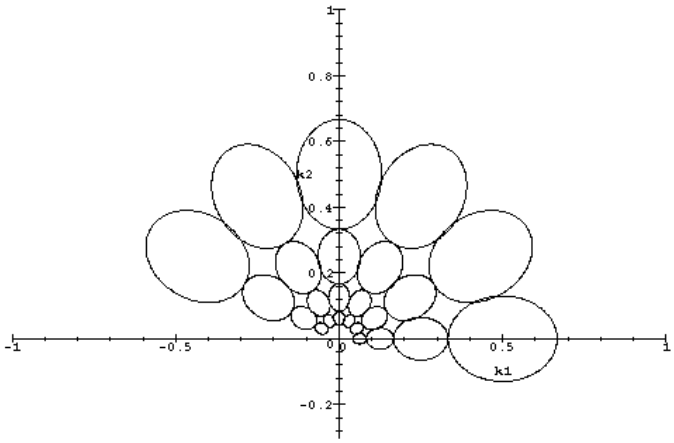


**Figure 2.4 - Contours indicate the half-peak magnitude of the filter responses**

Howarth and Rüger in [36] provided improvement options and performance evaluation for GLCM, Tamura, and Gabor feature extraction where Gabor performed best and the other two showing solid performance.

### 2.1.4 Shape Descriptors in CBIR

Shape information as a property of objects or images is not directly used by humans to describe content due to their learning capabilities, long-term memory, and intelligence where objects are rather distinguished by their semantic meaning. Nevertheless, shape is an important feature in image analysis to describe objects or image regions. Before

applying a shape descriptor, an object extraction or a highly accurate strong segmentation is required in order to extract meaningful regions or objects, which is hard to achieve due to the severe limitations and infeasibilities of such processes over general purpose images. Therefore, shape descriptors are rarely used in CBIR systems. There are few exceptions especially in region-based CBIR systems such as [9], [56], and [88] where shape is used to describe image region properties. Generally speaking, the existing shape descriptors are mainly applied either on binary image databases or image databases where objects are manually extracted. There are two types of shape descriptors: contour-based and region-based. As the names imply, the contour-based methods extract shape properties based on the object outline (contour), and region-based methods utilise the pixel distribution of the 2D object region. Region-based shape descriptors may use one or more region properties such as perimeter, (non-) compactness or (non-) circularity, eccentricity, elongation, rectangularity, and orientation as described in [5] , [42], and [74], or one advanced method such as distance mask [5], medial axis transform [50], and Voronoi diagram [14]. Some of the contour-based shape descriptions are Chain Codes [26], Curvature Scale Space descriptor [65], and Fourier descriptors [92].

Generally, the aforementioned shape descriptors are limited to the object perspective, which means that sophisticated objects usually have different shapes from different perspectives and this makes it even more difficult to obtain a robust and comparable description for similar objects.

### 2.1.5 Spatial Properties

Spatial distribution of objects or regions in a visual scenery plays an important role in HVP. Therefore, this concept has been introduced to CBIR, where regions with similar colour and texture properties can only be distinguished by imposing their spatial constraints. (i.e., blue regions such as “ocean” and “sky” may have similar colour properties, but their spatial locations in images are usually different). Therefore, region locations or spatial relationships between multiple regions in an image are valuable information for the visual description. Here, modelling the spatial relationship is normally expressed by direction information (left_[of], right_[of], above, etc.) [10] and topological description (before, overlaps, inside, etc.) [69].

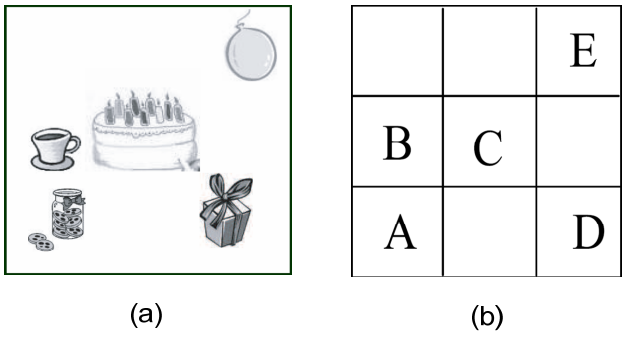


**Figure 2.5 - An original image (left) and its symbolic projection (right)**

A common representation of spatial relationship is the 2D string proposed by Chang et al [10]. It is constructed over the "symbolic projections" of a picture (superimposed grid) along the x and y axes as illustrated in Figure 2.5. In order to obtain the final 2D string construction, objects are labelled by symbols of object icons and different directional relationships between objects by symbols of spatial operators. As an extended variation, the 2D G-string [11] cuts all the objects along their Minimum Boundary Rectangle (MBR) boundaries and extends the spatial relationships into two sets: one defines directional relationships and the other defines the global spatial relationships, indicating that the symbolic projection of objects are disjoined, adjoined or positioned at the same location. In addition, 2D C-string [51] is proposed by Lee and Hsu to minimise the number of cutting objects by leaving the leading object as a whole. This improves the issue of the 2D G-string cutting progress where unnecessary cut objects were generated. 2D-B string [52] presented by Lee, Yang, and Chen does not apply a cutting process as the previous approaches. An object is represented by two symbols, defining the beginning and ending boundary of the object. Matching two images based on their 2-D String representation then comes down to an exhaustive string matching. To speed up the "picture matching" by their symbolic interpretation, a graph representation is proposed by [69] leading to an efficient matching algorithm. In addition to the various 2D string modifications, *spatial quad-tree* [77] , and *symbolic image* [31] are also used for spatial information representation.

However, describing and retrieving images based on spatial relationships of their regions remains a challenge in CBIR since reliable object segmentation is often infeasible except in limited applications. Some systems simply divide the images into sub-blocks with fixed size [84] by which only partial success has been achieved since most natural images are not spatially constrained to such sub-blocks. Other systems such as [56] and [82] integrate spatial properties over segmented image regions. They utilise traditional region properties

Centre of Mass (CoM) and MBR to describe the spatial region relationships among the regions. The drawback of CoM and MBR is that they vary for different region shapes, such as convex and concave as illustrated in Figure 2.6, and they lack a proper description of spatial properties for various region constellations demonstrated in Figure 2.7. The example on the left side shows the problem for a region surround another region without touching it. The CoMs for both regions fall into the same location, which will eventually lead to a wrong spatial property interpretation. The example on the right side of Figure 2.7 demonstrates the issue of MBR where their relationship would be described as Region 1 includes Region 2.

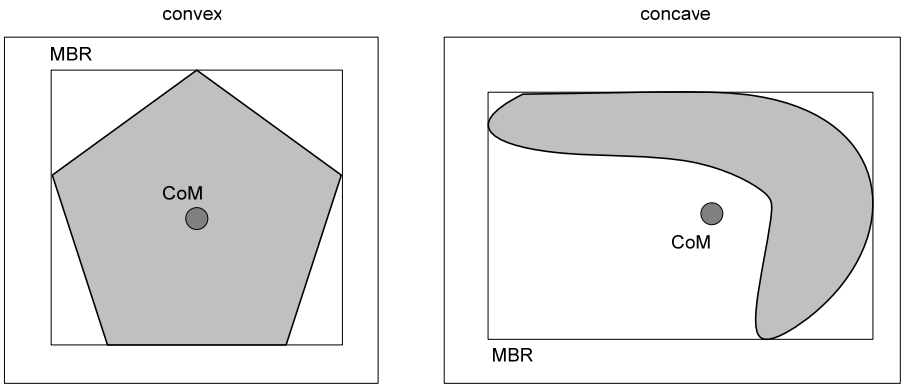


**Figure 2.6 - Region representations by CoM and MBR**

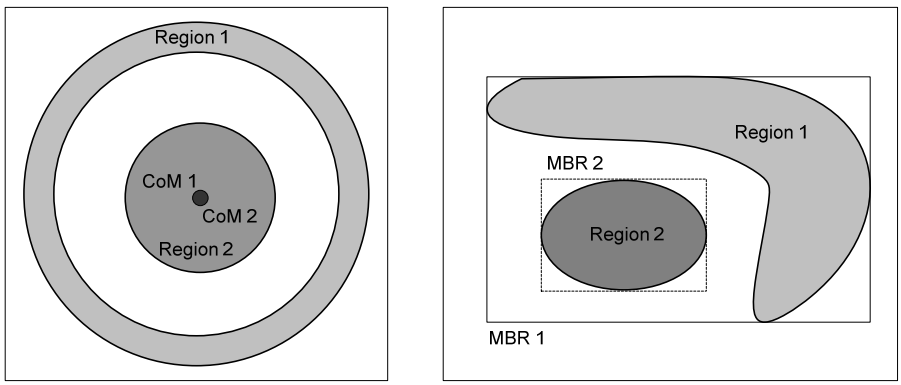


**Figure 2.7 - Drawbacks of CoM and MBR for certain region constellations**

A clear general advantage of exploiting spatial properties of region or feature locations is illustrated by a colour example. For instance, all images (a-f) in Figure 2.8 have the same global colour proportions for Blue, Red, and White. Hence, based on their global colour proportions, these images would be considered similar. By applying spatial properties, globally or locally, would certainly enhanced discrimination among images (a)-(f).

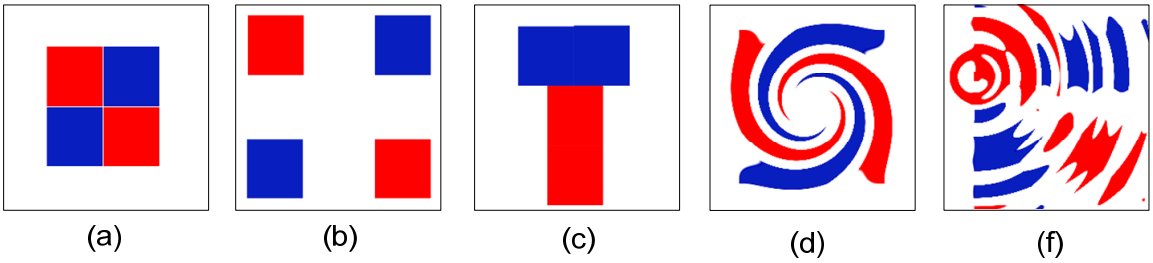


**Figure 2.8 - Images with same colour distributions and different layout**

## 2.2 Retrieval

In any CBIR system, the main objective is the retrieval of the most relevant items in the earliest possible time and retrieval positions (ranks), which traditionally depend on the eminent role of extracting efficient and discriminated features incorporated into an intelligent database management scheme. In order to accomplish query retrieval, similarity distances are compared between feature vectors of the queried object, which can be an example item, a region, a sketch, a region constellation outline, and the database items. Afterwards, the database items are ranked based on (dis-) similarity distances, which are calculated by particular functions provided by their corresponding feature extraction modules. The retrieval process using the extracted features can be divided into three parts, selecting a query object, querying the selected object in a database, and display query results by similarity ranking. An optional fourth part is evaluation of those query results.

The simplest operation of querying is the Normal Query (NQ), which is an exhaustive search based sequential scan. In this case, the query object is compared to every database item using their visual features present. Once the query operation is completed, the retrieval results are ranked based on their similarity scores to the query object and presented after all database items have been processed. Using NQ in large databases may take infeasible time before the user is able to see the final query results. Therefore, a more advanced approach is the Progressive Query (PQ) technique [46]. Its design to offer a solution for querying in large databases by providing the opportunities to split the ongoing query process into intervals either automatically or by user interaction. The query process can be stopped anytime returning results for the already processed database items.

To enhance the speed of a query process, two main indexing categories, Spatial Access Methods (SAMs) [6] and Metric Access Methods (MAMs) [12], can be applied to multimedia databases. Using MAMs and SAMs, Range and *kNN* searches were

introduced to speed up the query process. In Range search, given a query item $q$, a database $D$, and maximum similarity distance (SD) threshold, $\varepsilon$, the problem is to find all database items, $i$, satisfying $\left(i \in D \mid SD\left(q,i\right) \le \varepsilon\right)$. In *kNN* search, the issue is finding the database items, $i$, with the closest similarity distances to the query item, $q$, in a database, $D$, $\left(i \in D, \forall j \in D \mid SD\left(q,i\right) \le SD\left(q,j\right)\right)$. Retrieval provided by those two querying techniques might not be efficient from user's perspective because of the necessity of their parameters, which a user might not be able to provide in an efficient manner.

### 2.2.1 Query Types in CBIR

Any retrieval process starts with selecting or sketching a query object, which the actual querying process is based on. In a CBIR system there are three major query schemes.

The first and probably most popular method is the so-called query by example (QbE) applied by many CBIR systems such as [21], [29], [67], [71], [73], [87], and [88]. In this scheme, the user selects an image item, which is then queried in a database using visual features. An extended form of QbE for images is the query by group where a set of images is selected for retrieval.

Query by sketch (QbS) is another scheme to query in an image database. Here, the general notion is that the user draws a rough coloured outline of the image content in form of region constellations. In this case, a system [82] employing the QbS-scheme ranks images based on their similarity to the user's sketch in terms of colour, shape, and spatial information between drawn regions. Another QbS definition can be seen as sketching the contour outlines of a particular object and then seek images picturing the same or similar objects.

The third type, which is mainly used in Region-based Image Retrieval (RBIR) systems such as [9] and [56], is query by region (QbR). These systems enable the user to select one or multiple regions provided by image segmentation. Then, the system compares the query region(s) with all image regions in the database and ranks the query results (images) based on their region similarities to the queried region.

Figure 2.9 displays a sample query from each of the query schemes. On the left, QbE, the query utilises the entire image as an example. In the centre image, QbR, the query aims for a specific region or object (e.g. *an elephant*). On the right, QbS, the user might look for scenes with a green region in the bottom part and a greyish region in the top.

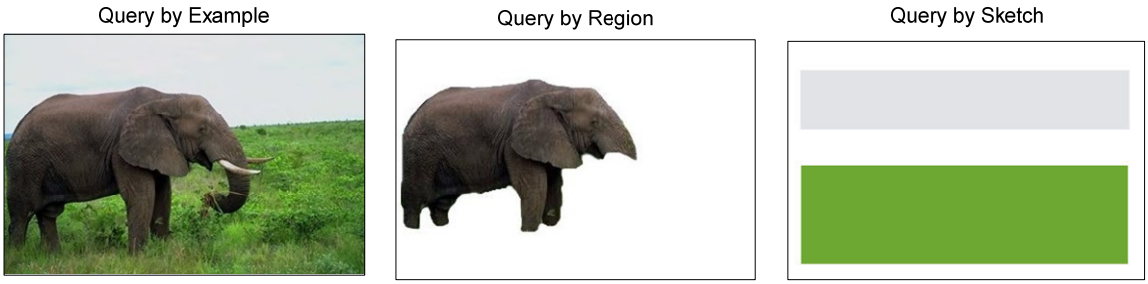


**Figure 2.9 - Examples for different query schemes using images**

### 2.2.2 (Dis-) Similarity Metrics and Models

During indexing and retrieval, (dis-)similarity between two items is expressed via distance calculation. This is mainly achieved by so-called distance metrics. Thus, a metric space is a pair of *(X, d)* where *X* is a set of two entities *x* and *y,* and *d* is distance function. A metric should have the following properties:

1. identity $d(x,y)=0$ if and only if x = y
2. positive definiteness $d(x,y)\geq 0$
3. symmetry $d(x,y)=d(y,x)$
4. triangle inequality $d(x,z)\leq d(x,y)+d(y,z)$

In general, the distance function *d* between any two points in n-dimensional space may be expressed via the equation given by Minkowski also referred to as $L_p$-norm. This generic equation form is defined as,

$$D_p(x,y)=\left(\sum_i^N |x_i-y_i|^p\right)^{\frac{1}{p}}$$
$$D_\infty(x,y)=\max_i(|x_i-y_i|) \qquad (3)$$

where *N* defines the data dimension and *p* determines the degree of distance. The most frequently used Minkowski's distances are the norms of degree one, two and ∞, also called Manhattan distance ($L_1$-norm), Euclidean distance ($L_2$-norm), and Chebyshev distance ($L_\infty$-norm), respectively. The $L_1$ and $L_2$-norm can be directly derived from equation $D_p$ and the expression of the $L_\infty$-norm by $D_\infty$ is given in Eq. 3. Their role in CBIR is mainly calculating the similarity distance between two image items (query and database item) using their feature vectors and to rank the database items thereafter according to the similarity distances to the query item. The $L_p$-norms are commonly used due to their low complexity even for higher dimensional data. However, a drawback of those norms is that

they only take into account the computation between the same data locations (i.e. distance of histograms is calculated bin by bin). Here, possible correlations across dimensions are not taken into account during the distance computation.

In order to address this problem, the quadratic-distance was introduced in QBIC [21] which considers all bin distances within the overall similarity distance. The Quadratic distance $D_Q(X,Y)^2$ is defined as follows:

$$\begin{aligned} D_Q(X,Y)^2 &= (X-Y)^T A(X-Y) \\ &= \sum_i^N \sum_j^N \left(w_i^X - w_i^Y\right)\left(w_j^X - w_j^Y\right) a_{ij} \end{aligned} \tag{4}$$

where histograms $X$ and $Y$ are represented by their colour bins $c_i$ and weights $w_i$ . $A$ with its elements $a_{ij}$ stands for the matrix representing similarities between those histogram colour bins. This allows comparison between different histogram bins having a certain degree of cross similarity. Moreover, the quadratic distance can be applied to *colour sets* [58] as well.

A special case of the quadratic distance is the Mahalanobis distance [19] where the transform matrix $A$ is equal to the inverse of the covariance matrix. In order to apply the Mahalanobis distance, the data are treated as random variables [19]. Therefore, this distance addresses the question whether a particular item would be considered as an outlier relative to another particular item set. It is expressed as,

$$D_M(X,Y) = \sqrt{\left(X-Y\right)^T \Sigma^{-1} \left(X-Y\right)} \tag{5}$$

where $X$ and $Y$ are the data sets and $\Sigma^{-1}$ is the inverse of the covariance matrix.

A widely used metric especially in CBIR is the Earth-mover distance (EMD) [76]. It reflects the minimal amount of work that must be performed to transform one distribution into the other by moving “distribution mass” around. The notion of “work” is based on the user-defined ground distance, which is the distance between two features. For instance let the feature vector of the query image be earth heaps and the feature vector of the database (target) image represents earth holes. The EMD then calculates the minimum total work required to fill the holes of one feature vector with the heaps of the other feature vector given a cost measure. Thus, EMD can be described as,

$$EMD(X,Y) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{6}$$

where $d_{ij}$ is the ground distance between feature representation $X_i$ and $Y_j$ , $f_{ij}$ a set of flows to minimise the overall cost, and $m$ and $n$ denote the number of elements for $X$ and $Y$. The denominator is the normalisation factor. The computation of EMD corresponds to the well-know travelling salesman problem [34] and can be solved by linear programming [16]. An evaluation of most of the aforementioned metrics was conducted in [19].

Yet another metric is the Hausdorff metric or Hausdorff distance. It measures how far two compact non-empty point sets of a metric space are apart from each other. Thus, the distance for two sets $A=\{a_1,...,a_n\}$ and $B=\{b_1,...,b_m\}$ can be described as

$$\begin{aligned} h(A,B) &= \max_{a\in A}\left\{\min_{b\in B}\left(\|a-b\|\right)\right\} \\ H(A,B) &= \max\left\{h(A,B), h(B,A)\right\} \end{aligned} \tag{7}$$

where $h(A,B)$ is called the direct Hausdorff Distance and $\|\cdot\|$ is some underlying norm (i.e. $L_1$, $L_2$ or $L_\infty$) . By representing shape or region information as such point sets, the Hausdorff distance can be used for comparing images [38] and as a distance measure for regions or polygons using region pixels and polygon vertices, respectively. However, this distance is usually not used in CBMR to express similarities based on distances.

Other approaches to calculate the similarity between two feature sets based on statistics are Kullback-Leibler divergence [49] and its numerical stable derivative Jeffery divergence, Kolmogorov-Smirnov distance [28], Chi-Square statistic [19], and G-Statistic [70].

### 2.2.3 Performance Measures

To measure the retrieval performance of a conducted query, the query results need to be evaluated. Generally, multimedia databases are divided into predefined categories (classes), which are generated by subjective classification and usually represent similar semantic content. These categories are also used for evaluating retrieval performance where all category items are the so-called ground-truth data, which are considered relevant for a particular category. After obtaining the results for a query, there are exist different methods evaluating them. Automatic evaluation can be done without human interaction

where items are considered relevant for the retrieval if they belong to the same predefined category as the query and irrelevant otherwise. In manual evaluation items are judged relevant if they present similar (semantic) content to a human observer as the query. Here, personal assessment plays an important role because what might be relevant to one person may be irrelevant to another.

Performance measures provide numeric evaluation of retrieval results. Normally, the automatic evaluation is preferred due to its fast calculation. However, its performance will depend on the subjective category classification in the database. On the one hand, if a database contents coarse category classifications then it might become difficult to retrieve those semantic category elements by low-level features what will eventually result in poor performance. The reason for this is that such categories will contain similar semantic objects represented by different low-level features. On the other hand, if the category content is well-separated by its low-level features then a meaningful evaluation for descriptors may be provided.

However, note that different visual and aural features may have different ground truth data in the same database. Moreover, there might be also a difference due to semantics where items may be considered the same category but have nothing in common feature-wise. Therefore, an appropriate category classification is a necessity for evaluating different visual features.

During retrieval, conducting only one query per class or feature does not give enough statistics for the evaluation of the retrieval performance by numeric measurements. Hence, multiple queries are carried out to obtain a healthy statistics about the performance using some numeric methodologies, as detailed next.

I. Retrieval Rate (RR)

Probably the simplest and most straightforward approach is the ratio between relevant retrieved items for a query $q$ against its ground truth data. This measure is the so-called retrieval rate (RR) [58] and is defined as,

$$RR(q) = \frac{NF^{\alpha}(q)}{NG(q)} \leq 1$$

$$ARR(q) = \sum_{q=1}^{NQ} \frac{RR(q)}{NQ} \leq 1 \tag{8}$$

where $NF^{\alpha}(q)$ are the relevant items found within the $\alpha \cdot NG(q)$ retrieval results and $NG(q)$ is the number of ground truth items in the database for a query $q$. $RR(q)$ returns a value in the range of [0,1] where 0 indicates that no relevant items were retrieved and 1 denotes a perfect retrieval. If multiple queries are performed then average retrieval rate is obtain by $ARR(q)$ where $NQ$ is the number of queries with $q \in [1..NQ]$. The $\alpha$ in RR can be seen as a tolerance weight with possible values $\alpha \geq 1$. On the one hand, setting $\alpha = 1$ is the most general measurement; however this might be too small for certain classes or features, which would not allow any tolerance since ground truth items in rank $NG(q)+1$ would be excluded from contribution. On the other hand, $\alpha > 1$ leaves more tolerance to the retrieval but will also be less discriminative between excellent and poor results. This is due to the fact that retrieval positions (ranks) are not considered at all, which means that the rank of the relevant items retrieved do not affect over the performance measure.

II. Precision and Recall (PR)

Another traditional measure, Precision and Recall, uses a similar approach as RR. Recall (R) is the ratio of retrieved relevant items to the number of relevant items in the database, and Precision (P) is the ratio of retrieved relevant items to the total number of retrieved items during querying. It can be seen that recall is similar to RR with $\alpha > 0$ and precision is an additional measure for evaluating the retrieval quality between relevant and irrelevant retrieved items. Hence, both are formulated as follows

$$P = \frac{NF_q}{NF_q \cup NR_q} \leq 1 \qquad R = \frac{NF_q}{NF_q \cup NQ_q} \leq 1 \tag{9}$$

where $NQ_q$ is the set of relevant not retrieved items, $NR_q$ is the set of irrelevant retrieved items, and $NF_q$ represents the set of relevant retrieved items. The left side of Figure 2.10 shows a graphical illustration of these sets in Eq. 9. Both values, P and R, are given in terms of percentages.

The trade-off between Precision and Recall appear, in practice, to be inversely related where improvement in either tends to be connected with poorer performance of the other [8]. Hence, if recall increases more relevant items are found but, generally, this will decrease precision because to find these additional relevant items more irrelevant items are supposedly retrieved. However, a major drawback of this method is that PR is also

invariant to the ranking information of the relevant items, similar to RR. A sample PR graph is plotted in Figure 2.10 (b) where the dashed line represents a perfect retrieval case.

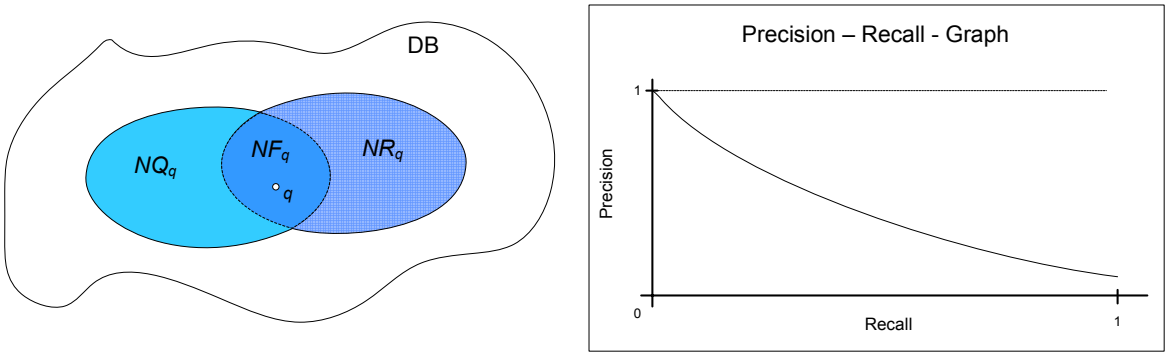


**Figure 2.10 - Graphical illustration of retrieval sets (left), a sample PR - graph (right)**

III. Average Normalised Modified Retrieval Rank (ANMRR)

Ranking information of the relevant items retrieved in a query operation is important for evaluation to avoid misleading performances such as in RR and PR. In order to address this, a normalised measure called Average Normalised Modified Retrieval Rank (ANMRR) is defined and used as the retrieval performance evaluation criterion in MPEG-7 [41]. The average rank for query results of a query $q$ is defined as

$$AVR(q) = \sum_{k=1}^{NG(q)} \frac{Rank^*(k)}{NG(q)}$$
$$Rank^*(k) = \begin{cases} Rank(k), & if\ Rank(k) \le W \\ W+1, & if\ Rank(k) > W \end{cases} \quad \text{and } W = 2\,NG(q) \tag{10}$$

where $NG(q)$ defines the number of ground truth items in the database for the query $q$, $Rank(k)$ provides the ranking of the ground truth items during retrieval within a window of $W$ retrieved items, which are taken into account querying $q$. Here, $W$+1 is used as a penalty assigned to the ground truth items with ranks outside the considered window size $W$. The formula for the unbiased normalised modified retrieval rank $NMRR(q)$ is expressed as,

$$NMRR(q) = \frac{AVR(q) - NG(q) - 1}{2W - NG(q) + 1}$$
$$ANMRR = \sum_{q=1}^{NQ} \frac{NMRR(q)}{NQ} \le 1 \tag{11}$$

As a (unit) normalised value *NMMR(q)* operates in the range of [0,1]. The final step is averaging *NMMR(q*) for all queries where *NQ* is the number of queries with q∈[1..*NQ*]. This leads to the final formula for *ANMRR*, which indicates the retrieval performance achieved. It is in the range of [0, 1] and the smaller the value the better the retrieval result where 0 indicates a perfect retrieval and 1 denotes that no relevant items were retrieved.

## 2.3 The MPEG-7 Standard

MPEG-7 [41] is an ISO/IEC standard from Motion Picture Experts Group (MPEG). It started during the late 90's and was formally known as Multimedia Content Description Interface. Its main purpose is to describe multimedia content data independent from the application. The following two sections will give a brief introduction into the MPEG-7 standard with its objectives

### 2.3.1 General Overview

During the past, aural and visual information were mainly consumed by humans. Nowadays besides human beings, there are more and more systems, which create, exchange, retrieve, and re-use audio-visual information and content. These systems are used, for instance, in image understanding, media conversion, information retrieval, and filtering the content information in streamed data. This creates a certain demand to integrate information about the content within the audio-visual data. The integration development of such content information was mainly driven by new technologies, trends in markets as well as in user needs. These factors increased the demand for tools and systems for indexing, searching, filtering, and managing audio-visual content in stored and streamed (live) media. A certain form of representation is needed to allow interpretation of the meaning of the information. The MPEG-7 standard emerged to form that representation. Therefore, it provides a flexible and generic interface defining syntax and semantics of different description tools and brings systems and applications closer, all of which can then be used in generating, managing, distributing, and consuming descriptions for audio-visual content. The scope of MPEG-7 is illustrated in Figure 2.11. Various tasks (such as identification, retrieval, or filtering of audio-visual data or information) and systems (such as mobile phones, set top boxes, PC, etc.) are supported. One of the intentions of MPEG-7 is to make the web searchable for multimedia content as it is for

text today. Other possible applications (broadcast media selection, digital libraries, multimedia directory services, multimedia editing) and tasks can be enlisted as follows:

- multimedia - generate a customised program guide or summary of broadcast audio-visual content according to the user's preference and history
- graphics - draw a few lines on a screen and get, in return, a set of images containing similar graphics, logos, or ideograms
- scenario - on a given audio-visual content, describe actions and get in return a list of scenarios where similar actions take place.

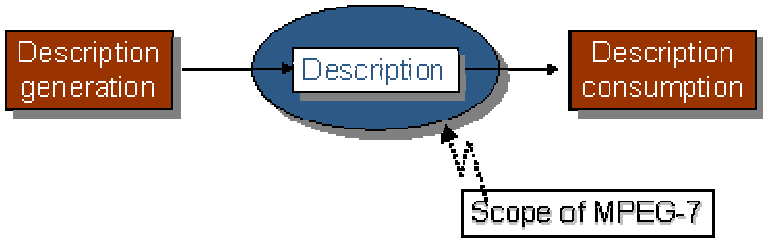


**Figure 2.11 - Scope of MPEG-7**

### 2.3.2 MPEG-7 Structure and Components

The following entities are provided in this above mentioned generic framework. MPEG-7 provides *descriptors* **(**Ds**)** to define syntax and semantics of audio-visual content features - such as low-level (colour, texture, shape) and high-level (semantics) such as events, abstract concepts, content genres- and *descriptor schemes* **(**DSs**)** to generate complex descriptions by specifying structure and semantics of relationships between components (descriptors or descriptor schemes). Figure 2.12 illustrates the structure of MPEG-7 with the integration of Ds and DSs. These descriptions can be applied to different perceptual and semantic levels. Furthermore, several media types are supported such as still images, graphics, 3D models, audio, speech, video, and their composition. However, MPEG-7 does not define how to extract and/or process these descriptions. All this is designed in such a way to leave maximum flexibility to applications but it is meant for standardising or evaluating these applications.

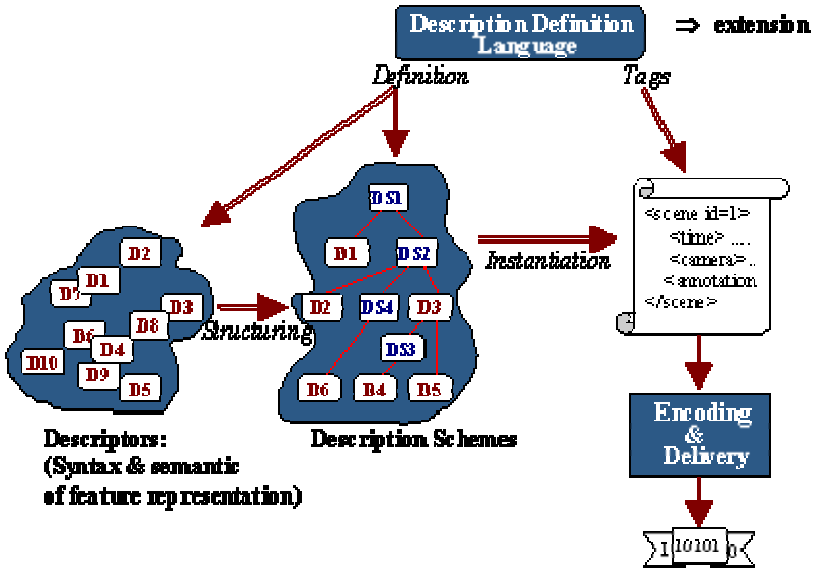


**Figure 2.12 - Overview of MPEG-7 structure**

The MPEG-7 standard is divided into seven main parts, which are listed and briefly described below.

1. **Systems**
   - for system level functions
   - efficient transport and storage of descriptions, sync content and description

2. **Description Definition Language (DDL)**
   - generate new D/DS, extend and modify existing D/DS
   - extension of XML (eXtended Mark-up Language)

3. **Visual**
   - set of standardised visual D/DS
   - colour, texture, shape, motion

4. **Audio**
   - set of standardised audio D/DS
   - 4 classes: pure music, pure speech, pure sound effects, arbitrary soundtracks
   - D/DS use features as: silence, spoken content, timbre, sound effects, melody

5. **Multimedia Description Schemes**
    - provides possibility for generic description of all types of multimedia including audio, visual, and textual data
    - content management and content description
    - navigation and access
    - user interaction
    - organisation
6. **Reference Software**
    - reference implementation of significant parts known as experimentation software (XM) [90]
7. **Conformance**
    - present guidelines and procedures for testing the conformance of MPEG-7 implementations

# 3 CBIR Systems

As mentioned earlier, there are systems such as MUVIS [67], which provides a framework for indexing and retrieval that is easily extendable by any third party using their on description modules, which is also the major reason for using the MUVIS system to accomplish the proposed framework in a dynamic and efficient way. In the following subsection we will first introduce MUVIS and then provide an overview of other similar CBMR systems and frameworks.

## 3.1 MUVIS Framework - A Sample System

The first MUVIS [13] was developed in the late 90's as a JAVA application and was intended as a platform for testing CBIR algorithms for the Image Analysis Group in the Signal Processing Institute at the TUT. It was capable of indexing images on low-level features such as colour, texture, and shape. With the beginning of 2000, a new framework [45] was introduced. It provided a generic solution as a unified global approach for indexing and retrieval supporting various multimedia data types such as images, video, and audio on windows-based computer-platforms. In order to accomplish this, its main goal was to support a dynamic integration process for different feature extraction methods especially for the third parties. This new system introduced a novel query technique namely, the Progressive Query (PQ), as well as an efficient indexing scheme, the so-called Hierarchical Cellular Tree (HCT) [47]. The current version MUVIS (Xt) v1.8 extends the framework structure by spatial segmentation and shot boundary detection modules. Besides the computer version of MUVIS there also exists version called M-MUVIS [2] for mobile devices. The following sections provide an overview of the MUVIS framework, describing its main components and properties.

### 3.1.1 General Overview

The main objective of the MUVIS system is to provide the necessary functionality for indexing and retrieval of multimedia databases. Figure 3.1 shows an overview of the MUVIS framework. MUVIS provides two main applications *DbsEditor* for indexing and *MBrowser* for retrieval where both interact with several external modules, multimedia

items, and databases. These modules can be developed as Dynamic Linked Libraries (DLLs) and will be linked automatically to the applications at run-time, if required.

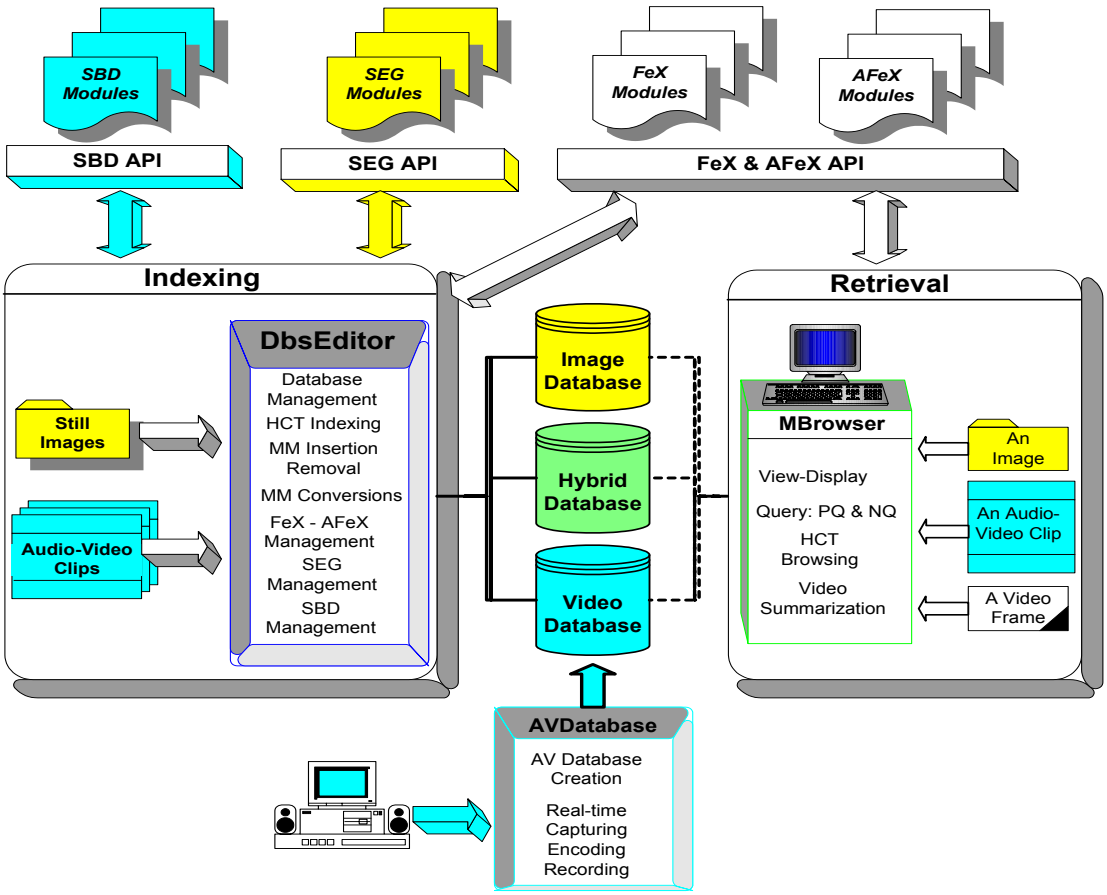


**Figure 3.1 - Overview of the MUVIS framework with its individual components**

Three database types are supported within MUVIS as follows:

- *Image*: holding images and their extracted feature information
- *Video*: holding video clip, extracted key frames, and feature information
- *Hybrid*: supports combination of images and videos in a single database

By supporting various database types, it is also necessary to provide the support for various multimedia formats. Therefore, MUVIS facilitates the most common image, audio, and video formats where Table 3.1 and Table 3.2 provide an overview of supported image types and audio-video formats, respectively.

**Table 3.1 - MUVIS image types**

| MUVIS Image Types | | | | | | | |
|---|---|---|---|---|---|---|---|
| Convertible Formats | | | | | | | |
| JPEG | | JPEG 2K | BMP | TIFF | PNG | | |
| Non-convertible Formats | | | | | | | |
| PCX | GIF | PCT | TGA | PNG | EPS | WMF | PGM |

**Table 3.2 - MUVIS audio and video formats**

| MUVIS Audio | | | | | MUVIS Video | | | |
|---|---|---|---|---|---|---|---|---|
| Codecs | Sampling Freq. | Channel Number | File Formats | | Codecs | Frame Rate | Frame Size | File Formats |
| MP3 | 16, 22.050, | Mono | MP3 | | H263+ | 1..25 fps | Any | AVI |
| AAC | 24, 32, 44.1 KHz | Stereo | AAC | | MPEG-4 | | | MP4 |
| G721 | Any | | AVI | | YUV 4:2:0 | | | 3gp |
| G723 | | | MP4 | | | | | |
| PCM | | | 3gp | | RGB 24 | | | |
| AMR | 8 – 16 KHz | Mono | AMR | | H.264 | | | |

I. Indexing - *DbsEditor*

The database management and organization capabilities in MUVIS are provided by the application *DbsEditor*. The left part of Figure 3.1illustrates its role in the overall system. It uses external feature extraction modules and organizes multimedia databases. Moreover, as its GUI shown in Figure 3.2, users are able to carry out the following actions:

- appending/removing images and video clips
- appending/removing visual and aural features
- appending/removing segmentation algorithms for frames and images
- appending/removing shot boundary detection and key-frame extraction methods for video clips
- similarity-based indexing a database by HCT

The necessary modules for the feature extraction, segmentation, and shot boundary detection are automatically detected and linked to the application at start up. The existing modules appear in the application GUI from where the user can select particular methods (embedded in their modules), set their parameters, and finally features, segmentation masks, and video shots that are selected initially, are extracted from appropriated media items in the current database. *DbsEditor* further performs similarity indexing on a multimedia database using HCT.

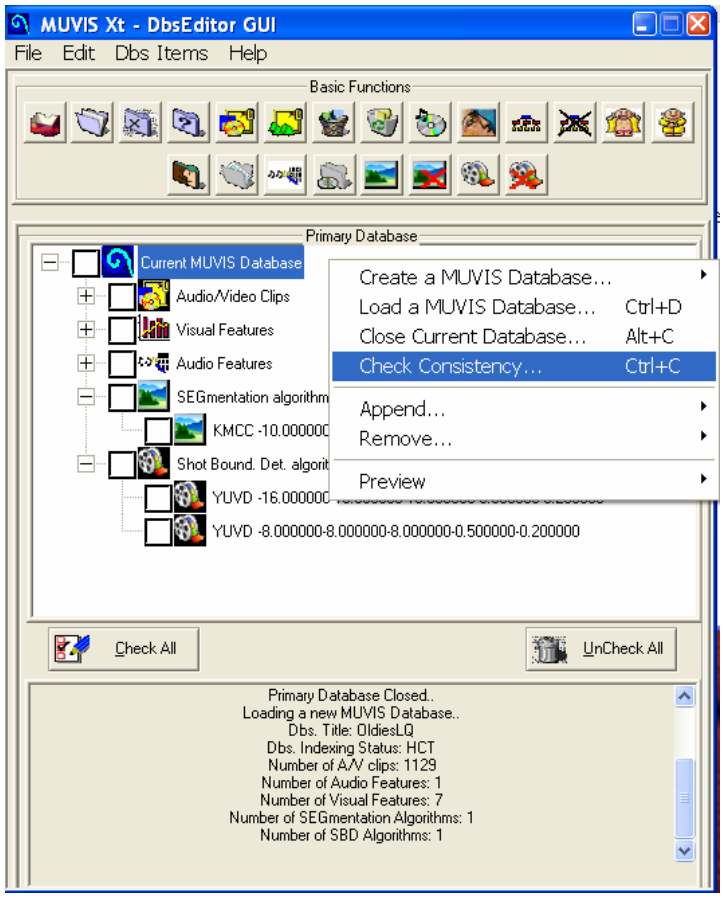


**Figure 3.2 - GUI of DbsEditor**

## II. Retrieval - *MBrowser*

*MBrowser* can basically be used to browse in a MUVIS multimedia database created by *DbsEditor*. Whilst browsing *MBrowser* the user is enabled to perform a content-based query of an image or video (i.e. QbE) in a database by using the visual or aural descriptors. A snapshot of an image query is shown in Figure 3.3. For the selection of the query image, the user has two options: choosing an image from the active database or loading an external image. The user can enable or disable particular features and their weights a prior to a query operation. Otherwise default scheme is that all features are equally weighted. After the selection of the query item and features, the actual query process can be conducted. For this, MUVIS supports three querying methods: NQ, Progressive Query (PQ), and Interactive Query (IQ). Here, IQ denotes the implementation of the PQ scheme over HCT, which provides an option to obtain best matching results first during an ongoing PQ process. For more detail the reader is referred to references of PQ and HCT.

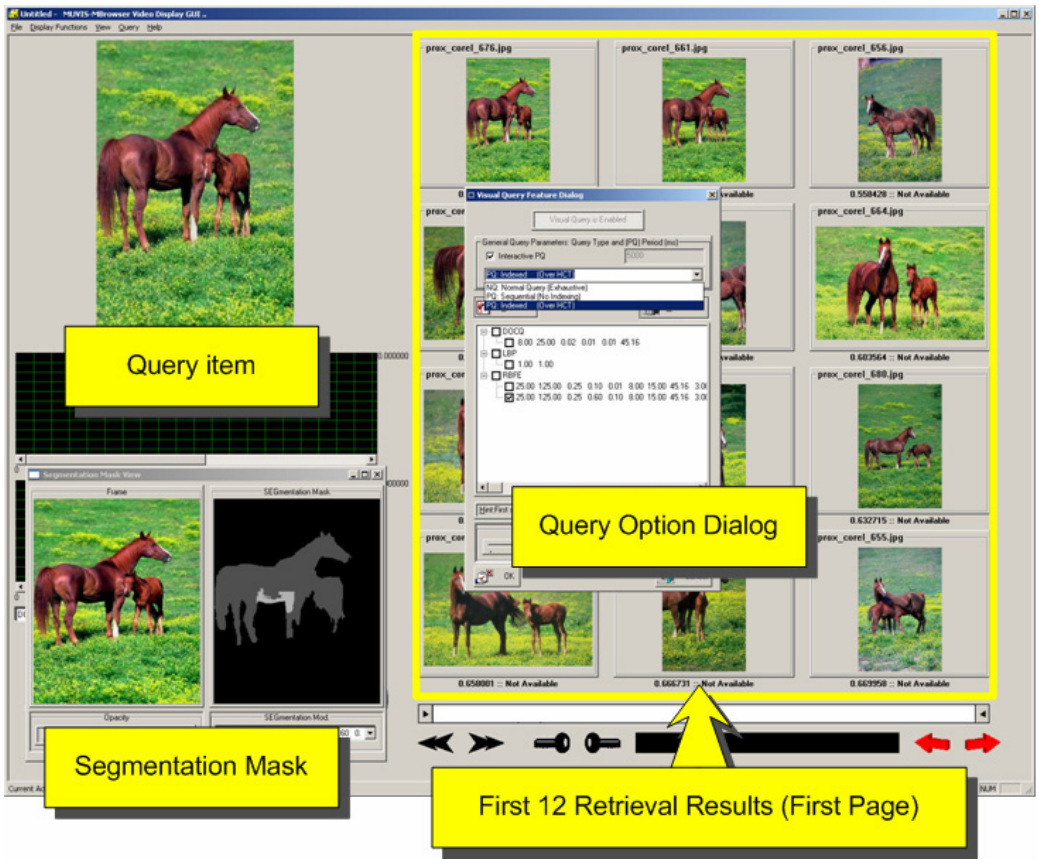


**Figure 3.3 - GUI of MBrowser**

### 3.1.2 MUVIS Xt: The eXtended framework

The two main applications described above only provide the indexing and retrieval functionality within MUVIS. Yet another essential part of the MUVIS framework is the eXtended framework introducing different module frameworks and APIs [32]. This allows the development of external modules as DLL to implement and test feature extraction methods, segmentation and shot boundary detection algorithms. Generally, there are four types of modules provided by the extended Framework:

1. Aural Feature eXtraction (AFeX).
2. Shot Boundary Detection in video (scene changes) (SBD)
3. Visual Feature eXtraction (FeX)
4. Spatial SEGmentation (SEG)

AFeX allows audio feature extraction over multimedia items (audio and video clips). With SBD modules shot boundary detection and key-frame extraction algorithms can be

implemented and tested for video clips. Since this thesis mainly copes with still images and thus only uses the FeX and SEG structure, we will detail them as follows.

**FeX and SEG frameworks**

FeX and SEG modules can be applied to multimedia databases. FeX modules incorporate descriptors to extract visual features such as colour, texture, and shape; whereas spatial image segmentation methods are implemented into SEG modules. Both frameworks provide five API (interface) functions which should be implemented by the module creator. The FeX-framework structure and its API functions are shown in Figure 3.4.

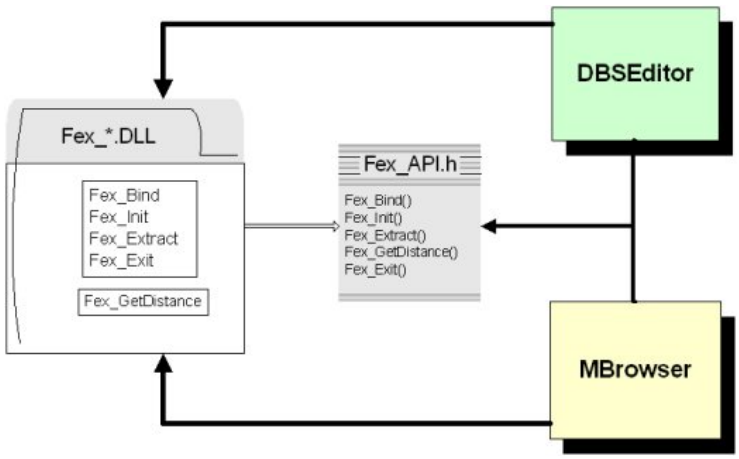


**Figure 3.4 - Overview of FeX Framework structure**

The five functions, namely `Fex_Bind`, `Fex_Init`, `Fex_Extract`, `Fex_Exit`, and `Fex_GetDistance`, are the API functions between the module and MUVIS applications, *DbsEditor* and *MBrowser*. `Fex_Bind` registers the module to the application, `Fex_Init` initialise the module with the user-parameters. `Fex_Extract` provides the main functionality for the corresponding module type and it is called for every database item. `Fex_Exit` terminates the module operation and frees allocated memory. `Fex_GetDistance` returns a similarity measure calculating the distance between two database items based on their feature vectors.

In the FeX-framework, `Fex_Extract` employs the actual feature extraction algorithm and returns the feature vector as an output for a database item (e.g. the current image or a key-frame of the current video under FeX operation). The current FeX realisation requires that all extracted feature vectors have the same length.

In the SEG-framework, `Fex_Extract` employs the actual image segmentation algorithm and returns as output the corresponding segmentation mask. So far, the segmentation masks are only used for test and evaluation purposes. *DbsEditor* extracts and stores the segmentation masks whereas *MBrowser* provides the necessary GUI support to render the segmentation masks of images or video key-frames in the database as shown in Figure 3.3. Segmentation masks are stored as an 8 bit grey-level value image in PGM format which practically limits the number of regions to 256, which is a convenient upper limit. Hereby, the module is responsible to assign unique grey-level values to the image pixels and segments.
The following list shows some of the FeX and SEG modules, which have been integrated into MUVIS so far:

| | |
|---|---|
| Colour: | RGB, HSV, YUV histograms, Dominant Colour [58] |
| Texture: | Gabor-filter, GLCM, LBP, Wavelet-transform |
| Shape: | MPEG-7 Edge Histogram [58], 2D-Walking Ant Histogram [24] |
| Segmentation: | KMCC [61], JSEG [18], graph-based segmentation [23], quad-tree split and merge, watershed |

## 3.2 Other CBIR System Approaches

There are two types of CBIR systems: the so-called content-based retrieval and its branch region-based retrieval systems. The following two subsections will give a brief overview of these types and introduce a few selected for each category.

### 3.2.1 Content-based Image Retrieval Systems

The majority of the existing systems such as Virage [87], MARS [71], Photobook [73], and PicToSeek [29], exploit single or multiple frame-based image properties such as colour, texture, and shape, using the QbE-scheme for retrieval. IBM's QBIC [21] was the first commercial retrieval system for image and video providing a huge variety of options to the user such as simple segmentation for certain images categories, several features, and query possibilities. They use low-level features from colour, texture, shape, and motion. Colour feature can be either the mean colour for an object (region) or a 256-bin colour

histogram for an image in a selected colour space such as RGB, YIQ, CIE-Lab, and Munsell. Texture is described by the modified versions of the coarseness, contrast, and directionality features. Shape features are described by area, circularity, eccentricity, major axis orientation, and a set of algebraic moment invariants. Based on these features, QBIC allows various query types on objects and regions (e.g. “Find images with a red, round object”), scenes and images (e.g. “Find images that have approximately 30-percent red and 15-percent blue colours”), video shots (e.g. “Find all shots panning from left to right”), and combinations of these (e.g. “Find images that have 30 percent red and contain a blue textured object”). To compare the different features during retrieval, a weighted Euclidean distance is used for colour average, texture, and shape features. QBIC was the first system that introduced the quadratic histogram distance for colour histograms in order to catch inter-colour similarity between histogram bins. Besides all options for feature extraction and retrieval, QBIC was also the one of the first systems that applied multidimensional indexing to enhance the speed performance of the system.

Another system following a slightly different approach is VisualSeek [82], which integrates QbS as its main retrieval scheme. The query process consists of finding images that contain the most similar arrangements of user-sketched region outlines. For this, database images are automatically decomposed into segments of equally dominant colours where regions are then extracted by a back-projection technique. From these regions various features such as colour set, region centroid, area, and the width and height of the MBR are extracted. In order to start a query, the user sketches some regions on a grid providing the position and dimension information and then selects a colour for each region. Moreover, the user can indicate boundaries for location, size and/or spatial relationships between regions. If the user defines spatial boundaries for the query region, then its distance to a target region becomes zero if the target region centroid falls inside these boundaries, and is otherwise given by the Euclidean distance between the centroids. The distance between two region areas is the absolute difference, whilst the Euclidean distance is used for computing the distance between the MBRs of two regions. As the colour set similarity, a modified quadratic distance is applied. The system returns thumbnail images of the best matches as the retrieval results and allows the user to perform further QbE operations using these images as (query) examples.

### 3.2.2 Region-based Image Retrieval Systems

With the ongoing research, aforementioned limitations describing the image content using frame-based features occurred. Hence, to over come the frame-based limitations and the desire to model the human visual system lead to the development of region-based retrieval systems. In general, the idea is to segment an image into homogeneous regions where segmentation ought to be consistent in a manner that similar images (scenes, regions) are partitioned in a similar way. Then, feature extraction and retrieval is performed based on those regions. Basically two approaches evolved: direct region-based image retrieval (RBIR) and regionalised CBIR (rCBIR) where both use region-based feature description. However, they differ in their query scheme. The rest of this sub section present an overview of both approaches and introduces a few systems for each approach.

RBIR usually considers the retrieval of one or sometimes multiple user-selected image region(s) that are obtained by segmentation. The major objective of such QbR systems is the fact to search for specific objects or regions rather than the “whole picture”. Therefore, selecting a single region to find images with similar regions tries to model this human search behaviour. One of the first systems introducing this approach was Blobworld [9]. It uses an Expectation-Maximization algorithm for segmentation and they call a region as “blob”. They apply a CIE-Lab histogram (5x10x10 bins for L, a, and b components) as the colour descriptor and texture is described by contrast, anisotropy, and polarity. Furthermore, region location and shape information are described using approximate area, eccentricity, and orientation of the regions. Various retrieval options are provided to the user such as the selection of a category for the image, which limits the search space. Furthermore, blob and feature relevance (importance) can be indicated using reserved words such as ‘not’, ‘somewhat’, or ‘very’. Also the system allows the selection of more than one region for querying. During the retrieval, Mahalanobis distance is computed between regions to measure their feature similarity. Moreover, the system allows a one-to-many region matching. Thus, a selected region from the query image can either match one or more regions in the target image.

Another region-based system, IKONA [22] introduces a fast coarse segmentation with a fine colour description. Segmentation is achieved for uniform and textured regions by colour quantisation clustering using Local Distributions of Quantized Colours (LDQC) over a sliding window. As a region representation they introduce Adaptive Distribution of Colour Shades (ADCS) using a CIE-Luv colour histogram (including region colours and

their shade variations), which describes colour and texture; thus, a separate texture descriptor is not applied. They further use region area, position (region centroid), and compactness (ratio of the sum of region contour lengths and region area) as region properties. As a retrieval measure, a combination of the single region features is extracted and expressed as follows:

$$d_{final}\left(R_Q, R_C\right) = \alpha_{ADCS} \cdot d_{quad}^{ADCS}\left(R_Q, R_C\right) + \alpha_A \cdot d_{L_1}^{A}\left(R_Q, R_C\right) + \alpha_P \cdot d_{L_2}^{P}\left(R_Q, R_C\right) + \alpha_C \cdot d_{L_1}^{C}\left(R_Q, R_C\right) \quad (12)$$

where $d_{quad}^{ADCS}$ is the quadratic distance between the colour shades, $d_{L_1}^{A}$ is the $L_1$-norm between region areas, $d_{L_2}^{P}$ is the Euclidean distance between region positions, and $d_{L_1}^{C}$ is the $L_1$-norm between region compactness. The $\alpha$ - values are relative importance weights set by the user.

A system introducing some new approaches is NeTra [56]. It applies a contour-based segmentation and also uses the aforementioned region features utilising different descriptors. Colour information is described by salient colours using a 256 RGB-colour codebook, whereas the texture descriptor uses mean and standard deviation of Gabor-Wavelet-Transform, and shape is described by a combination of a curvature function of the contour, a centroid distance function, and a complex coordinate function. Besides all these region-based features, the system further integrates *spatial locations* between regions. In order to accomplish this, a region is described by its CoM and MBR. To initiate a query, the user selects a region and also chooses a specific feature for retrieval. If *spatial location* feature is selected, colour and spatial location of the region(s) have to be provided. For querying by *spatial location,* two bounding boxes have to be defined by the user to mark the area of interest. The inner box is used to define the regions integrated into the query process if image regions are overlapping with this box, and the outer box restricts such regions that have to be in the query. Thus, if the regions exceed this outer box, they will not be included into query process. In order to measure the region similarity, colour matching between regions is defined as follows: match each colour in region A with its closest colour match in region B where the colour distance is calculated by a weighted Euclidean Distance in the RGB colour space. Texture and shape matching is performed using the $L_1$-norm and the Euclidean distance, respectively.

The main difference of RBIR systems to rCBIR systems is that all image regions are included in the retrieval process of an rCBIR system rather than one or a few. Hence, it

can be seen as a special case of RBIR using multiple regions for querying. The QbE-scheme is used where each image region might contribute to the overall image similarity. This has the advantage that the burden is taken away from the user to select a particular region for retrieval, which may be sometimes rather difficult if objects are represented by multiple regions or occlusions occur among the objects.

There are several systems applying this approach such as Walrus [68] and Windsurf [4] but one of the most commonly known systems of this kind is SIMPLicity [88] with its integrated region matching (IRM) approach. It applies block-based k-means clustering for segmentation where features are extracted by Wavelet transform over 4-by-4 image blocks. Extracted regions are then described by colour, texture, and shape. The colour descriptor is the average region colour in CIE-Luv space, texture is described by the mean of the square root of 2$^{nd}$ order moment of the sub-band wavelet coefficients, and shape description uses normalised inertia of the region. During querying, the image similarity is measured by integrating all image regions. Therefore, to cope with inaccurate segmentation, a many-to-many region matching is incorporated where one region of an image may match several regions of another image as shown in Figure 3.5. A region-to-region match is obtained when their extracted region features are significantly similar. They apply such a matching principle that requires the most similar region pairs to be matched first. The matching of two regions is defined as:

$$d\left(R_1, R_2\right) = \sum_{i,j} s_{ij} \cdot w_{ij} \cdot d_{ij} \tag{13}$$

where $s_{ij}$ is the significance credit, $d_{ij}$ is the region similarity, and $w_{ij}$ is an adjustment parameter for the two regions on the similarity measure. Region similarity, $d_{ij}$, is calculated via a weighted $L_1$-norm. The significance credit indicates the importance of the matching between the two regions and it is calculated using the region area weights. Furthermore, whilst measuring the region similarity, a region's image location is also considered where central regions are slightly favoured over the regions nearby the image boundaries. For more detailed information about IRM see [88].

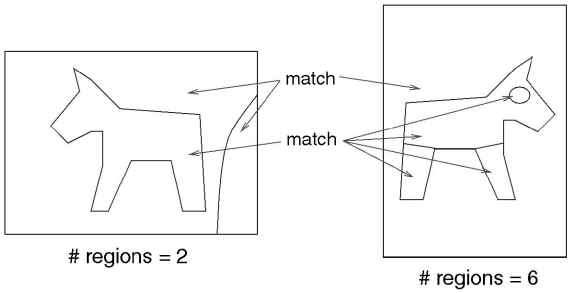


Figure 3.5 - IRM matching scheme (figure taken from [88])

# 4 A Regionalised Content-Based Image Retrieval Framework

In the previous chapter, various CBIR systems and approaches were presented. From those rCBIR systems it can be seen that their general approach is similar, i.e. first segmenting the image and then extracting low-level features over the obtained regions. However, none of these systems proposed a framework structure for generic, flexible, or exchangeable integration of segmentation methods and feature extraction approaches. As mentioned earlier in this thesis our aim is not to develop a specific, stand-alone system; instead the goal is to propose a global approach for regionalised CBIR utilising region-based features and spatial properties. In this chapter, a framework, which is designed to provide a model for the regionalised content-based image indexing and retrieval, is introduced.

## 4.1 Generic Overview

The general approach of this framework is similar to previous aforementioned regionalised approaches, which are basically based on image segmentation and region-based feature extraction. However, the main difference is that this framework allows the applicability of any segmentation algorithm and feature extraction method rather than specifically tuned algorithms. Along with these two parts, the framework is extended and introduces two independent stages, grouping and spatial information. Figure 4.1 illustrates an overview of this framework, which can be divided into two processing phases: indexing and retrieval. During the indexing, there are four main steps such as *segmentation*, *grouping*, *local* and *spatial feature* extraction, and recall from the earlier discussion that two among them, *segmentation* and *local feature* are considered as black-

boxes.

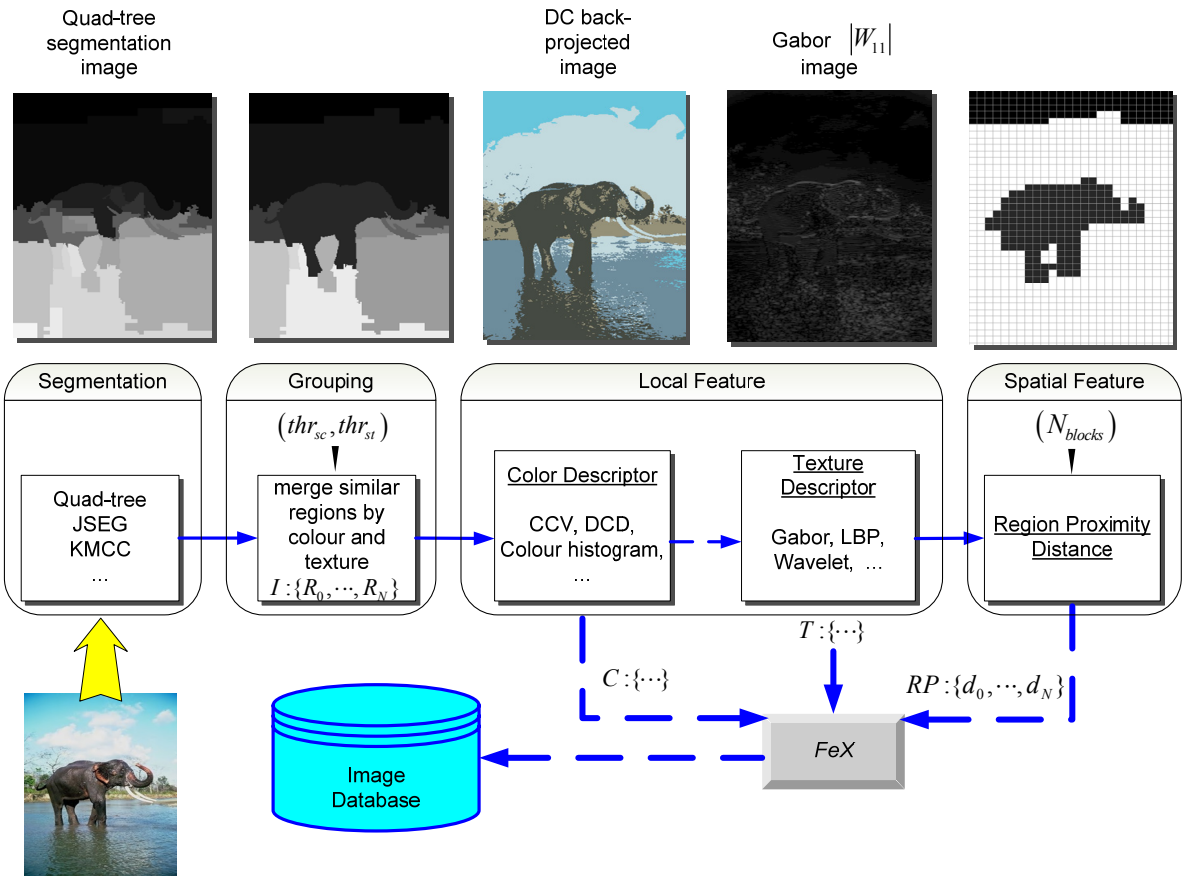


**Figure 4.1 - Overview of the regionalised content-based retrieval framework**

In this work segmentation is used to achieve as homogeneous regions as possible for meaningful region and spatial descriptors. Also note that advanced methods (e.g. [18], [20], [78], and [89]) will probably produce better results than simpler ones (i.e. segmentation by threshold). It is actually desired to tune the segmentation method in order to obtain over- segmentation errors rather than under-segmentation because the *grouping* step is designed to correct such over-segmentation faults. This step is performed by grouping adjacent and highly similar regions where the similarity distance is computed by using the local features such as colour and texture. Nevertheless, under-segmentation faults should be avoided at all costs since they are unrecoverable in the proposed framework. The *grouping* stage is separated from the actual segmentation process since any segmentation method may be utilised, basic or advanced, so as to keep the aforementioned "black-box" property of the framework. Hence, this step provides an option to improve segmentation results, if feasible, which will also reduce the complexity during retrieval due to the reduced number of regions.

After the *grouping* step, local region features can be extracted. Since the *local feature* stage is considered as a black-box, any visual descriptor can be extracted, and describe the regions based on some low-level features. Due to region-based approach, descriptors should be applicable to any arbitrary-shaped region. To avoid the feature extraction region by region, a back-projection approach is employed where the features are first extracted over the entire frame and then back-projected to the regions. This approach has one motivation. There might be visual descriptors, which are not directly applicable to an arbitrary-shaped region. An additional step such as padding regions into rectangular shapes [53] for a texture descriptor is therefore not required with this approach.

Generally speaking, extraction of texture features might be computationally complex. Therefore, a texture indicator is used to detect if there is any texture present in the image, and, thus the decision of applying texture extraction over all regions is made accordingly. After the *local feature* extraction stage, the so-called region proximities are extracted and utilised along with the aforementioned regionalised visual features. The main intention to use them is to spatially exploit the image regions by describing their relationship among each other. The basic assumption is that images with similar regions (locally) as well as similar region relationships (spatially) are perceived as similar. Yet two regions, which are not similar locally, may yield a certain image similarity if their surrounding is similar. Surroundings for such two regions mean that both have regions with similar local and spatial region properties in their proximity. Figure 4.2 illustrates a simplified scenario of such a region (spatial) similarity where locally dissimilar (white, brown, and black) regions with similar neighbour regions are present in two similar scene compositions. In this example although the regions do not match locally, the images may be considered similar.

**Figure 4.2 - Sample Images illustrating the idea of similarity based on region surroundings**

Some traditional spatial features (region-based) are presented in section 2.1.5. One major difference of the region proximity feature is that the spatial property between two regions is expressed by region proximities (average distance between region pairs) without any

directional and topological descriptions. To describe the average distance between two regions, a coarse approximation is sufficient rather than a high-resolution (e.g. in pixel accuracy) distance measure to allow a simple discrimination between small and large distances. Therefore, a block-based region representation is applied. The proposed framework integrates variable block size region representation for the region proximity feature based on two motivations. The first and the major one, the region representation is independent from the image dimensions. Imagine two images with different dimensions displaying the same content. In the fixed block size case, the regions will have a completely different block representation from each other whereas in the variable block size case, the images will have the same block-wise region representation as illustrated in Figure 4.3. The second motivation derives from the first one in the sense that for images with high dimensions the regions should be represented and approximated in an efficient way so that the distance calculations does not require a high computational complexity. Due to this reason, a block-based representation with a particular block size (e.g. 4-by-4 pixels) is not considered. Furthermore, traditional region properties such as CoM and MBR are also not suitable for calculating region distances due to there earlier mentioned drawbacks. Moreover, an option such as utilising all region pixels would not be effective since such a pixel based representation causes high computational complexity describing the region distance particularly for large regions.

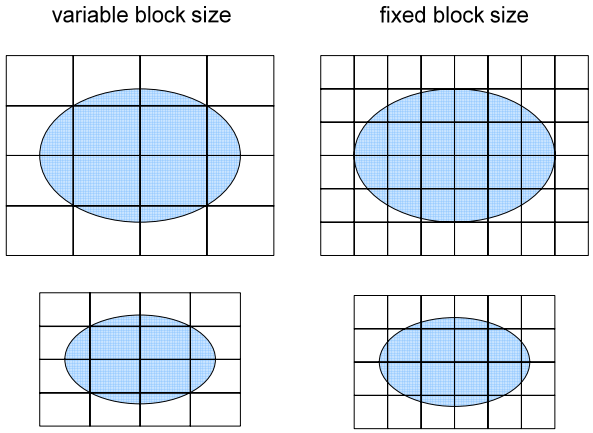


**Figure 4.3 - Block-based region representation**

Based on the planar 2D image representation the following abstraction is made to calculate the distance between two image regions. The region distance should take into consideration the important region properties such as the region area, shape, and location. In order to accomplish this, an average region distance over the block-based region

representation is introduced rather than applying the commonly known Hausdorff distance.

The distance measure applied between two regions is expressed as follows:

$$h_{\Sigma}(A,B) = \frac{\sum_{a \in A} \min_{b \in B} \left( |a-b| \right)}{N}$$
$$H_{\Sigma}(A,B) = \min \left\{ h_{\Sigma}(A,B), h_{\Sigma}(B,A) \right\} \tag{14}$$

where $a$ and $b$ are region blocks of the block-based region representations $A$ and $B$ and $N$ denotes the number of blocks for $A$. Moreover, $h_{\Sigma}(A,B)$ is the direct distance for each region, $H_{\Sigma}(A,B)$ is the overall distance between both regions, and $|\cdot|$ represents the $L_{\infty}$-norm distance between two region block indices. The motive of applying this distance is that due to the average operation it takes region shape and area better into consideration than the Hausdorff distance. Especially for concave shapes, the drawback of the Hausdorff distance is that it will depend on the region shapes and their area distribution due to its maximisation operations. Furthermore, combining this approach with the block-based region representation leads to an intended coarse distance approximation. The calculation of $H_{\Sigma}(A,B)$ assures the symmetry property, i.e. $H_{\Sigma}(A,B)=H_{\Sigma}(B,A)$. Figure 4.4 (a)-(d) demonstrate the difference between the Hausdorff distance and the proposed distance with their advantages and drawbacks. For the example in (a), the Hausdorff distance provides a large distance even though the two regions are rather close. Moreover, the examples in (b) and (d) illustrate the drawback of the maximisation operations in the Hausdorff distance for concave region shapes and enclosing region constellations. For these three cases (a), (b), and (d), the approach of the proposed distance returns an enhanced distance measure based on region shape (b) and constellation (d). The example in (c) demonstrates the drawback of the minimisation operation in $H_{\Sigma}(A,B)$ for certain region shapes and their constellation. Moreover, Figure 4.5 demonstrates the $H_{\Sigma}(A,B)$ distance (green line) for two horse regions in a natural image. After the distances between all possible region pairs are calculated, each region holds distance information to any other region in the image where distances are unit normalised by the maximum possible block distance, 32.

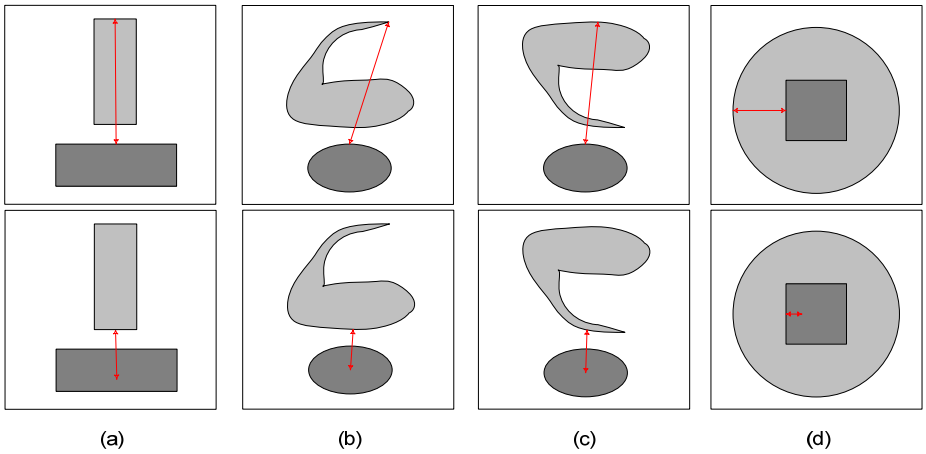


**Figure 4.4 - Examples for distance calculation Hausdorff distance (top), proposed distance (bottom)**

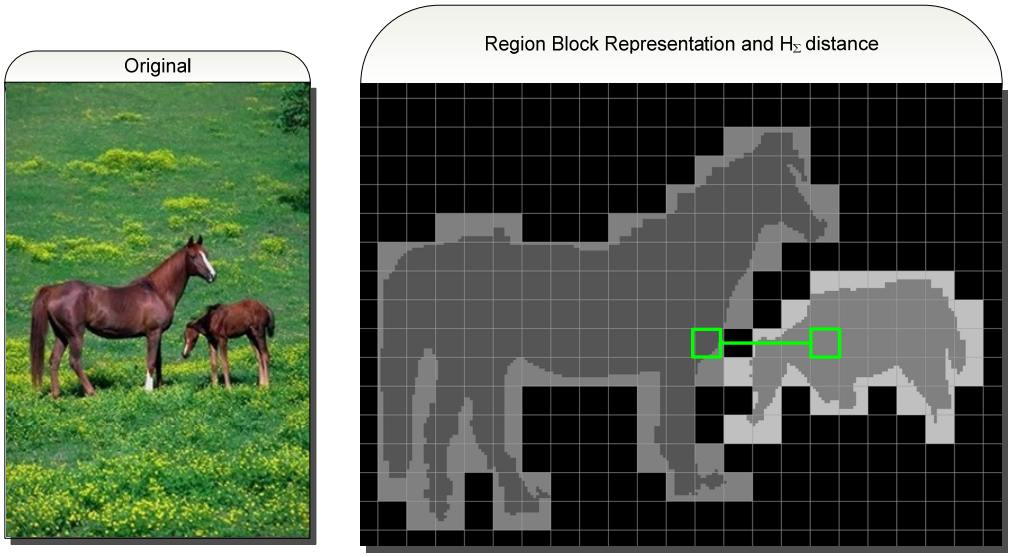


**Figure 4.5 - Example of a natural image with its region representation and distance between two regions**

The extracted features are stored in a feature vector representing all image regions with their local and spatial features. During retrieval, the overall idea is a similarity maximisation approach. Similarity is computed from the distance between features and it can be used as a matching criterion between features, regions as well as images. Image similarity between two images comes from the cumulative region similarities, where each region in a query image $I_Q$ tries to find its best match in the regions of a database image $I_T$ based on the earlier mentioned extracted features. The approach can be seen as a many-to-one matching scheme where several regions from $I_Q$ might find the same region in $I_T$ as their best match. The overall image similarity depends on the matching between regions of $I_Q$ and $I_T$, which means that higher region similarities result in higher matching between regions (and vice versa), which will eventually determines the overall similarity score between two images.

High region similarity is achieved if all features colour, texture, and region proximity provide a high feature similarity. Hence, the fewer of those three features provide high similarities, the lower the region similarity and the lower the contribution to the image similarity, eventually.

## 4.2 The Prototype System

After introducing the general framework approach, one of many possible combinations of segmentation methods and visual feature descriptors is presented as a prototype implementation. We employ quad-tree region splitting [35] colour segmentation, also known as Split and Merge (SPMG), and in the *local feature* stage colour and texture features are extracted and described by DCD and Gabor filter, respectively. They are used due to their efficient and compact description. The following subsections detail the prototype system where sections 4.2.1 to 4.2.5 present the indexing phase and section 4.2.6 presents the retrieval phase.

### 4.2.1 Segmentation

The *segmentation* stage is separated into several consecutive steps: pre-processing by bilateral filtering [86], SPMG, and post-processing where the original SPMG and the post-processing step form the modified quad-tree segmentation (mSPMG) as illustrated in Figure 4.6. Since SPMG is a colour segmentation method, bilateral filtering is first applied to the image to degrade noise and high frequency information to speed up the SPMG algorithm. Bilateral filtering was chosen due to its edge-preserving smoothing as shown in Figure 4.6.

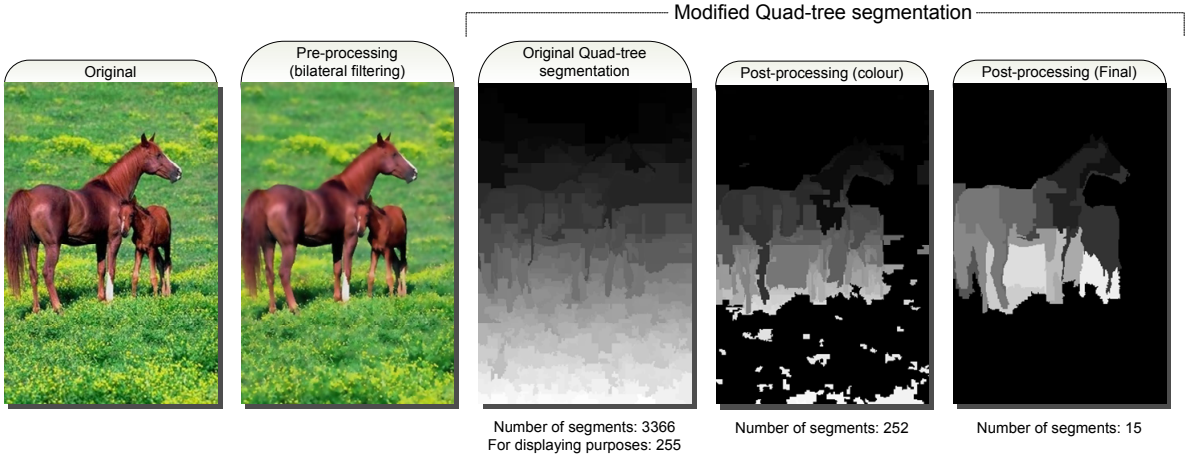


**Figure 4.6 - Segmentation steps with example image**

After this pre-processing step, the SPMG method is applied. The main idea is to segment the image by quad-tree splitting it into several homogeneous sub-images based on a homogeneity criterion such as the variance of colour distribution in this implementation. The splitting is achieved by starting with the entire image, split it into quadrants, and then process these quadrants and so on. After the splitting phase for a sub-image is finished, a merging phase for this particular sub-image takes place to merge adjacent regions if they satisfy the homogeneity criterion. The general algorithm can be described as follows:

***Split*** *(parent)*

- calculate quadrants homogeneity criterion
- if criterion < splitting threshold
  - mark complete quadrant as a region; **Return**
- else
  - split further, let $SI^0$ = *parent*
  - For $\forall p \in [1,\ldots,4]$ do:
    - ***Split***( $SI^p$ )
  - ***Merge*** *(parent)*
    - merge regions of quadrant *parent*; calculate new homogeneity criterion
    - if new region homogeneity criterion < splitting threshold
      - merge
- **Return**

and Figure 4.7 illustrates an example of this algorithm where the sub-images are proceeded from top to bottom, left to right and the Xs mark the quadrants, which have not been processed yet. Sub-images labelled with *S* belong to the splitting phase and with *M* to the merging phase.

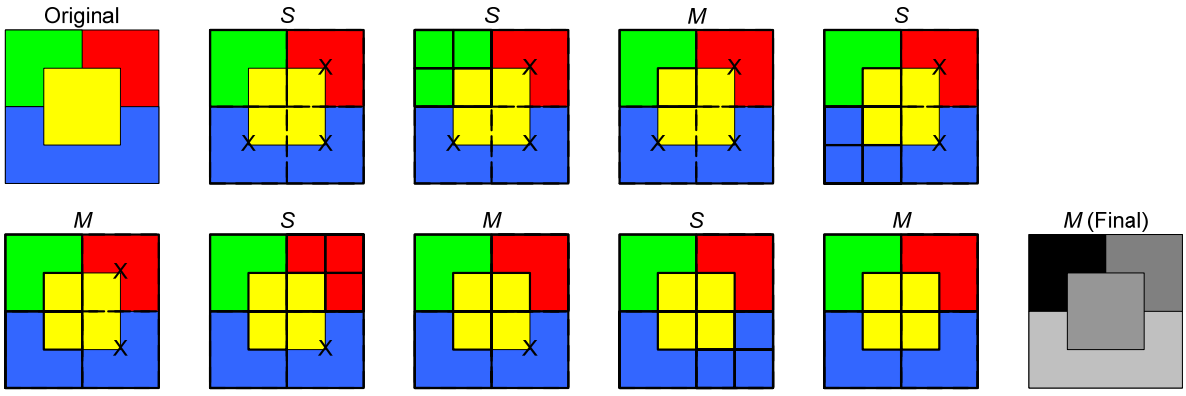


**Figure 4.7 - Example of quad-tree (split and merge) segmentation**

The maximum quad-tree depth is also an important factor. One possibility is going down to the pixel-level achieving best possible region boundary representation or alternatively stopping if a sub-image is smaller than or equal to a pre-defined block-size (depth) such as 4-by-4 or 8-by-8 for coarser region border approximation, which is faster but presents some blocking artefacts. In order to obtain the best possible region boundary representation, the quad-tree splitting algorithm in this work splits down to the pixel-level, which unfortunately increases complexity especially for noisy or fine-textured images. Due to the applied homogeneity criterion (variance), this SPMG segmentation may result in highly over segmented images with a large number of regions such as the example in Figure 4.6. To reduce the number of regions and hence obtain a suitable and useful segmentation, some SPMG post-processing is employed, which can be divided into three steps: merging by colour, merging by size, and, finally, merging to a maximum number of segments. The first step is achieved by merging regions based on their average colour difference until a certain threshold, *cThr*, is reached. The average colour difference is defined by the normalized Euclidean distance as,

$$d\left(R_i, R_j\right) = \frac{\sqrt{\left(C_i^1 - C_j^1\right)^2 + \left(C_i^2 - C_j^2\right)^2 + \left(C_i^3 - C_j^3\right)^2}}{\min\left(\sqrt{\left(C_i^1\right)^2 + \left(C_i^2\right)^2 + \left(C_i^3\right)^2}, \sqrt{\left(C_j^1\right)^2 + \left(C_j^2\right)^2 + \left(C_j^3\right)^2}\right)} \tag{15}$$

where $C_i^n$ is the average colour value of the n$^{th}$ colour component of the applied colour space for region $R_i$. Figure 4.6 shows an example after applying merging by colour, which illustrates the merging effect reducing the region number from 3366 to 252. In the second step if the area of a region is smaller than one per cent of the total image size then this region is merged to one of its neighbours based on the best colour matching criterion. This is based on the assumption that such tiny regions are not perceived by the human eye [64]. The final step merges the remaining number of regions until a pre-fixed maximum number of segments, *maxS*, is reached. This limit, *maxS,* is an approximated pre-fixed value. This third step employs a simple method, which can be seen as a worst case scenario. The regions are sorted by size, and then the *maxS*-th largest region size is used as a threshold to re-perform the second step. However, to avoid that each and every image ends up with *maxS* number of segments, the underlying SPMG algorithm plus the first two post-

processing steps should already achieve a reasonable number of regions so that the last step may not be needed.

Note that this segmentation method is not fully automatic due to few parameter dependencies. This means that the parameters that are empirically determined are not suitable for all image types. Moreover, since mSPMG is only based on colour; better results might be achieved by methods, which further use texture and/or edge properties.

### 4.2.2 Grouping

After *segmentation*, the following process merges over-segmented regions, if exists;

***Grouping*** *(list of image regions* $R^N$*):*

- calculate similarities *S(i,j)* between all adjacent regions
- Repeat:
  - find maxSim = max{*S(i,j)*} and group regions *i* and *j*
  - update similarities between merged region and its new neighbours
- Until maxSim < simThr

In this grouping process if two regions achieve higher similarity score than the grouping threshold by their local properties (features) they are grouped. Furthermore, there is no processing order (i.e. left to right) for the grouping so that region locations do not play any role. Possible cases where grouping might take place and enhance image regions are illustrated on synthetic examples in Figure 4.8. In the figure with all three cases, segmentation performs fairly good, yet leaving some margins for improvements. Figure 4.8 (a) illustrates a case when region similarity differences between adjacent regions in the upper image part are slightly changing, Figure 4.8 (b) shows when several similar regions are adjacent to the same region and Figure 4.8 (c) displays a case where region similarity differences between adjacent regions are first increasing and then decreasing again. In all three examples, *grouping* provides an enhanced segmentation scheme with less homogeneous regions closer to real objects. Figure 4.9 shows a grouping example of a natural image of brown horses on a green field. It can be seen that *segmentation* method produces the main regions; however, results in over segmenting the objects, especially the bigger horse due to the colour variations on brown surface. From a certain point of view, segmentation does provide homogeneous regions but these brownish shades can be seen as similar, thus better regions may be obtained. Thus *grouping* can provide a difference in the final region representation as seen in Figure 4.9 where the number of regions is

decreased from 15 to 3 and they are closer to representing meaningful objects or their major parts. *Grouping* method uses both colour and texture features, which are described in section 4.2.3. Moreover, it might lack the same parameter issues as the employed segmentation method since as a stopping criterion for the grouping process thresholds are used for measuring region-based colour and texture similarities. This means that the empirical parameter values might produce different results for different image content. However, they can be experimentally set in such a way that for certain image types and categories with representative and distinguishable objects, object-based image regions may be achieved.

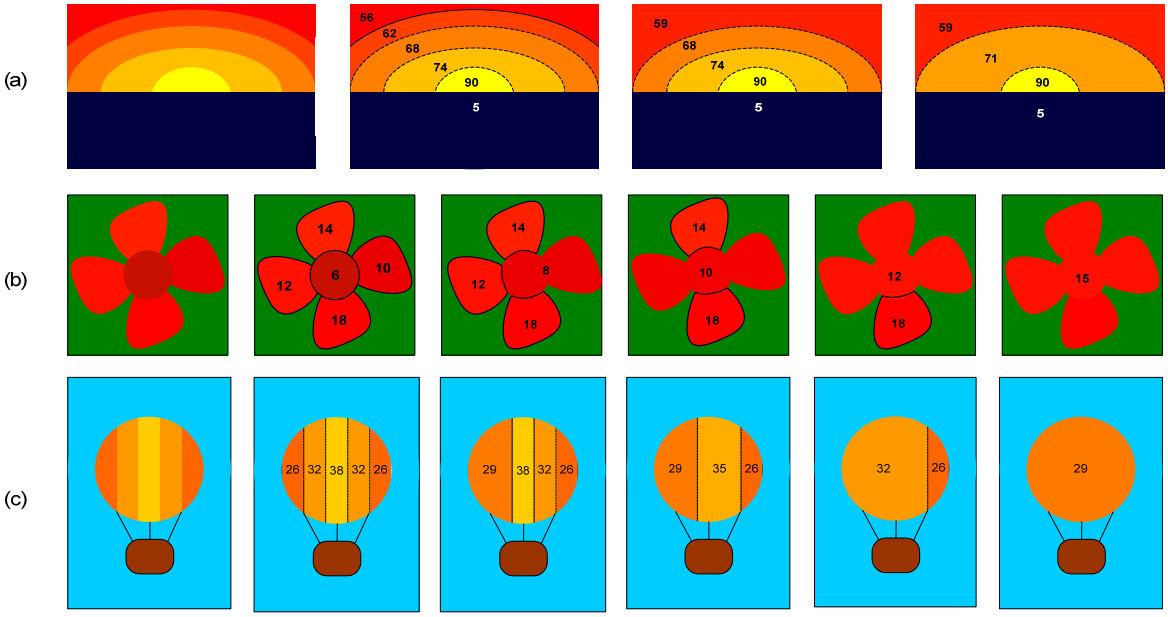


**Figure 4.8 - Examples of possible grouping illustrated by synthetic images**

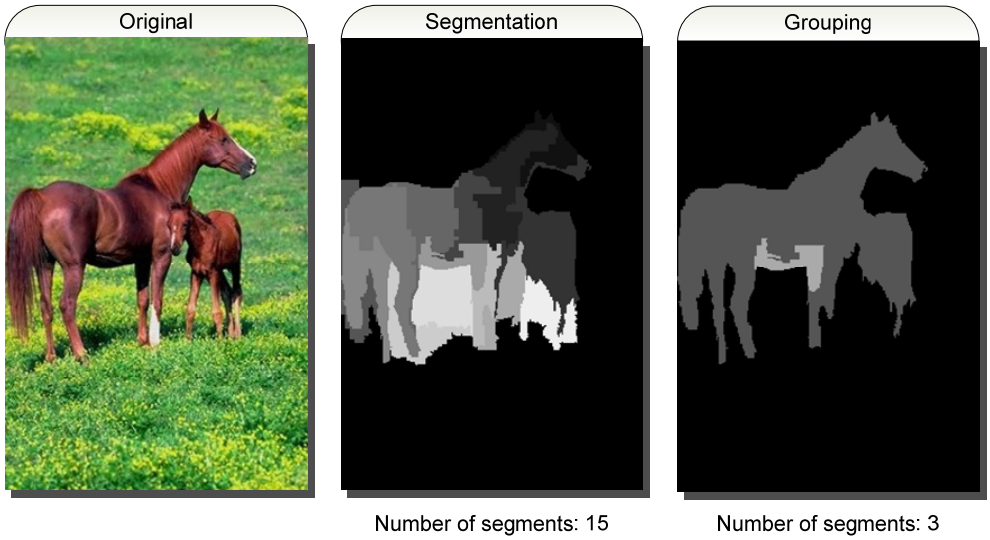


**Figure 4.9 - Grouping effect illustrated on a natural image**

### 4.2.3 Local Features

As stated earlier, visual descriptors, which are applied after obtaining the final regions, are versatile. In this implementation, the chosen colour and texture descriptors are DCD and Gabor-Filter, respectively. Shape information is not purposefully integrated since most segmentation methods do not produce meaningful regions from which meaningful shape information can be obtained.

The DCD and Gabor filter descriptors are utilised by the aforementioned back-projection approach. For DCD, in order to extract the dominant colours of an image, a dynamic colour quantisation -in the CIE-Luv- space is employed by the so-called General Lloyd Algorithm also known as K-Means clustering [57]. In this approach, colours are quantised coarsely in detailed regions and finer in smoother regions. A cluster $C_i$ is represented by a colour centroid $c_i$, and the initial K-Means clusters are determined by a weighted distortion measure $D_i$. Both, $c_i$ and $D_i$, are calculated as follows:

$$c_i = \frac{\sum w(p) \cdot x(p)}{\sum w(p)}, \quad x(p) \in C_i$$
$$D_i = \sum w(p) \left\| x(p) - c_i \right\|^2, \quad x(p) \in C_i \tag{16}$$

where *w(p)* is a smoothness weight for each pixel *x(p)* in a local window. The process of splitting the clusters is based on the distortion measure $D_i$. This process stops if either a maximum number of clusters (DCs) or a maximum allowed distortion ($\varepsilon_D$) is reached. Hence, colours represented by a smaller number of pixels (detailed regions) are assigned to fewer clusters. Finally, agglomerative clustering is applied where similar colour clusters are merged together so that they are presented by a single cluster. Merging of clusters is based on two criteria, a similarity threshold, $T_S$, and colour area, $T_A$, where $T_S$ describes the maximum colour distance allowed between two clusters, and $T_A$ describes the minimum area covered by a colour to be considered dominant. Otherwise it is just considered as an outlier and, therefore, merged to its closest (dominant) colour cluster.

After the DCs are extracted, they are back-projected onto the image pixels by replacing a pixel colour with the closest DC. The region DCs are simply obtained by accumulating the DCs over the region pixels. There are then represented as a set $DC_i : \{c_i, w_i\}$ by the DC centroid $c_i$ and the local colour weight of $c_i$ in the region. In order to compute the colour similarity distance between two DC sets, the quadratic distance formulation is applied as presented in section 2.2.2. A sample DC extraction and back-projection onto regions is

shown in Figure 4.10 for the sample image in Figure 4.9. The red lines indicate the major colours mapped to the horse (black, brown) and field (light-green, dark-green) regions.

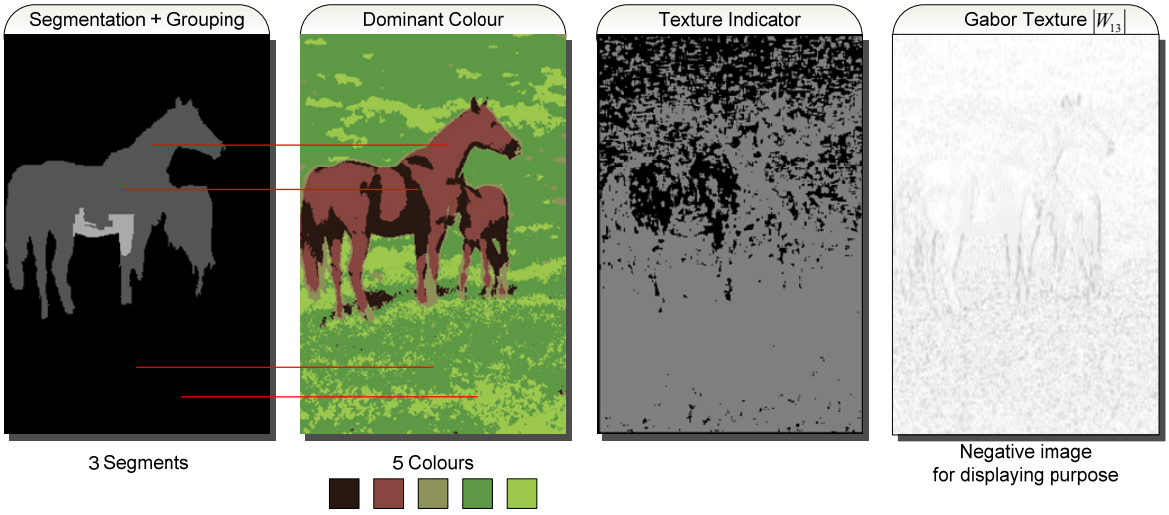


**Figure 4.10 - The local features extracted over the entire frame**

The regionalised texture description is also obtained by the aforementioned back-projection approach. At first, a Gabor-Wavelet-Transform as described in section 2.1.3 is applied over the entire image in frequency domain using three scales and four orientations as a trade-off between complexity and performance. This results in 12 magnitude responses $|W_{mn}|$, one for each scale and orientation. Such a response is shown in Figure 4.10 for the first scale and third orientation, $|W_{13}|$. From each of those $|W_{mn}|$, the mean and standard deviation are calculated per region. To avoid bias by the region boundaries, pixels are excluded from calculation if their 3-by-3 pixel neighbourhood is not completely covered by the same region. This produces a region-based texture feature vector with size 24. Similarity distance between two texture feature vectors is computed by the Euclidean distance.

The texture indicator is a Boolean operator, whose outcome indicates if an image pixel carries texture information or not. The main approach is to check each N-by-N pixel neighbourhood, horizontally and vertically, for local extrema (minima and maxima) as illustrated in Figure 4.11. If within this neighbourhood, the overall amount of local extrema exceeds a certain threshold then the texture indicator is set for this pixel. To minimise the influences of noise and region contours in this approach, two modifications are made: First, the luminance values of the image are statically quantised (to multiples of 5), which degrades noise but leaves the dominant texture parts intact, and then the pixel

neighbourhood is reduced via setting N=5 in order to degrade the influence of region contours to nearby region pixels. After indicating if an image pixel carries texture information, this information is stored in a global texture indicator map such as shown in Figure 4.10. Based on this map, the pixel-based texture information is back-projected to the regions. By excluding pixels whose 5-by-5 pixel neighbourhood does not lie completely within the region, the bias at region borders is intended to be reduced. By this back-projecting, the ratio of textured region pixels and total number of region pixels is taken as measure for the degree of texture within a region. As the final outcome, the texture indicator will give the texture degree of a region as a value in a range of [0, 1] where zero (0) denotes no texture at all and one (1) indicates a completely textured region.

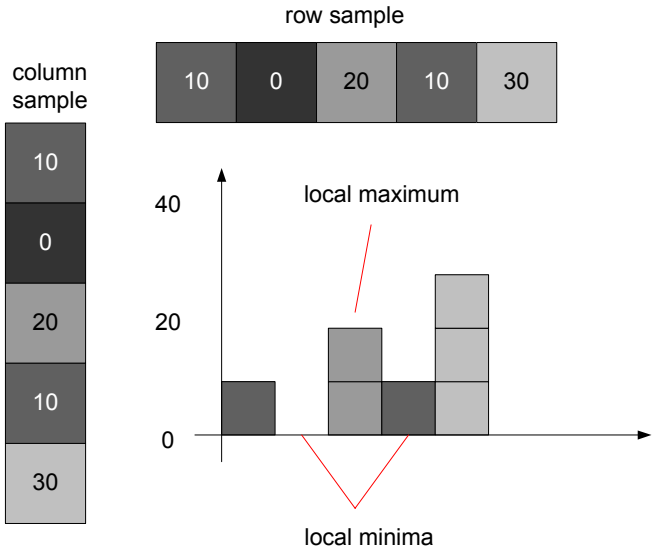


**Figure 4.11 - Local extrema for row and column sample in a 5x5 pixel neighbourhood**

### 4.2.4 Spatial Feature: Region Proximity

For variable block size partitioning the block size should not be too small or too big by the pre-fixed grid. If it is too small then the block size becomes too big, which degrades an efficient region representation, or if it is too small then this may increase computational complexity. Therefore, a grid of 32-by-32 blocks is employed, which promises a trade-off between the accurate region representation and efficient region proximity (distance) computation. The final region representation by blocks is obtained in the following two steps.

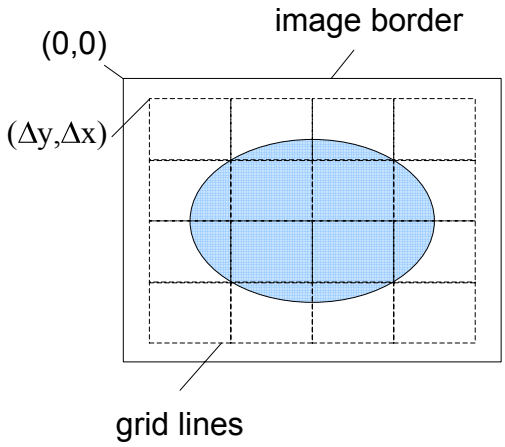


**Figure 4.12 - Superimposed block grid on image**

Firstly, the image is divided into the aforementioned grid. For simplicity, the grid is superimposed on the image in such a way that all blocks have an equal size, which may require discarding some of the border pixels if image dimensions are not multiples of 32. In the worst case, this means that 31 pixels are ignored, horizontally and vertically, as shown in Figure 4.12 where (0, 0) points to the top-left corner of the image and ($\Delta y$, $\Delta x$) points to the translated starting position of the superimposed grid. Discarding some boundary pixels is validated by the assumption that such an operation will not affect the final outcome since region proximity values are an approximation due to block-based approach. However, it might happen that due to this translated grid a border region cannot be represented by any blocks. Then the assumption is extended in a manner that such a thin region close to the image boundaries is not relevant to the image content and, therefore, can be left out. Afterwards superimposed grid blocks are assigned to regions. For the sake of simplicity, a block belongs to a region as long as at least one pixel of this region lies within that block. Therefore, this means that one block can be assigned to more than one region as seen in Figure 4.13. However, a slight drawback of this block-region assignment is that a region might have blocks representing just one or a few region pixels, e.g. see in Figure 4.13 especially around the head and the front leg of the bigger horse. However, the effect of this is negligible for the distance calculation. The distance calculation between two arbitrary shaped regions is implementation based on Eq. 14.

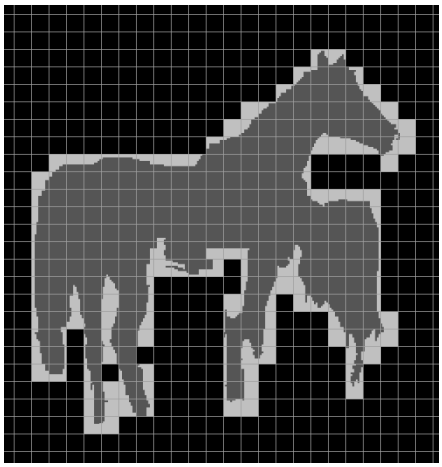

**Figure 4.13 - Variable block size grid superimposed on image**

### 4.2.5 Formation of the Feature Vector

After extracting the aforementioned local (colour and texture) and spatial (region proximity) features from each region in an image, a feature vector (FV) is formed for indexing and retrieval purposes. The FV presents a logical structure to represent the image content in form of its regions and their features. Therefore, it ought to combine all necessary information required during the similarity distance computation in retrieval process. Figure 4.14 outlines the feature vector structure implemented. The first three values are general information followed by the actual information for all image regions. The general information are the number of regions (N) in the image, a regulation coefficient (regCoeff) for retrieval normalisation, and the maximum colour similarity value between two colour sets $\left(D_{max}^{DC}\right)$, whose utilisation is described in more detail in section 4.2.6. The region information is stored in form of feature quartets, namely *area, colour features, texture features*, and *region proximity distances*. The *area* is (unit) normalised by the image size. The other three parts of the feature vector are stored and represented by the aforementioned extracted feature values. For the colour description the values are number of DCs ($N_{DC}$), centroid $c_i$ components, and local region weight $w_i$ for each DC. For the texture description, alpha ($\alpha$) as the texture indicator, mean and standard deviation for each Gabor magnitude response are stored. Region proximity description saves all distances to the other image regions. In the current module, the feature vector size needs to be known before hand due to the MUVIS FeX framework requirements

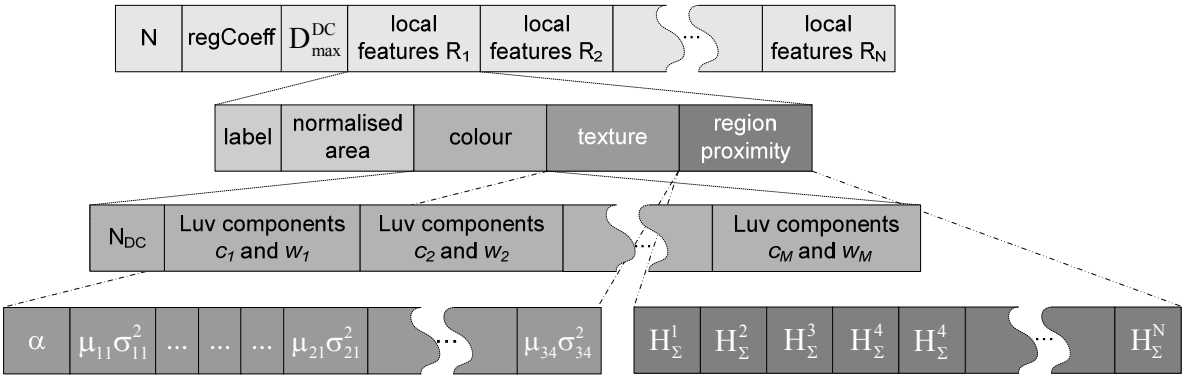


**Figure 4.14 - Formation of the feature vector**

### 4.2.6 Region Matching

The similarity between two images is the cumulative sum of all region similarities, where regions try to maximise their similarity scores. The region similarity score, *Sim(i,j),* is formulated as follows

$$\begin{aligned} Sim(i,j) &= S_L(i,j) + S_S(i,j) \\ S_L(i,j) &= \alpha \cdot S_{DC}(i,j) + (1-\alpha) \cdot S_{Text}(i,j) \end{aligned} \tag{17}$$

where $S_L(i,j)$ is the region-based local similarity score, $S_S(i,j)$ is the spatial similarity score and $S_{DC}(i,j)$ and $S_{Text}(i,j)$ are similarity scores for colour and texture features, respectively. $\alpha$ specifies the texture indicator as explained earlier. The region similarity is expressed as the weighted sum of other similarities (local + spatial) each of which is computed from feature vector distances (1-*d*).

Without the application of $\alpha$, colour and texture are weighted equally during measuring region similarities. To overcome the biasing effect of non-textured and heavily textured regions, $\alpha$ is utilised in the following manner. For non-textured regions, only colour contributes to the region similarity and for heavily textured regions ($\alpha<½$), colour and texture are weighted equally ($\alpha=½$) so as not to only rely on texture alone.

For DCD, similarity, $S_{DC}$, is derived from quadratic distance measure with a small adjustment to the general approach presented in [58]. There, inter similarities between colours are represented by elements $a_{ij}$ of matrix A, which are computed by $a_{ij} = 1 - \frac{|c_i - c_j|}{T_S}$ where $c_i$ and $c_j$ are the colour centroids and $T_S$ the maximum colour distance until two colours are considered similar. The above calculation of $a_{ij}$ directly uses $T_S$ from the

colour clustering procedure as its minimum similarity bound so that colours are dissimilar if $|c_i - c_j| > T_S$. There is just one major drawback with this due to the human colour perception, which is rather smooth, not step wise or abruptly changing. Based on the $a_{ij}$ calculation approach in [58], all colour clusters where $|c_i - c_j| > T_S$ have the same dissimilarity. However, separated colour clusters as illustrated in Figure 4.15 (a)-(e) may have different (dis-) similarity among each other especially for same colour values with different saturations such as Figure 4.15 (a)-(c). Therefore, a new empirically $T_S$ $\left(D_{max}^{DC}\right)$ is set to allow a smoother colour similarity $\left|c_i - c_j\right| < D_{max}^{DC}$ between two colour sets.

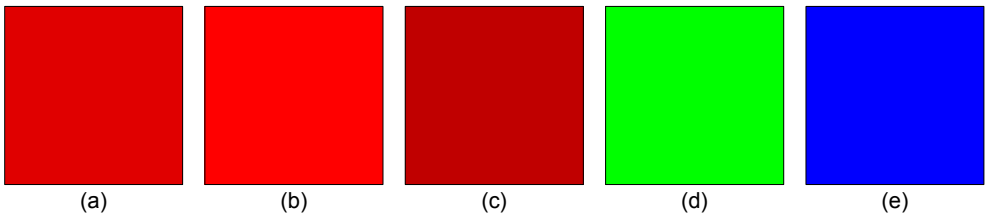


**Figure 4.15 - Colour patches: (a) red, (b) lighter red, (c) darker red, (d) green, (e) blue**

The spatial feature similarity, $S_S$, only considers image regions in $I_Q$ and $I_T$, which achieve a similarity higher than an empirically set threshold based on their local colour and texture features. A high $S_S$ for a region $i$ in $I_Q$ is achieved if there exists a region $j$ in $I_T$, which has regions with similar local features and similar region area weights in short distances in its proximity as region $i$. Therefore, $S_S$ formulation integrates the area and the distance of the similar surrounding regions to model the region constellations between region $i$ and $j$. Hence, this similarity is expressed as,

$$S_S(i,j) = \underbrace{\overbrace{\left(\frac{\min(w_l, w_k)}{\max(w_l, w_k)}\right)}^{\text{area ratio}} \cdot \overbrace{\left(\frac{1+\min(w_l, w_k)}{2}\right)}^{\text{area size}}}_{\text{area factor}} \cdot \underbrace{\overbrace{\left(\frac{\min(H_\Sigma^l, H_\Sigma^k)}{\max(H_\Sigma^l, H_\Sigma^k)}\right)}^{\text{distance ratio}} \cdot \overbrace{\left(1-\max(H_\Sigma^l, H_\Sigma^k)\right)}^{\text{distance}}}_{\text{distance factor}} \quad (18)$$

where $i$ and $j$ are the regions to match, $l$ and $k$ the similar local-feature-based regions in the surroundings of $i$ and $j$, respectively. Moreover, $w$ denotes the region area weight and $H_\Sigma$ is the region proximity distance. The labels *area factor* and *distance factor* represent the terms for measuring the influence of area and region distance, respectively, to favour surrounding regions with large region area weights and close region proximities. The two parts are further divided into terms of *area ratio* and *area size* as well as *distance ratio* and *distance*. The *area ratio* term favours regions of equal size and *area size* term further

favours larger region areas in the contribution of similarity score, $S_S$. In other words this means that the *area ratio* term punishes unequal region areas, and the *area size* term punishes small regions. Similarly, the *distance ratio* term favours distances of equal length and *distance* term further favours short distances in the contribution of similarity score, $S_S$. Thus, the *distance ratio* term punishes different distances and the *distance* term punishes large distances.

After calculating the similarity *Sim(i,j)* between each region in $I_Q$ and each region in $I_T$, the final region similarity comes from the region pair *(i,j)* with the highest similarity (maximum similarity score) *Sim(i,j)*. This is the part where similarity is maximized, or in other words, "similarity hunting" is applied in order to maximize each individual region similarity. Once the maximum possible similarity for each single region is obtained, the (overall) image similarity is the sum over all region similarities. The region similarities are weighted by a Region Area Factor (RAF) to determine the importance of each *Sim(i,j)* to the image similarity score. Here, the region importance is simply modelled by the region area. The overall image similarity between $I_Q$ and $I_T$ is then defined as,

$$\begin{aligned} RAF(i,j) &= \min(w_i, w_j) \\ TS(I_Q, I_T) &= \sum_{i \in I_Q, j \in I_T} RAF(i,j) \cdot Sim(i,j) \\ TS_\Sigma(I_Q, I_T) &= \frac{TS(I_Q, I_T) + TS(I_Q, I_T)}{2} \end{aligned} \tag{19}$$

where *RAF(i,j)* denotes the Region Area Factor for regions *i* and *j* based on their region area weights $w_i$ and $w_j$. Moreover, $TS(I_Q,I_T)$ represents the image similarity score for $I_Q$ and $I_T$ and $TS_\Sigma(I_Q,I_T)$ is the total image similarity score between two images.

The motivation of using the minimum of region area weights for the *RAF(i,j)* is to assure that two regions only match the area, which is covered by the smaller region. Moreover, utilising $TS(I_Q,I_T)$ and $TS(I_T,I_Q)$ for $TS_\Sigma(I_Q,I_T)$ assures that each region in either image finds its best match and, therefore, contributes to the total image similarity. Figure 4.16 (a) and (b) demonstrate this behaviour on two images, which illustrate two similar regions partitioned into different number of segments. Based on the many-to-one matching scheme, several regions in the left image ($I_Q$) in (a) find their best match in the same region of the right image ($I_T$). Due to considering only the region pair with the highest similarity score, some regions of $I_T$ in (a) are unmatched and so do not contribute to the image similarity. To overcome this problem, $TS(I_T,I_Q)$ is also calculated. However, the

main motivation for calculating $TS(I_Q,I_T)$ and $TS(I_T,I_Q)$ for $TS_\Sigma(I_Q,I_T)$ is the similarity symmetry

$TS_\Sigma(I_Q,I_T) = TS_\Sigma(I_T,I_Q)$. Figure 4.17 demonstrates the region similarity on some sample natural images with brown horses on a green field. It shows the highest region similarities of the two main regions (horse and field) of the query image to corresponding regions of three target images with similar sceneries. The red lines indicate which regions in the query and target images correspond to the highest region similarities.

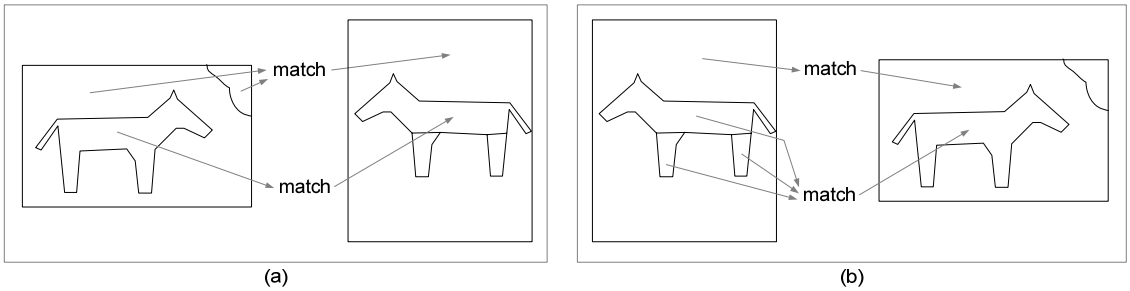


**Figure 4.16 - Region matching scheme: (a) $TS(I_Q, I_T)$, (b) $TS(I_T, I_Q)$**

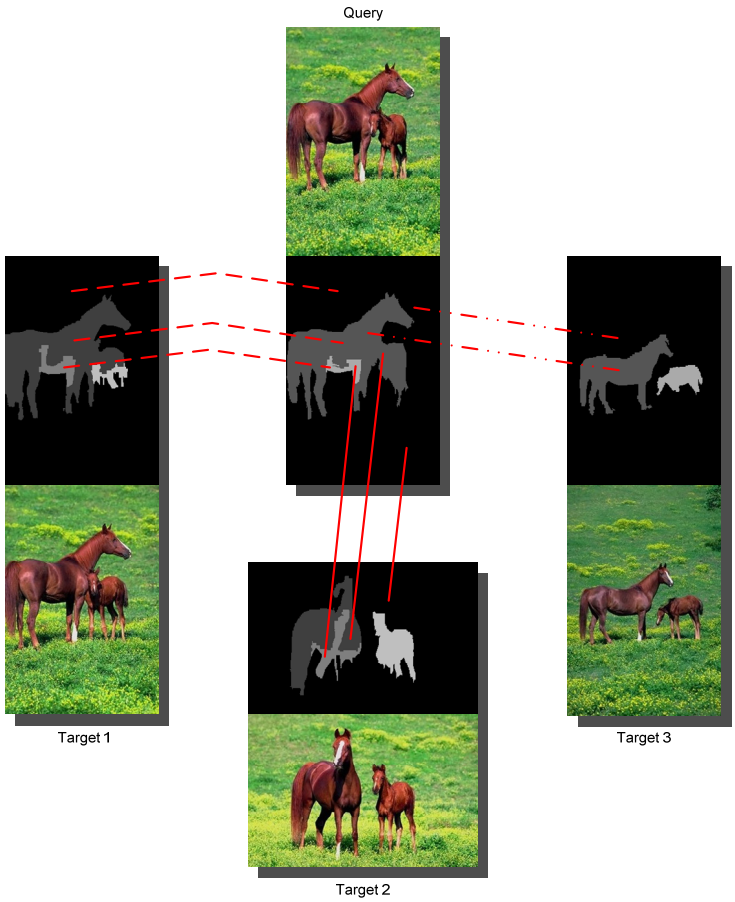


**Figure 4.17 - Region matching example for two regions in a natural horse image**

# 5 Experimental Results

Retrieval performance of the prototype system is evaluated as a FeX-module in the MUVIS framework. Therefore, MUVIS applications *DbsEditor* and *MBrowser*, described in section 3.1.1, are used for indexing and retrieval purposes, respectively. The FeX module is configured with 10 parameters as listed in Table 5.1, all of which are used in the four aforementioned stages -first three to *segmentation*, next two to *grouping*, following three to dominant colour extraction (DCD), and last two to texture extraction (Gabor-Wavelet-Transform). All indexing and retrieval tasks are executed on a Pentium4-3GHz computer with 1 GB of RAM.

**Table 5.1 - FeX module parameters**

| | | | |
|---|---|---|---|
| **maxS** | maximum number of segments | **maxC** | maximum number of DC clusters |
| **var** | variance threshold for SPMG | **minD** | colour distance ($T_S$) for clustering |
| **cThr** | colour threshold for post-processing | **simD** | colour distance ($T_S$) for retrieval |
| **simC** | colour similarity threshold for grouping | **sc** | number of scales for Gabor-Wavelet |
| **simT** | texture similarity threshold for grouping | **or** | number of orientations for Gabor-Wavelet |

In the following subsections the evaluation of the results for synthetic and natural images are presented and discussed.

## 5.1 Results on Synthetic Images

The motivation to test synthetic images is to show that the general theory of the proposed approach performs as expected when there are no segmentation faults. Furthermore, they are especially used to test the robustness and accuracy of the region proximity feature.

Firstly, there are two synthetic tests conducted. The first test is to illustrate the effect of the region proximity feature for images with the same local feature proportions. Therefore, the test uses a set of 12 images, shown in Figure 5.1. The second test uses a set of 11 images, shown in Figure 5.2, to demonstrate the effect of region proximity based on surrounding regions. During indexing of those synthetic images, we used the parameters presented in Table 5.2. Since those images do not contain any texture, parameters marked with an X are set in such a manner to void the contribution of the texture over the similarity score.

**Table 5.2 - FeX parameters for synthetic tests**

| maxS | var | cThr | simC | simT | maxC | minD | simD | sc | or |
|---|---|---|---|---|---|---|---|---|---|
| 25 | 50 | 0.25 | 0.6 | X | 8 | 15 | 45 | X | X |

The results for the first image set are shown in Figure 5.1 where the top row shows results for circles in a close proximity and vice versa for the bottom row (first image is the reference and from left to right the results are sorted according to their similarity score with respect to the reference image). It can be clearly seen that the images are ranked according to their distance between the two circles to the reference image. It is obvious that sorting based on a local feature such as DCD would not yield such results since all images are identical in terms of black and white colour proportions, as they only differ spatially.

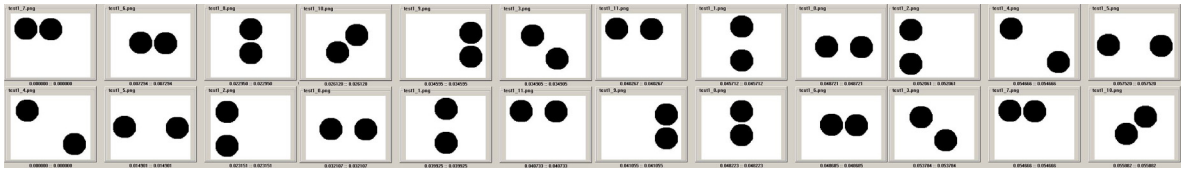

**Figure 5.1 - Results for similar objects with different distance layouts (first image is the reference and from left to right results)**

Results for the second image set are given in Figure 5.2. The top row shows results for the image with two coloured squares in a close proximity and vice versa for the bottom row. In this example the proximities of the regions are exploited. It can be seen that results are sorted in the order of proximity (distance) between the two colour squares even though their colours mostly do not match with the reference image. Therefore, the local colour feature for these squares does not contribute to the image similarity but region proximity does contribute due to the white background and the black square regions.

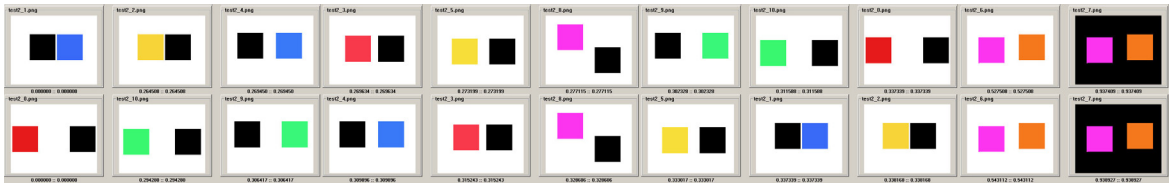

**Figure 5.2 - Results for similar scenery with white background and black square (first image is the reference and from left to right results)**

Secondly, tests were conducted in a synthetic database with 1119 images. The database includes a mixture of different coloured region arrangements and image dimensions. Figure 5.3 and Figure 5.4 show six queries and their results using the proposed framework with and without the region proximity feature in the synthetic database. Q1 and Q2 demonstrate the region proximity distance as a factor for the similarity measure among similar region compositions. Q1 illustrates the effect of similar region surroundings for a coloured square. The improved results comes from the fact that the black and white regions are perfectly matching in their local features (colour properties and region area), and this brings a higher contribution by the region proximity to the similarity score than the local and spatial similarity score for the two black circles. Without the region proximity feature, images with black circles appear earlier because both circles bring high similarity scores to the black square. Q2 demonstrates the region proximity distance among same regions with different spatial compositions. Here, query results are ranked by their distance between the two black circles. It can be seen that the region proximity improves the retrieval results compared to utilising just local features since only the spatial feature may provide the necessary discrimination among those images. Q3 and Q4 in the next figure illustrate the influence of the region proximity for region constellations. With considering only local features in Q3, images in the $10^{th}$ and $11^{th}$ rank are retrieved later whereas for the query using region proximity the location of the coloured centre region is taken better into consideration. In Q4 results, the same performance is observed as before. Note that the relevant retrievals with region proximity, especially the $2^{nd}$, $5^{th}$, and $6^{th}$ ranks are not retrieved within the first 11 ranks of the query using only local features. Moreover, $1^{st}$ and $6^{th}$ rank of the local feature query do not appear in the first 11 ranks for the query with region proximity due to their region constellations. Q5 just illustrates smaller influence of the region proximity feature if regions have same and similar local features and region constellations. It is illustrated that the $8^{th}$ and $9^{th}$ rank in the retrieval of the query with only local features are retrieved earlier in the query with region proximity.

Note that due the similar local features the retrieval improvement is rather small. The final Q6 retrieval demonstrates query results where the utilisation of region proximity degrades the retrieval performance, i.e. images in 4$^{th}$ and 9$^{th}$ rank are appear earlier based on perfect locally matching of all the red squares from the query to database image and perfect spatially matching from database image to query due to symmetric similarity approach and according to the introduced approaches and formulae for region and image similarities. Note especially the missing image of 60x60 dimensions for the region proximity query. The reason for this is the block-based region approximation. The superimposed 32x32 block grid leads to a block size of 1x1 pixels for this image since the smallest multiple of 32 in 60 is 32. Hence, 28 pixels are discarded, 14 on each side, which leads to an imprecise region representation. In this particular example, the aforementioned assumption that discarding pixels around the image boundaries does not affect the overall similarity, fails due the extreme image dimensions.

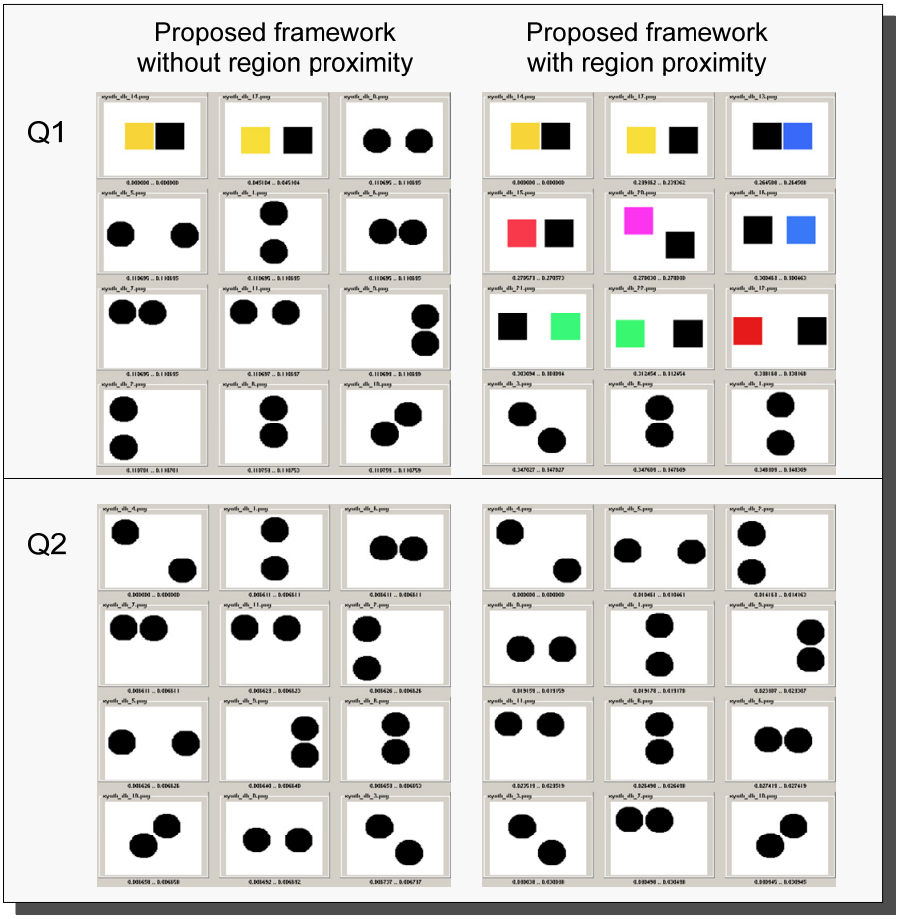


**Figure 5.3 - Two queries in synthetic database via proposed framework module. Top-left image is the query**

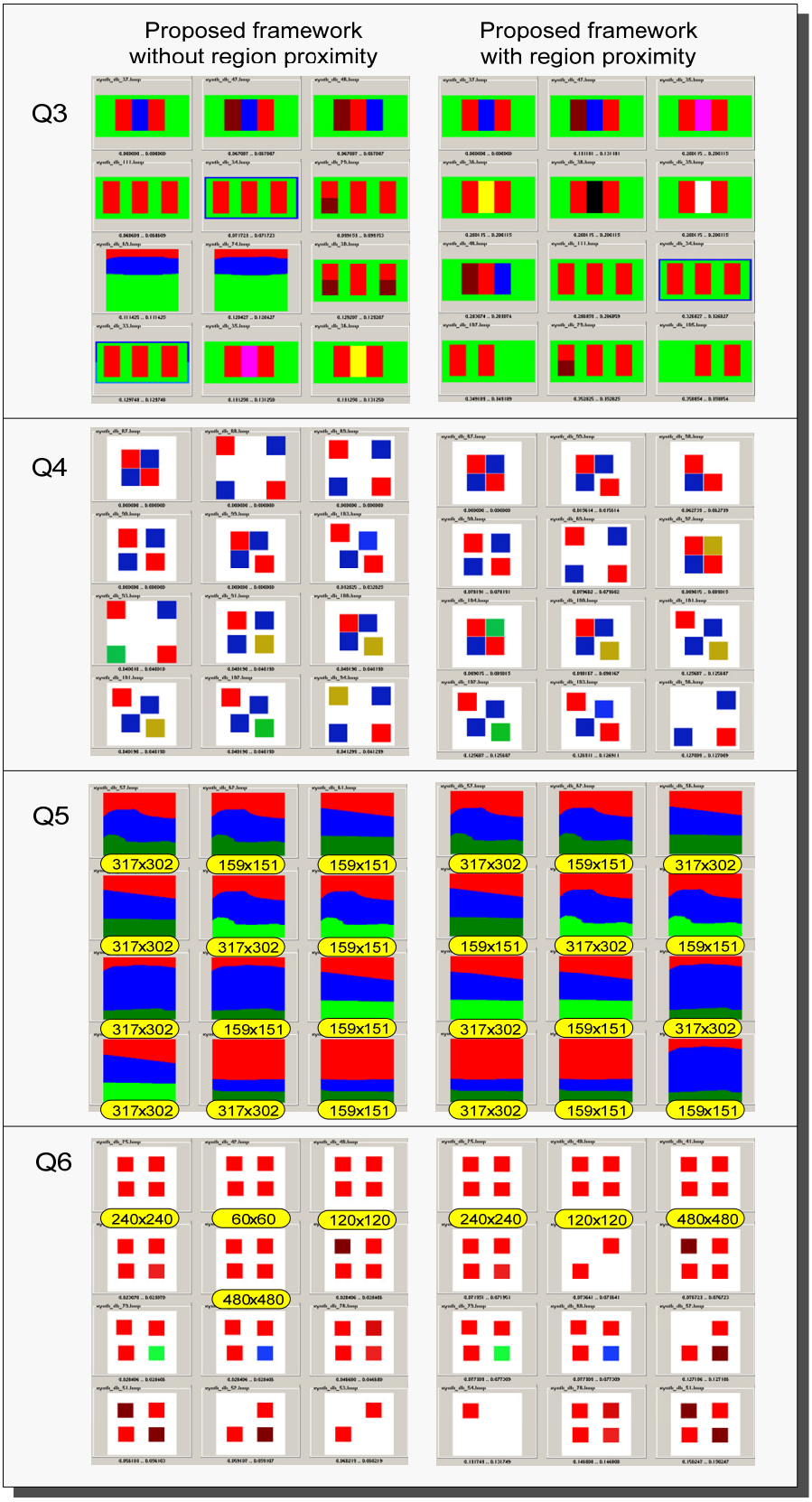


**Figure 5.4 - Four queries in synthetic database via proposed framework module R2. Top-left image is the query. Some dimensions are tagged in yellow boxes.**

## 5.2 Retrieval Results on Natural Databases

Most image collections contain natural, real-life images. For evaluating the retrieval on such images, three databases are created from the Corel image collection [15]. *Corel1K* includes 1000 images in 10 categories of *African natives*, *beaches*, *buildings*, *busses*, *dinosaurs*, *elephants*, *flowers*, *food*, *horses*, and *mountains*. *Corel10K* extends *Corel1K* to 10 000 images in 100 categories such as *vehicles*, *wildlife animals*, *sports*, *objects*, *peoples*, *textures*, etc. The third database, *Corel20K*, further expands the *Corel10K* database up to 20 000 images in 200 categories. These categories are similar to the *Corel10K* categories including a wider variation of image content. All images in these three databases have image dimensions of either 384-by-256 or 256-by-384 pixels. For evaluation, queries are selected in the following manner. For each of the 10 categories in *Corel1K*, 5 images are queried, all of which makes 50 queries in total. These selected images are then queried in all three databases *Corel1K*, *Corel10K*, and *Corel20K* and thus the influence of the database size and its content variation can be evaluated accordingly.

During indexing and retrieval operations in the sample Corel databases in this section the following parameter settings (Table 5.3) were utilised by the FeX module in which the prototype system is implemented. The parameters are set as trade-offs between speed and efficiency based on extensive testing with those general purpose images. The simC and simT parameter values are set in a manner to result in less and larger homogeneous regions, which yields in a better region representation and a more time-efficient retrieval process.

**Table 5.3 - FeX parameters for natural databases**

| maxS | var | cThr | simC | simT | maxC | minD | simD | sc | or |
|---|---|---|---|---|---|---|---|---|---|
| 25 | 125 | 0.25 | 0.6 | 0.05 | 8 | 15 | 45 | 3 | 4 |

The segmentation masks shown in Figure 5.5 are based on the *segmentation* and *grouping* phase in prototype system. The four columns from left to right show the original image, segmentation mask from mSPMG method, segmentation mask with *grouping* using simC as 0.5 and 0.6, respectively. For the different original images in Figure 5.5 (a) to (h), it can be seen that the applied segmentation scheme (*segmentation* + *grouping*) achieves respectable results. Furthermore, the regions (segments) for images (a)-(f) present a proper object representation especially for (b), (d), and (f) with minor differences between the

two different *grouping* thresholds. Masks for images (g) and (h) also display a proper region representation for a single image object. However, in corresponding grouping masks for image (g) the white ship is well segmented but sea and sky are grouped due to their low feature discrimination. The effect of different grouping thresholds can be clearly seen in corresponding masks for image (h). With simC=0.6 a proper object region is extracted but one might argue that the green-black background is under segmented. All masks so far have one characteristic in common that the initial segmentation method, mSPMG, produced an over segmented image, which can then be processed in the *grouping* stage. Masks for images (i) and (j) illustrated examples for the case if the segmentation method provides instead an under-segmentation scheme. In this case, *grouping* is not able to correct those faults, as expected. Based on the segmentation masks obtained, *grouping* returned acceptable results in general. The last masks for images (k) and (l) show where *grouping* process fails due to applied low-level features and thresholds, respectively. Since *segmentation* and *grouping* approaches are threshold based, it can be seen that what produces suitable results for some images might fail for others.

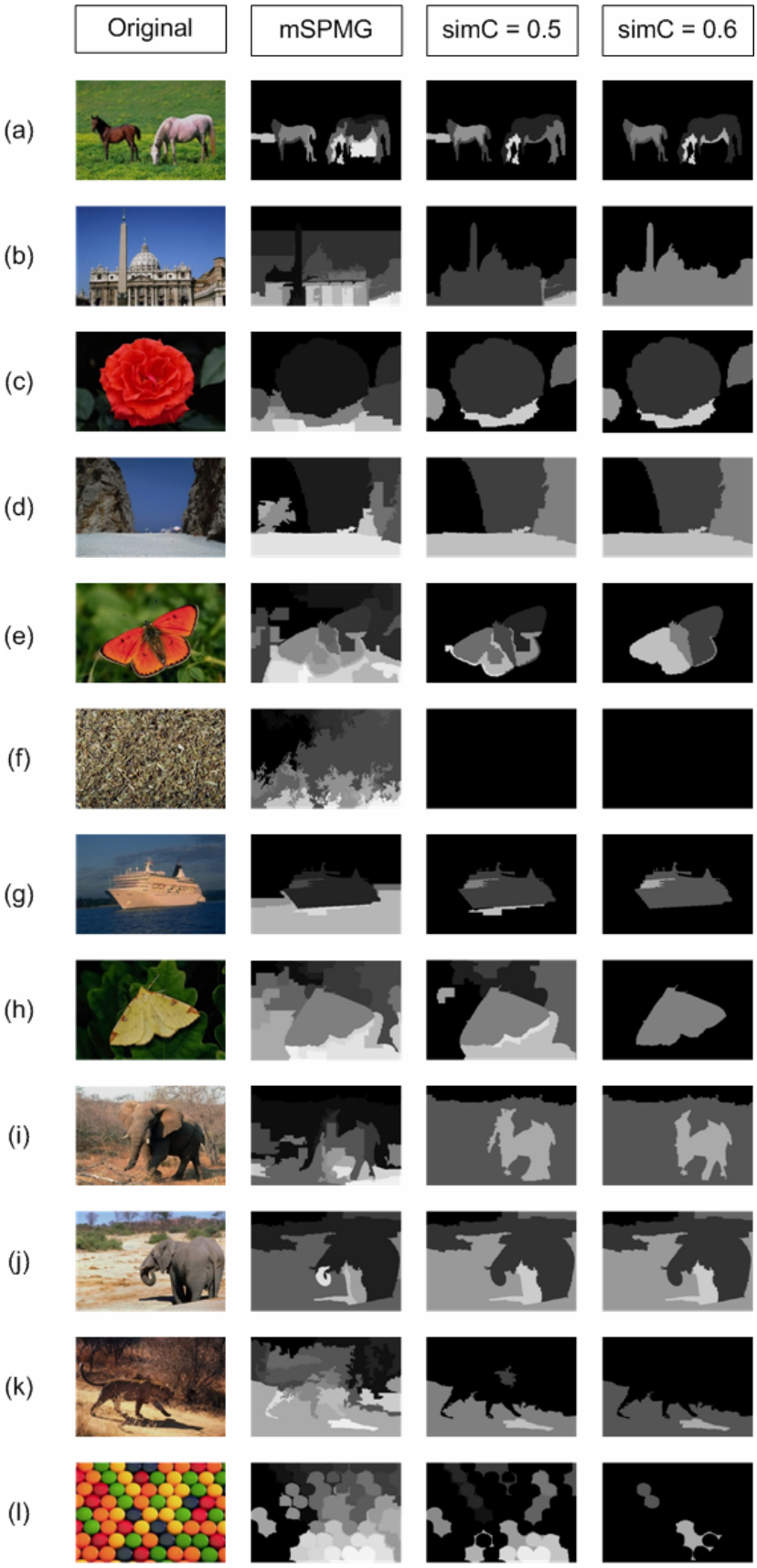


**Figure 5.5 - Segmentation and grouping examples**

The computational time needed for the feature extraction phase is discussed next. It can be divided into five parts *segmentation*, *grouping*, colour extraction, texture extraction, and region proximity computation to demonstrate the effect of each individual part. The Tables 5.4, 5.5, and 5.6 present the processing time for feature extraction spent for Figures 5.6, 5.7, and 5.8, respectively, where the image in Figure 5.6 is partially textured due to the blurred background, Figure 5.7 shows a pure textured image, and Figure 5.8 shows a non-textured image. The results from all three tables indicate that the extraction times for colour, texture, and region proximity feature extractions are constant due to the equal image dimensions. However, the texture extraction using Gabor-Wavelet-Transform is the slowest process whereas the time required for colour and region proximity extraction is almost negligible in our implementation. Moreover, the tables show that segmentation speed also depends on the degree of texture dominance in the image. If the image is heavily textured as in Figure 5.7, the segmentation time increases. The reason is that the heavy textural structure causes SPMG method to split the image down to the pixel level resulting in a massive amount of tiny segments, which are then further processed in the post-processing step. Based on this outcome, the speed of *grouping* depends on the number of regions and their features. The more regions with similar feature representations, the more grouping will be applied, which will eventually take more time.

**Figure 5.6 - Horse image**

**Table 5.4 - Timing for horse image**

| Step in the method | Time to complete |
|---|---|
| Segmentation | 7*s* 159*ms* |
| Grouping | 2*s* 955*ms* |
| Colour Extraction | 0*s* 296*ms* |
| Texture Extraction | 4*s* 908*ms* |
| Region Proximity | 0*s* 031*ms* |
| *Total FeX operation time* | 15*s* 349*ms* |

**Figure 5.7 - Pure texture image**

**Table 5.5 - Timing for pure texture image**

| Step in the method | Time to complete |
|---|---|
| Segmentation | 72*s* 447*ms* |
| Grouping | 5*s* 251*ms* |
| Colour Extraction | 0*s* 281*ms* |
| Texture Extraction | 4*s* 705*ms* |
| Region Proximity | 0*s* 016*ms* |
| *Total FeX operation time* | 82*s* 700*ms* |

**Figure 5.8 - Fighter image**

**Table 5.6 - Timing for fighter image**

| **Step in the method** | **Time to complete** |
|---|---|
| Segmentation | 2*s* 907*ms* |
| Grouping | 2*s* 142*ms* |
| Colour Extraction | 0*s* 234*ms* |
| Texture Extraction | 4*s* 940*ms* |
| Region Proximity | 0*s* 031*ms* |
| *Total FeX operation time* | 10*s* 254*ms* |

Retrieval is processed by MUVIS's QBE-scheme and the performance is measured by ANMRR. The retrieval results are further evaluated subjectively over the pre-defined category classification in *Corel10K* and *Corel20K*. There are two motivations for this type of evaluation. Firstly, some categories contain images with similar semantic content which cannot be described discriminated enough by low-level features. Hence, it is difficult to relate them to their category only based on their low-level feature similarities. Secondly, similar semantic content represented by similar low-level descriptions exists in those databases due to their pre-defined category classification. For example, say a query image contains a mountain. Now, there might be similar images displaying mountains in another class such as *roads* or *sports*. Moreover, *food* images are separated into classes such as *desserts*, *dinner*, *baking*, etc.; the semantic meaning of the content is impossible to describe by low-level features so that *food* is considered as a general class here. Therefore, images retrieved from categories with such similar content are considered as ground-truth.
The three databases are indexed with three MUVIS (frame-based) FeX modules: a combination of DCD and Gabor texture (C+T), the prototype system with *grouping* enabled for homogeneous regions (R1) where *grouping* parameters simC and simT are set

to 0.1 and 0.05, respectively, and the prototype system with *grouping* enabled for object-based regions (R2) where grouping parameters are set as in Table 5.3. Due to the lack of access to regionalised CBIR systems, only one frame-based method (module) is used here for comparison purposes, which employs the same feature descriptors as the prototype system.

The ANMRR results for all three FeX modules per sample database are presented in Table 5.7. Generally, it can be seen that performance degrades in larger databases independent from the approach and yet both region-based methods perform slightly better than the frame-based method for all three databases. Furthermore, note that module R1 achieves best results for *Corel10K*, R2 achieves best results for *Corel1K*, and the performance difference among the three modules in *Corel20K* is slightly better for both region-based approaches. Moreover, the performance for module C+T in *Corel20K* does not degrade as much as for modules R1 and R2.

**Table 5.7 - ANMRR results for natural databases**

| | C+T | R1 | R2 |
|---|---|---|---|
| Corel1K | 0,2054 | 0,1634 | 0,1435 |
| Corel10K | 0,4733 | 0,4189 | 0,4350 |
| Corel20K | 0,5150 | 0,5009 | 0,5002 |

Considering the fact that an ANMRR value smaller than 0,3 indicates acceptable accuracy for the retrieval and larger than 0,7 presents unsatisfactory or almost useless retrieval performance, the influence of the different database sizes is recognisable. Since the overall ANMRR result is obtained over a variety of 50 queries, these results represent a mixture of all classes with different content and features so that an examination of the NMRR values for each category per module and database will provide a more detailed (content-based) evaluation of the different categories. Thus, Figure 5.9 shows the NMRR diagram for *Corel1K* where y-axis represents the NMRR values and x-axis the image categories with a representative image example. Generally, it can be seen that the different modules have different results for different categories even though modules R1 and R2 represent the same proposed framework approach. There are classes where all three modules work equally well (*dinosaur, horses*) or equally poor (*mountains*). Furthermore, modules R1 and R2 outperform C+T for categories *African natives* and *elephants* whereas R2 performs better on categories *beach* and *busses* than R1. Moreover, the frame-based module C+T performs better for *flowers* than the region-based modules. There are various reasons for

such outcomes. Module C+T performs worst on *African natives* and *elephants* because their low-level features are not discriminative enough to the content of other categories. Moreover, all three modules fail for *mountains* category due to the variety of different low-level descriptions for the semantic content mountains. The reason for R2 performing better than R1 for *beach* category is mainly due to segmentation. R1 is set up to group for homogeneous regions, which eventually result in larger number of regions as for grouping in R2. This means several smaller regions do not have the same influence on similarity than a few larger regions due to the region area weight utilised in the region matching scheme. However, this also depends on the content of the query image because it can be seen from Figure 5.9 that sometimes more regions may improve matching and, the retrieval performance for the categories such as *buildings* and *food*. The high retrieval performance of C+T for *flowers* content is probably due to the frame-based texture since texture feature can describe spatial pixel distributions, edges and contours, so it may have a significant influence especially in the frame-based texture extraction of such flower images. Due to the content of most of these flower images -light-coloured flower with dark-coloured background- strong edges are present and may be dominant in the texture description whereas the regionalised texture description looses such edge information.

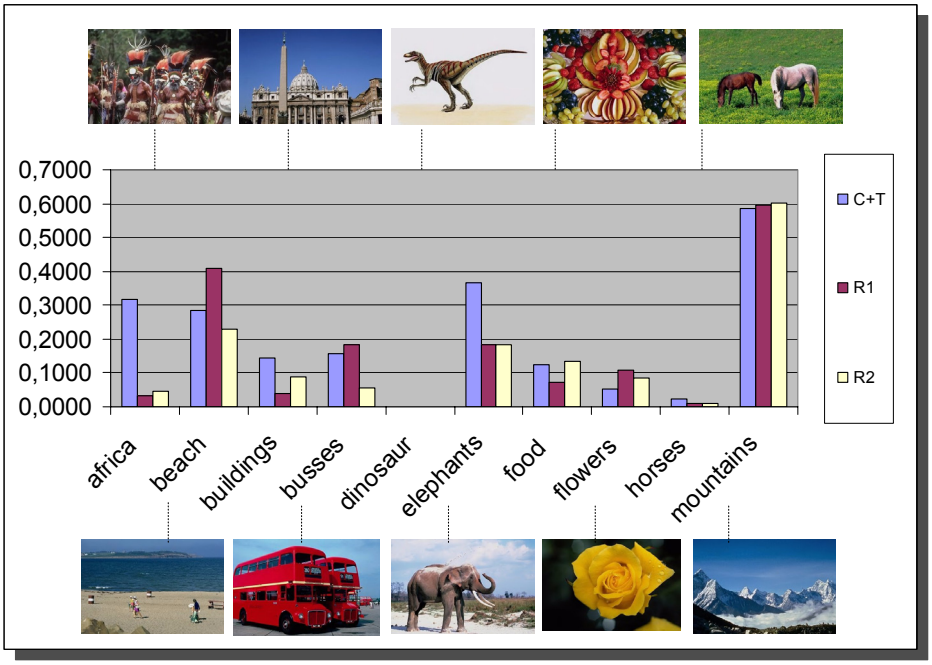


**Figure 5.9 - NMRR results for categories in Corel1k (NMRR values on the y-axis, the image categories on the x-axis)**

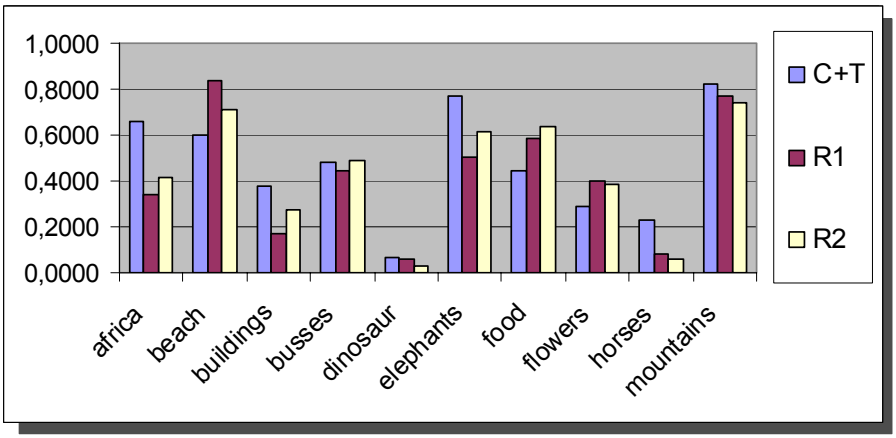


**Figure 5.10 - NMRR results for categories in *Corel10k* (NMRR values on the y-axis, the image categories on the x-axis)**

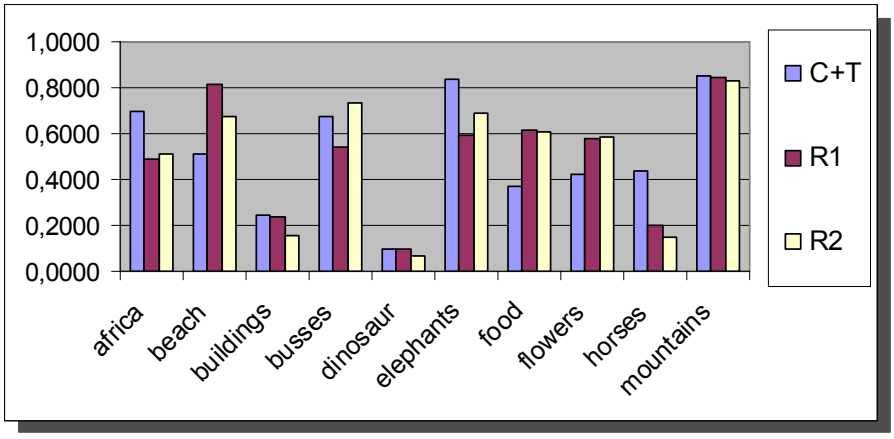


**Figure 5.11 - NMRR results for categories in *Corel20k* (NMRR values on the y-axis, the image categories on the x-axis)**

After evaluating the performance for *Corel1K*, it will be interesting to see the influence of the database size on individual category retrieval performance where Figure 5.10 and Figure 5.11 display NMRR results of these categories for *Corel10K* and *Corel20K*, respectively. Both figures represent the NMRR values on the y-axis and the image categories on the x-axis. It can be observed that, individual category performance is similar to the observations for the total ANMRR values and for both NMRR plots look similar since both databases contain same/similar images and the same images are used as queries for retrieval. An observation for both databases as for *Corel1K* is that *dinosaur* and *horses* categories perform the best and the *mountain* class performs the worst for all three modules. These performances can be explained by the image content because all *dinosaur* images picture one drawn dinosaur on big white background and mostly all of the *horses* images show one, two or more horses on a big green background (field, grass,

trees). This means for these two particular classes, their image content is discriminated by similar feature descriptions to other classes due to the rather large unique backgrounds. The main reason for the poor performance in the *mountain* class is that all images show different types of mountains and correlation between those images is highly semantic. Therefore, the basic assumption of low-level features related to underlying content description fails. This also means that the rest of the categories perform rather poor due to semantic issues. Furthermore, it is interesting to see that module C+T performs best for the *food* and *flowers* categories in both larger databases. The images of the *food* categories include many different colours, which result in a mixture of all these colours for module C+T. Furthermore, due to the content of those images, there are a lot of small objects producing a lot of edge information, which can be well described by texture. But the major reason for this is the fact that such images probably have unreliable under segmentation results resulting in poor descriptions for the region-based methods. This eventually generates poor feature descriptions for those regions due to the mixture of colour and texture features. Thus, during retrieval of the region-based modules these features have a low discrimination. For the better performance of module C+T for the *flowers* class in *Corel10K* and *Corel20K* apply the same reasons as for *Corel1K*. It is also interesting to notice that the buildings class in *Corel20K* performs better than in *Corel10K* for modules C+T and R2, and slightly degrades for module R1. This is due to the larger category variation in *Corel20K*, which includes several different *buildings* categories. The better performance of modules C+T and R2 compared to module R1 are related to coarser segmentation for and strong edge information between sky and building for R2 and C+T, respectively.

Figure 5.12 shows the first page of MUVIS *MBrowser* query results for four different queries in *Corel20K*. For the *building* and *horse* query, relevant images are retrieved. However, for the elephant and the mountain query, the retrieval performance degrades while returning only 2-3 relevant images within the first 11 ranks. The low performance of the elephant retrieval is due to the low discrimination of the region-based features. Nevertheless, note that 10 out of 11 images show a bluish-white part of similar size in the upper part of the image. The mountain query performs poorly due to the same reason as for *Corel1K*. In this particular query the image content is represented by bluish-whitish colours, which are also presented in other images; however, representing different semantic content.

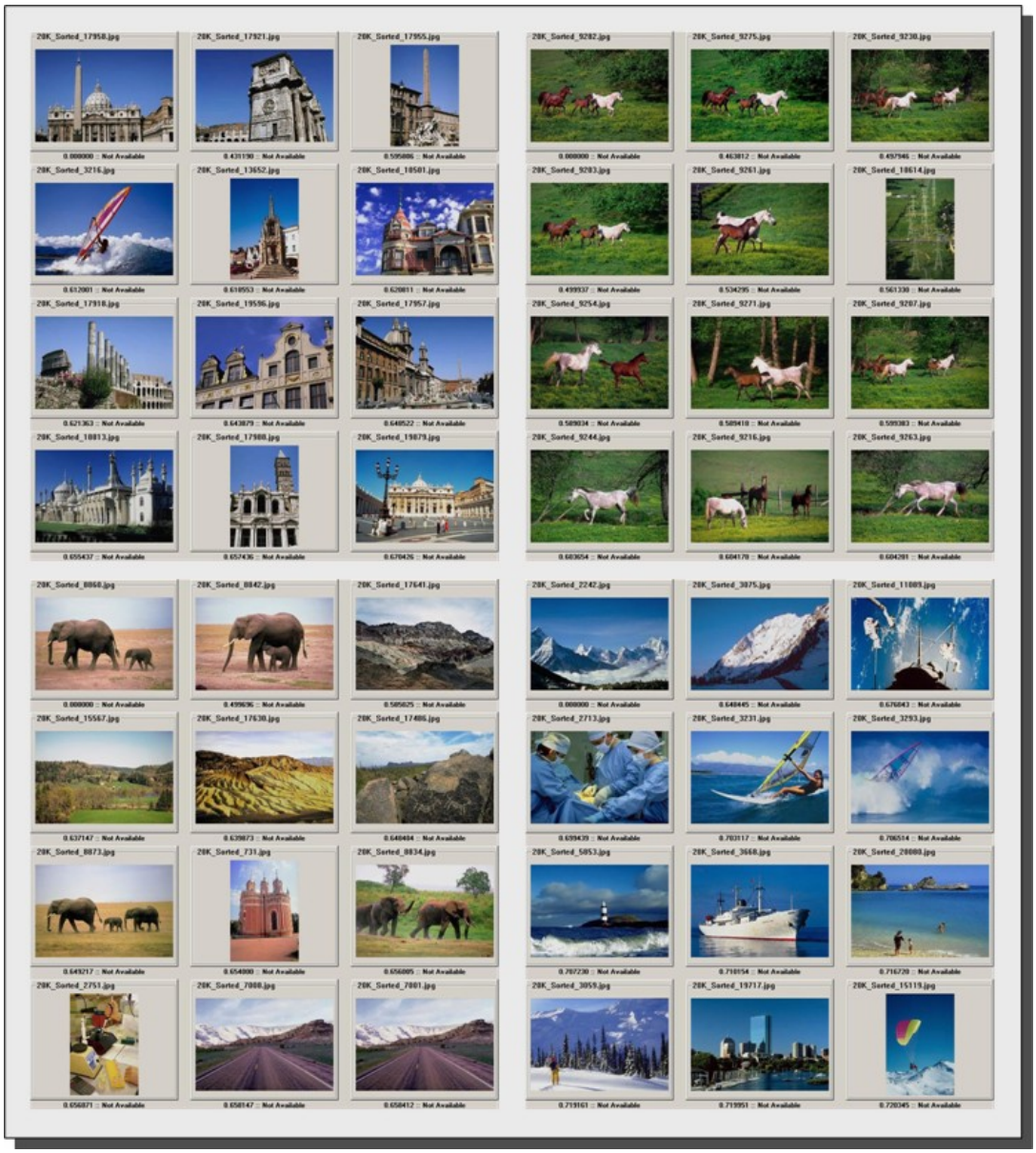

**Figure 5.12 - Four queries in *Corel20K* via module R2. Top-left image is the query**

The ANMRR values in Table 5.7, and NMRR values in Figure 5.9, Figure 5.10, and Figure 5.11 show that, generally, there is no major performance gain visible among the three applied modules with the exceptions for categories such as *dinosaur* and *horses*. Hence, subjective evaluation can further be focused on the actual image content with respect to human visual perception. For instance imagine a particular query image with two horses, one brown and one white, on a green field. Subjective evaluation considers mainly general semantic meaning, e.g. as long as there are horses in a green environment (i.e. green field with grass and trees) they were considered relevant. But how successful would one consider a retrieval showing only “brown horses on a green field” on the higher ranks and images with the actual content, “a brown and a white horse on a green field” are

retrieved in lower ranks? Based on this definition, subjective evaluation is further divided into semantic evaluation and category content evaluation. The former reflects the evaluation criterion based on human scene interpretation as described by the example given above. The latter one just considers the semantic classification of a category such as *beach*, *flowers*, *horses*, *buildings*, etc. Here, certain object or semantic details are not taken into account.

This extended evaluation definitions are tested on *Corel10K* and *Corel20K* using the FeX-module R2. The idea is to demonstrate by four query examples that if similar content is represented by similar regions the proposed spatial feature region proximity may enhance retrieval results. For those four queries, the first 23 ranks (two pages) of MUVIS *MBrowser* are considered and compared between the two different subjective evaluation measures. In the next upcoming figures representing retrieval results the top row always shows retrieval results for the proposed FeX module without applying region proximity (module R2 minus region proximity – R2-RP) and the bottom row includes region proximity (module R2), if not explicitly stated otherwise.

The first query example is an image showing one brown and one white horse on a green field. Table 5.8 represents the retrieval results, which are interpreted in the following way. For R2-RP in *Corel10K* it means there are 6 (relevant by semantic evaluation) out of 9 (relevant by category content evaluation) on the first page (11 items) and 13 out of 18 on the first two pages (23 items). For this particular case, the table shows that for module R2 on the first and second page more relevant images are retrieved representing better similarity to the query example in form of one brown and one white horse on a green field. Note further that R2 retrieves same number of images obtaining same retrieval results for *Corel10K* and *Corel20K*. This means that the features, describing the image content, are discriminative enough throughout the entire image databases. Figure 5.13 shows the first 12 retrieval results for this query in *Corel20K* where the left image is the query and the following are the first 11 best results.

**Table 5.8 - Extended subjective evaluation results for the query in Figure 5.13.**
**Numerals format: No. of relevant images by semantic evaluation (No. of relevant images by category content evaluation) in first page / second page of MUVIS *MBrowser***

| | R2-RP | R2 |
|---|---|---|
| **Corel10k** | 6 (9) / 13 (18) | 10 (11) / 16 (19) |
| **Corel20k** | 7 (11) / 13 (19) | 10 (11) / 16 (19) |

**Figure 5.13 - Query results excluding (top) and including (bottom) region proximity for brown and white horse example in Corel20K (first image is the query and from left to right best results)**

The second query example is an image with a pair of martial art fighters, which are referred to as fighters in the rest of the text. The class contains images with one or few fighters, which are dressed in black and stand in front of a white background as in Figure 5.13. Thus, the two query images contain two fighters, one with both in a close proximity (Figure 5.13 top-left image) and another with both in a farther proximity (Figure 5.8). Retrieval results for the two fighters with close proximity are shown in Table 5.9 and for the ones with farther proximity are presented in Table 5.10. From the results in Table 5.9 can be seen that both modules R2-RP and R2 obtain similar retrieval results for *Corel10K* and *Corel20K*. The reason is that segmentation successfully extracts both fighters from the white background so that the overall local and spatial region features yield a high similarity among those fighter images. In Table 5.10, the influence of region proximity is clearly visible when compared to the results in Table 5.9. The first page returns the same results but the second page brings a significant difference since R2-RP retrieves almost no relevant items based on semantic evaluation in *Corel10K* and *Corel20K*, respectively. On the other side, querying module R2 returns a second page full of relevant results. It should be noted that the one irrelevant item retrieved on the first page for R2-RP in *Corel20K* is a black and white painting with black borders in the top and bottom. Even though local region features yield a high similarity, this retrieval deficiency can be corrected by the region proximity feature. For visual evaluation, some query snapshots are illustrated in Figure 5.14, Figure 5.15, and Figure 5.16. Figure 5.14 shows the retrieval for querying the two fighters with close proximity in *Corel10K*. A figure representing image results for *Corel20K* is excluded due to the same content as in Figure 5.14. In Figure 5.15 and Figure 5.16, the difference between R2-RP and R2 is clearly visible. Both figures show only the second retrieval page (i.e. ranks 13 to 24) since the first page returns comparable results as in Table 5.10. The query image is shown in Figure 5.8. Note that the retrieved images in *Corel10K* and *Corel20K* and their ranks are identical for R2 and that R2-RP captures only single fighters and irrelevant images. In this example the difference occurs from

segmentation because the region of the left fighter is divided into three parts: upper body, belt, and lower body. Due to this, region proximity makes the essential difference between those three body parts and the second fighter.

**Table 5.9 - Extended subjective evaluation results for the query in Figure 5.14. Numerals format: No. of relevant images by semantic evaluation (No. of relevant images by category content evaluation) in first page / second page of MUVIS *MBrowser***

| | R2-RP | R2 |
|---|---|---|
| **Corel10k** | 10 (11) / 15 (23) | 10 (11) / 16 (23) |
| **Corel20k** | 10 (11) / 15 (23) | 10 (11) / 16 (23) |

**Table 5.10 - Extended subjective evaluation results for the queries in Figure 5.15 and Figure 5.16. Numerals format: No. of relevant images by semantic evaluation (No. of relevant images by category content evaluation) in first page / second page of MUVIS *MBrowser***

| | R2-RP | R2 |
|---|---|---|
| **Corel10k** | 10 (11) / 11 (18) | 10 (11) / 22 (23) |
| **Corel20k** | 10 (10) / 11 (18) | 10 (11) / 22 (23) |

**Figure 5.14 - Query results excluding (top) and including (bottom) region proximity for fighters with close proximity in Corel20K (first image is the query and results are ranked from left to right, top to bottom)**

**Figure 5.15 - Query results (2nd page) excluding (top) and including (bottom) region proximity for fighters with farther proximity in Corel10K (query is Figure 5.8 and results are ranked from left to right, top to bottom)**

**Figure 5.16 - Query results (2nd page) for fighters with farther proximity in Corel20K (query is Figure 5.8 and results are ranked from left to right, top to bottom)**

The third query example is a white flower (rose) in a dark (blackish) background as shown in Figure 5.17 (top-left image). A more abstract description of this image would be there is a light object in the centre of a dark environment. It is indented to demonstrate two facts here. Firstly, the influence of region proximity describing region surroundings is illustrated. Secondly, the basic assumption of low-level features related to underlying content description might fail (due to semantic gap). Therefore, Figure 5.17 and Figure 5.18 present the first two pages of retrieved items (top row first page, bottom row second page) in *Corel20K* for modules R2-RP and R2, respectively. Based on the semantic evaluation on the content, both obtain a similar overall performance retrieving 9 relevant images where 8 of R2-RP relevant retrievals are based on query region-based features and one is based on region proximity feature. R2 retrieves 5 and 4 relevant images, respectively. By leaving the semantic content out for a moment whilst considering the more abstract description presented earlier, it can be seen that the region proximity still provides a proper description and acceptable retrieval because most of the images in Figure 5.18 display an object (single region) in a dark environment except images in the 13$^{th}$, and 18$^{th}$ rank. In Figure 5.17, there are 5 (11$^{th}$, 17$^{th}$, 18$^{th}$, 21$^{st}$, and 22$^{nd}$ rank) images without representing the abstract description.

**Figure 5.17 - Query results first two pages of white flower example in Corel20K without region proximity (query is top-left image and results are ranked from left to right, top to bottom)**

**Figure 5.18 - Query results first two pages of white flower example in Corel20K with region proximity (query is top-left image and results are ranked from left to right, top to bottom)**

The final query example examines the effect of different segmentation of similar content on the retrieval results and the advantages and drawbacks of the region proximity feature. The query images are shown in Figure 4.17 with their corresponding segmentation results where the images labelled as *query* and *target1* are hereby used as *example1* and *example2*, respectively. It can be seen from Figure 4.17 that both images show identical

content with minor differences of the foal's head and tail. But, this minor content difference results in a different segmentation due to a blackish region of the larger shade besides the foal's body. Both example images are queried in *Corel20K* and retrieval results are used to demonstrate the effect of segmentation over the retrieval accuracy.

The Figure 5.19 and Figure 5.20 present the first 12 retrieval results without (top row) and with (bottom row) region proximity for *example1* and *example2*, respectively. An observation for these queries is that an insignificant change in the segmentation may result a significant degradation over the retrieval performance, probably due to the proposed region matching scheme. Furthermore, in both figures 5.17 and 5.18 is shown that in large databases different content is described with low-level features with limited discrimination power, which lead to mismatches due to the semantic gap (e.g. third image top row in both figures). As stated and shown earlier, region proximity might improve the retrieval of region-based mismatches, which can be seen by comparing top and bottom rows. However, both examples return new mismatches when using region proximity such as the *hawk* image in the 9$^{th}$ rank and the *rhinoceros* image in the 11$^{th}$ rank in bottom row of Figure 5.19 or the *food* image in the 4$^{th}$ rank and the *house* image in the 6$^{th}$ rank in bottom row of Figure 5.20. For such images, the general region proximity assumption fails, i.e. if regions have similar surrounding regions they may have similar content. All of these four mismatches include brownish regions as in the query image but due to the well-known gap between the semantic content and its low-level feature representation it is impossible to know if these brownish regions represent "a horse" or something else in such particular queries.

**Figure 5.19 - Query results excluding (top) and including (bottom) region proximity for brown horses (*example1*) in Corel20K (first image is the query and results are ranked from left to right, top to bottom)**

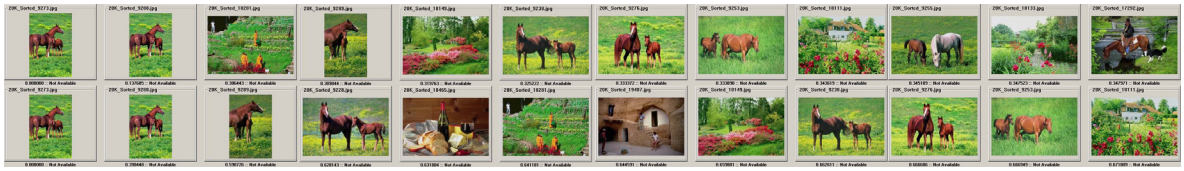

**Figure 5.20 - Query results excluding (top) and including (bottom) region proximity for brown horses (*example2*) in Corel20K (first image is the query and results are ranked from left to right, top to bottom)**

In summary, performance results on natural image databases can be seen from different perspectives. Firstly, they show that region-based feature extraction improves retrieval performance compared to frame-based extraction. Secondly, based on the region-based extraction scheme, it is shown that extraction for regions closer to a meaningful object representation enhances the overall retrieval performance compared to homogeneous regions. But bear in mind that this may depend on whether an image content is divisible into meaningful (object-based) regions or homogeneous regions. Based on an enhanced segmentation, performance improvements will result from the region proximity feature if regions represent meaningful objects as demonstrated by the two martial arts fighters with farther proximity.

## 5.3 Effects of Regions on Texture Description

Due to the observed degradation of texture description over regions, experiments have been conducted to evaluate the performance of several texture descriptors over arbitrary-shaped regions by means of the proposed framework in MUVIS. In order to accomplish this, six modules have been implemented: frame- and region-based versions of a 3-scale Haar-Wavelet, 3 scales and 4 orientations Gabor filter, and LBP texture descriptors. Wavelet and Gabor have been chosen because of their extensive use in texture retrieval and LBP as an alternative due to low complexity, good performance, and its easy applicability to regions. Wavelet and Gabor features are compared by Euclidean distance and LBP features by the G-Statistic.

For testing purposes, the features aforementioned earlier are extracted for the three sample databases generated and indexed using synthetic Brodatz textures [7] and natural images from Corel. The first database (DB1) contains 1760 images of size 160x160 pixels (110 classes with 16 images) from Brodatz texture album. It was generated by cutting the original Brodatz images into 16 equally sized sub-images. This database is indexed by frame-based feature extraction modules. The second database (DB2) contains 2370 images where each image represents an arbitrary-shaped textured region on a uniform background. Textures are taken from Brodatz collection where 79 of 110 textures are used. The rest is sorted out due to strong non-homogeneity and lack of proper representation of the overall texture in an arbitrary-shape region. DB2 is generated in the following way. At first 30 region masks are created and divided into 2 classes. On one hand there are geometric shapes such as rectangles and circles and on the other hand some

arbitrary shapes. Each class contains a shape in three different sizes: 75%, 50%, and 25% of the image area. Then, for each mask a patch is cut out from the original texture image and is overlaid with this specific mask. This guarantees each region has the same overall texture with a little variation. For the sake of simplicity the region masks are placed in the centre of the newly generated image. DB2 is indexed only by region-based FeX modules. The third database (DB3) contains 1000 natural images from Corel classified into 10 classes. It contains classes such as beach, flowers, busses, dinosaurs, horses, and etc. This database is indexed both by frame- and region-based (texture) FeX modules.

### 5.3.1 Evaluation of the Retrieval Performance

Retrieval experiments are carried out separately for each database and each texture feature. DB1 and DB2 are queried against ten texture classes chosen from Brodatz, namely D4, D9, D10, D15, D24, D37, D54, D68, D95, and D109. During the retrieval process five randomly selected items per class selected were queried. This test performs frame-based queries against a frame-based indexed database. For DB2 five regions with different shapes of size 25% are selected as shown in Figure 5.21. The shapes are named as *square, circle, triangle,* and two arbitrary shapes, *arb1 and arb2*. Each shape among the selected textures is used as a query and a region-based query is only performed over a region-based indexed database.

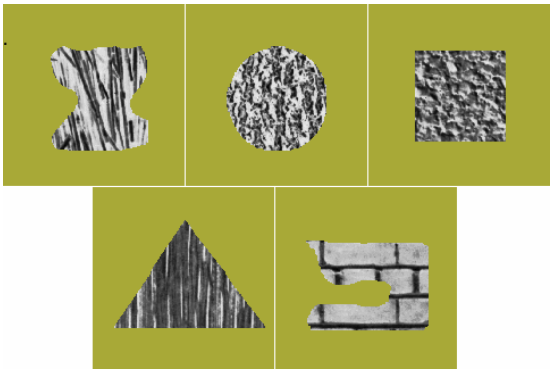

**Figure 5.21 - Arbitrary-shaped regions from top-left: arb1, circle, square, triangle, arb2**

Performance evaluations will be carried out in DB1 and DB2 separately as well as mutually. In DB3 five images randomly chosen per class were queried. The same images were used both for frame-based and region-based retrieval experiments.

The retrieval process is based on the query by example (QBE) scheme. For the frame-based case this simply means comparing the query features to features of each database

item using the aforementioned $L_p$-norms. The region-based cases are compared by the similarity introduced in 4.2.6 applying only texture feature comparison. For all retrieval tasks, performance is evaluated by using ANMRR. Moreover, for all experiments performed in this section *NG(q)* is fixed as 15.

### 5.3.2 Synthetic Texture Retrieval Results

Synthetic texture retrieval experiments were carried out on DB1 and DB2 where DB1 is the frame-based and DB2 is the region-based database. According to the ANMRR results presented in Table 5.11, LBP performs best on DB1 and is slightly better than Gabor whereas Wavelet is significantly worse. A similar observation can be made for DB2 where LBP performs far better than Gabor and Wavelet. Furthermore, it can be seen that in general for all three methods, the region-based retrievals produce the worst results.

**Table 5.11 - ANMRR results for synthetic texture database**

| Brodatz | Gabor | LBP | Wavelet |
|---|---|---|---|
| Frame | 0.1065 | 0.0849 | 0.4129 |
| Region | 0.5255 | 0.3607 | 0.7454 |

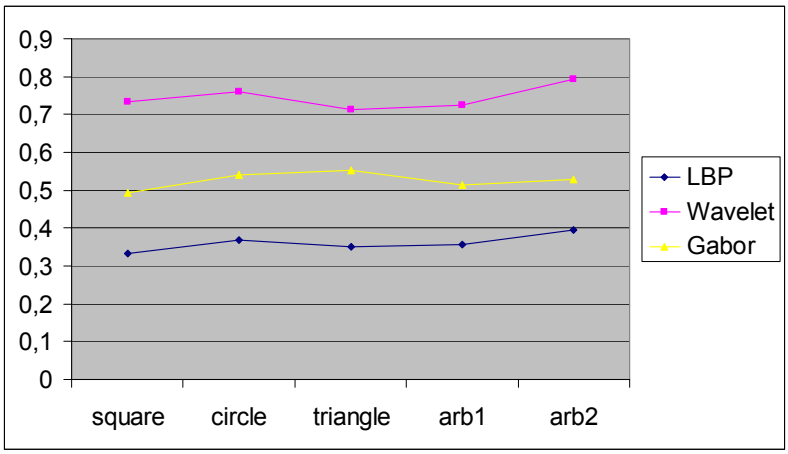


**Figure 5.22 - ANMRR results for different shapes for three descriptors**

Figure 5.22 displays the individual results for each shape where different shapes have different performances but none has a larger impact than the others. The y-axis represents the ANMRR value and the different shape types are presented in the x-axis. For LBP and Gabor the *square* shape performs best but surprisingly the triangular shape achieves best performance for Wavelet (yet the worst performance for Gabor). Furthermore, it can be seen that shape *arb1* performs well for any texture method; however arb2 results the worst

retrieval performance particularly for LBP and Wavelet. Note that all three methods vary within a NMRR range of 0.1, illustrating the insignificant effect of different shapes over the texture retrieval performance.

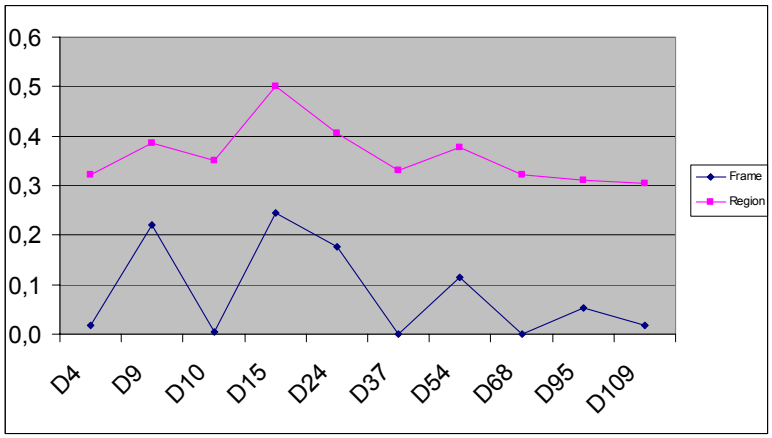


**Figure 5.23 - ANMRR plots for frame vs. region-based LBP descriptor**

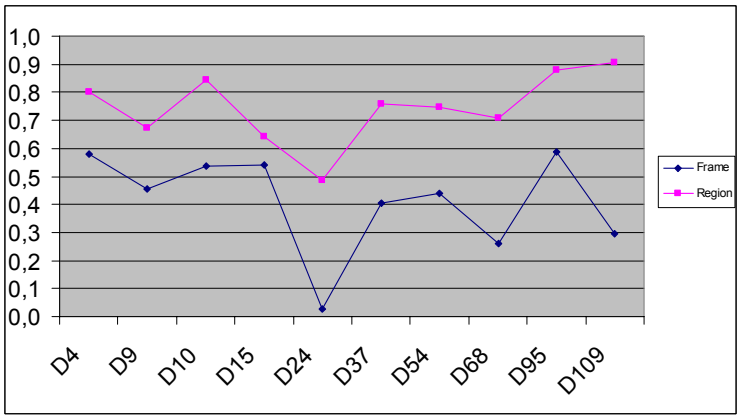


**Figure 5.24 - ANMRR plots for frame vs. region-based Wavelet descriptor**

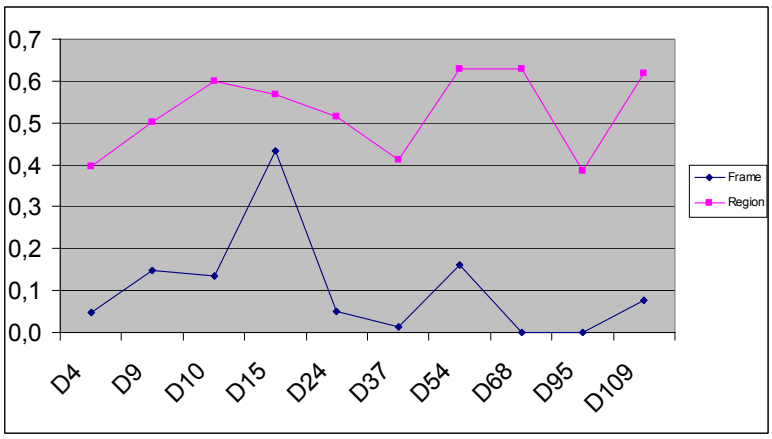


**Figure 5.25 - ANMRR plots for frame vs. region-based Gabor descriptor**

Figures 5.21, 5.22, and 5.23 present ANMRR results for frame- vs. region-based retrievals using LBP, Wavelet, and Gabor descriptors, respectively. In the figures the y-axis represents the ANMRR value and textures are represented in the x-axis. Various remarks can be made accordingly. First of all retrieval performance of the three texture descriptor can be compared directly. For instance LBP and Gabor can in general achieve far better performance than Wavelet since their worst (frame-based) ANMRR score is lower than the average performance of Wavelet. Furthermore, the Wavelet texture performance degrades the most when used in regions since LBP's worst ANMRR score is lower than the Wavelet's best value. For Gabor both plots are fairly different. Hence, this indicates that Gabor can have significant variations between frame- and region-based retrievals. Furthermore, these three plots show if a texture performs well on an entire frame, this does not necessarily mean that it will perform equally well on regions and vice versa. One can for instance see this in D68 and D15, D109, for Gabor and Wavelet, respectively.

The results mainly indicate that there is no particular shape over which any descriptor performs equally good or bad for any texture. It rather seems that certain shapes and certain textures have an influence on each other with respect to the texture descriptor employed as well. During conducting the retrieval experiments, the results seem to be size dependent. Most of the queries for LBP have almost perfect retrievals for shapes having the identical size (25%). Also other non-relevant items retrieved have the same region size than the query region even though the database includes images with the same texture per region but different region sizes. Wavelet and Gabor, on the other hand, retrieve several images of different region sizes even if the textures between the regions do not match. This indicates that region size seems to play an important role on all three methods, particularly on LBP.

### 5.3.3 Texture-based Retrieval Results for Natural Images

For the retrieval experiments over natural image database, DB3, both frame- and region-based feature extraction modules are performed for the three descriptors. Table 5.12 shows the ANMRR performance on natural images using frame- and region-based methods.

**Table 5.12 - ANMRR for Corel database**

| | Gabor | LBP | Wavelet |
|---|---|---|---|
| **Frame** | 0.2932 | 0.2524 | 0.5102 |
| **Region** | 0.6773 | 0.5918 | 0.6856 |

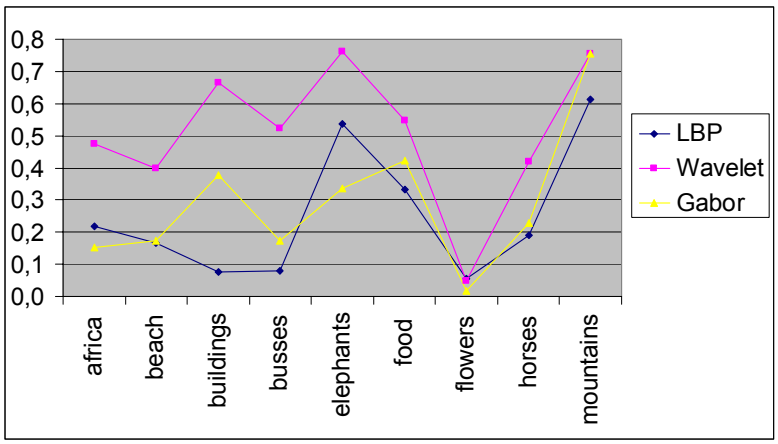


**Figure 5.26 - ANMRR plots for queries among different classes for three (frame-based) descriptors**

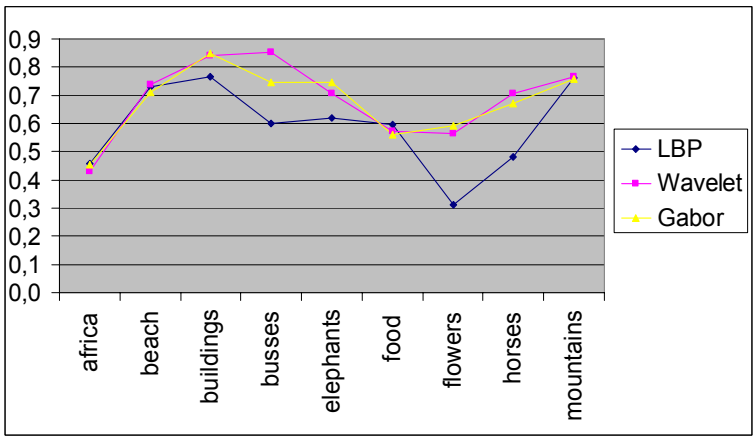


**Figure 5.27 - ANMRR plots for queries among different classes for three (region-based) descriptors**

Figure 5.26 presents similar results to the earlier tests with synthetic textures. Frame-based Wavelet's performance is significantly poorer than the other two. Therefore, the difference between frame and region for Wavelets is rather close. Further, it can also be noticed that all three region-based methods perform quite poor. Figure 5.26 and Figure 5.27 display the ANMRR plots of the three methods for frame- and region-based retrievals and Figure 5.29 displays two query results for frame-based and region-based retrieval of the three descriptors. It can be seen in Figure 5.26 that all three methods have different performances over the nine classes. Exceptions are flowers and mountains classes where performance is almost equal. Overall LBP and Gabor perform quite well on five to six classes out of nine. For region-based retrievals as shown in Figure 5.27 overall performance is similar between Wavelet and Gabor achieved close to equal results. LBP produces slightly better results for busses, flowers, and horses and similar results for the others. It can be seen in Figure 5.29 that all three region-based descriptors retrieve almost

the same images in different ranks. Note further that the classes such as *beach*, *buildings*, and *busses*, which perform reasonable well on the frame-based retrievals produce significantly worse results on regions. In the beach images there are some recognizable textures of sand and water. Due to the extraction over the entire frame the main contribution of the texture description comes from the edge between water, sand, and background. Region-based retrieval results suggest that without the presence of major edges, the internal textures in regions are not discriminate enough for retrieval. Similar observation can be made for the flower class, bride colour flowers such as yellow or red on a dark-greenish or blackish background result in strong edges. Here, it seems LBP descriptor is capable of describing those textures better than Wavelet and Gabor descriptors. The *building* class has also strong edges between the sky and buildings whereas on regions these objects cannot be described well enough by texture. The bus class emphasizes this even more. The main object in those images is a bus, which contains almost no texture at all. Therefore, a bus query results quite good retrieval on frame-based but poor performance on region-based retrieval. This is mainly to the bus shape with its contours over the entire image. A typical example, which further strengthens this claim, is presented in Figure 5.28. It basically shows the first two pages of retrieved images using frame-based Gabor descriptor where the query image is shown in top-left corner, i.e. “an antique building”. To be more specific 22 out of 23 retrieved items are busses since similar lines/contours are clearly present although the texture is not. This indicates two points. The first is frame-based texture extraction performs well due to main contours and boundary edge information. And the latter is that the texture extracted from regions seems to play an insignificant role in such natural images. Generally speaking, frame-based retrievals are much better than region-based almost for all classes even though texture information is homogenous (pure) in a region and one normally expects this the other way around. Hence, our conclusion for this surprising result is that frame-based texture descriptor further captures the major shape and boundary edge information whereas region-based methods lack. This favours images with strong boundary edges such as *beaches, buildings, flowers*, and *busses*.

**Figure 5.28 - Retrieval results for frame-based Gabor for an antique building query (top-left)**

**Figure 5.29 - Examples of frame vs. region-based retrieval with three descriptors, row 1: Gabor, row 2: LBP, row 3: Wavelet; column 1&3: frame-based, column 2&4: region-based retrieval.**

# 6 Conclusions and Future Work

Various regionalised CBIR systems and approaches have been developed during the past years all of which have a similar overall approach, i.e. first segmenting the image and then extracting low-level features over the obtained regions. However, none of these systems supports a generic, flexible, or exchangeable integration of segmentation methods and feature extraction approaches. This thesis presents such a regionalised CBIR framework approach applying a region-based feature extraction scheme including five main parts, four during indexing and one during retrieval.

The indexing phase of the framework incorporates *segmentation*, *grouping*, *local* and *spatial feature* as consecutive stages. *Segmentation* phase is designed as a black-box to provide the flexibility of applying any segmentation method. Due to this flexible design, a dedicated *grouping* phase is added to support any segmentation method by correcting possible over segmentation errors. Moreover, *grouping* may also yield to object-based segmentation for certain image categories and therefore, the segmentation method applied should be tuned to produce over segmentation rather than under segmentation errors since on the proposed framework has no means to correct them. Similar to the *segmentation* phase, the extraction of visual features from the regions is also considered as a black-box. This allows the integration of any visual descriptor in the framework. Furthermore, the proposed framework includes a spatial feature called region proximity, which describes a visual scenery through the local and spatial properties of its regions. The introduced concept of region-proximity is based on the naïve expectation that image and region similarities may be expressed based on similar region compositions in visual sceneries. We observed that degradations or deficiencies in segmentation usually yield in worse retrieval performances. Generally speaking since this framework can be seen as a chain of consecutive stages, the outcome will mainly rely on its weakest component. In the retrieval phase of the framework, the image similarity is based on a similarity maximisation approach for the individual region similarity by colour, texture, and region proximity features employed into a many-to-one region matching scheme.

The prototype implementation of the proposed framework utilises a simple colour segmentation based on region splitting by quad-tree. The initial approach is modified by a pre-processing step (bilateral filtering) and several post-processing steps to enhance the

segmentation results towards a better and possibly smaller region (over-segmentation) representation. The colour and texture features used in regions are DCD and Gabor filter, respectively. These descriptors are selected due to their efficiency and compactness. In the current implementation, no shape information is used since object extraction is not feasible and thus, meaningful shape description cannot be achieved. If it would be integrated it might be only reliable for certain content but would be unreliable for many others providing false matches. However, shape descriptors may be easily integrated in the future as long as this deficiency is successfully addressed. The region proximity feature applies a block-based approach due to its efficient and robust region representation where the image is superimposed by a 32-by-32 grid. Regions are described by grid blocks, which are used to calculate region distances by an average distance measure. This approach presents a more robust distance computation than the traditional Hausdorff distance. After the feature extraction, the descriptions are stored in an image feature vector, which represents an efficient and compact representation of the extracted region features. The feature vector is used in retrieval for feature-based similarity calculation. As a FeX framework requirement, the feature vector size needs to be known before hand, which might lead to a significant overhead for images segmented into a small number of regions.

During retrieval, the similarity of two images is calculated by their region similarity scores where each region seeks its best match based on colour, texture, and region proximity features. The final image similarity score is then the cumulative sum of all weighted maximised region similarities. By integrating this approach into a many-to-one region matching scheme, it is necessary to calculate the image similarity score for both of the two compared images. The total image similarity between two images is then the average over the two separated image similarities scores. This assures similarity symmetry between two images but also biases the total image similarity score what might lead to indistinguishable results for certain images.

Results for synthetic images demonstrate a promising performance of the region proximity feature without any segmentation faults and degraded discrimination of local region features. Results on natural image databases show that region-based feature extraction performs better than frame-based extraction and further illustrates the contribution of region proximity to the description and retrieval performance. Moreover, results illustrate that performance may vary depending on several factors such as the database size and its

content variation, the way of utilising the extracted features, and in the case of region-based feature extraction, the segmentation method applied. This means even though local and spatial features from regions are used; results indicate that due to semantic content and the effect of segmentation faults, performance might differ for various image categories. Segmentation can be seen as an essential factor in regionalised content-based description and retrieval. On the one hand, if segmentation may provide object-based results (synthetic image databases) or consistent results for similar content (certain categories in natural image databases), the proposed framework achieves superior performance compared to frame-based approaches. Furthermore, with such segmentation results spatial information among regions further enhances results towards human understanding whenever querying certain semantic objects or visual scenery constellations. On the other hand, if segmentation provides rather poor region representation, region-based extracted features and spatial information are degraded in their description power. Moreover, in such examples frame-based feature description may be more efficient in retrieval.

During the first experiments, we observed a low influence and degradation of region-based texture in retrieval where we expected better performance. Therefore, experiments with three texture descriptors, namely Wavelet, Gabor, and LBP, were conducted to compare their frame-based and region-based texture performance. Results for synthetic Brodatz textures demonstrated that region-based texture always performed worse that frame-based texture for all three employed methods. Thus, the region area and shape of regions cause certain degradations over the descriptor. However, the impact of the region shape is rather minimal for those textures. In most cases, the texture structure within the region and the applied descriptor method played an important role. Furthermore, region size seems to be critical. The retrieval results obtained in natural database of one thousand images approves that region-based extracted texture is degraded in its retrieval performance compared against the frame-based extracted texture. This is mainly due to the fact that strong edges over the entire image have the largest influence on the texture description. In this case, shape boundaries have a stronger effect then the single frame-based textures, especially if images contain dominant contours. In the case of region-based texture extraction these shape boundary information are lost and, moreover, for most of the natural image regions, texture information are not discriminate enough for a good retrieval. Therefore, region-based texture features seem to play a minor role in retrieval of natural images with small dimensions or low resolutions.

Since robustness of segmentation is essential over region-based retrieval performance, the future work will concentrate on its improvements. Moreover, current and planned research work for the proposed framework include: adapting grouping parameters based on content features, integrating orientation and direction information into region proximity feature, enhance the region matching to a many-to-many scheme, employing fuzzy approaches for region distance description and region matching for an enhanced visual perception model. Furthermore, semantic colour names might be integrated to improve the semantic description and retrieval. Evaluating the effects of different segmentation methods with different visual feature descriptors is also considered.

# References


[1] R. Adams and L. Bischof, "Seeded Region Growing", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641-647, 1994.

[2] I.Ahmad, S. Abdullah, S. Kiranyaz, M. Gabbouj, "Content-Based Image Retrieval on Mobile Devices", *In Proc. of SPIE (Multimedia on Mobile Devices)*, vol. 5684, San Jose, US, Jan. 2005.

[3] M.L.G Althouse, C. Chang, "Image segmentation by local entropy methods", *Proceedings International Conf. on Image Processing*, vol. 3, pp. 61-64, Oct. 1995.

[4] S. Ardizzoni, I. Bartolini, M. Patella, "Windsurf: region-based image retrieval using wavelets", *10th International. Workshop on Database and Expert Systems App.,* p. 167-173*,* 1999.

[5] D. H. Ballard, C. M. Brown, Computer Vision, Prentice-Hall, 1982.

[6] C. Böhm, S. Berchtold, and D. Keim, "Searching in High-Dimensional Spaces - Index Structures for Improving the Performance of Multimedia Databases", *ACM Computing Surveys*, vol. 33, pp. 322–373, 2001.

[7] P. Brodatz, Textures: A Photographic Album for Artists & Designers. Dover, 1966.

[8] M. Buckland, F. Gey, "The relationship between Recall and Precision", *Journal of the American Society for Information Science*, vol. 45, pp. 12-19, 1994.

[9] C. Carson, S. Belongle, H. Greenspan, and J. Mailk, "Blobworld: Image segmentation using expectation-maximization and its application to image querying", *IEEE Transactions on PAMI*, vol. 24, pp. 1026–1038, 2002.

[10] S. K. Chang , Q. Y. Shi , C. W. Yan, "Iconic indexing by 2-D strings", *IEEE Transactions on PAMI*, vol. 9, pp. 413-428, 1987.

[11] S. K. Chang, E. Jungert, and Y. Li, "Representation and retrieval of symbolic pictures using generalized 2D string", *Technical Report*, Uni. of Pittsburgh, 1988.

[12] E. Chavez, G. Navarro, R. Baeza-Yates, and J. Marroquin, "Searching in Metric Spaces", *ACM Computing Surveys*, vol. 33, pp. 273–321, 2001.

[13] F. A. Cheikh, “MUVIS: A System for Content-Based Image Retrieval”, PhD. Thesis at Tampere University of Technology, Tampere, Finland, Apr. 2004.

[14] J. J. Chou, “Voronoi diagrams for planar shapes”, *IEEE Computer Graphics and Applications*, vol.15, pp. 52-59, Mar. 1995.

[15] “Corel Clipart and Photos”, http://www.corel.com/products/clipartandphotos/

[16] G. Dantzig, Linear Programming and Extensions, Princeton University Press, 1963.

[17] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, “Peer Group Filtering and Perceptual Color Image Quantization”, *Proc. of IEEE Int. Symposium on Circuits and Systems*, vol. 4, pp. 21-24, 1999.

[18] Y.Deng, and B.S.Manjunath, “Unsupervised segmentation of color-texture regions in images and video”, *IEEE Transactions on PAMI*, vol. 23, pp. 800-810, Aug. 2001.

[19] Z. Dengsheng, L. Guojun, “Evaluation of Similarity Measurement for Image Retrieval”, *In Proceedings of the International Conference on Neural Networks and Signal Processing*, vol. 2, pp. 928-931, 2003.

[20] D. Depalov, T. N. Pappas, D. Li and B. Gandhi “Perceptually Based Techniques for Semantic Image Classification and Retrieval,” *Human Vision and Electronic Imaging, SPIE Conference*, San Jose, CA, Jan. 2006.

[21] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber, “Efficient and Effective Querying by Image Content”, *Journal of Intelligent Information Systems*, vol. 3, pp. 231-262, 1994.

[22] J. Fauqueur and N. Boujemaa, “Region-Based Image Retrieval: Fast Coarse Segmentation and Fine Color Description”, in *Proc. of IEEE Int. Conf. on Image Processing*, Rochester, USA, Sep. 2002.

[23] P. Felzenszwalb and D. Huttenlocher, “Efficient Graph-Based Image Segmentation”, *International Journal of Computer Vision*, vol. 59, pp. 167-181, Sep. 2004.

[24] M. Ferreira, S. Kiranyaz and M. Gabbouj, “A Novel Shape Descriptor over Multi-Scale Edge Field: 2D Walking Ant Histogram”, *Proc. of IWSSIP 2006*, pp. 475-378, Hungary, Sep. 2006.

[25] Flickr: http://www.flickr.com

[26] J. D. Foley et. al., Computer Graphics: Principles and Practice, 2nd Edition, Addision-Wesley, 1997.

[27] H. Freeman and A. Saghri, "Generalized chain codes for planar curves", In *Proceedings of the 4th International Joint Conference on Pattern Recognition*, pp. 701-703, Kyoto, Japan, Nov. 1978.

[28] D. Geman, S. Geman, C. Graffigne, pp. Dong, "Boundary detection by constrained optimization", *IEEE Transactions on PAMI*, vol. 12, pp. 609–628, 1990.

[29] T. Gevers and A. Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval", *IEEE Transactions on Image Processing.*, vol. 9, pp. 102-119, 2000.

[30] Y. Gong, C. H. Chuan, G. Xiaoyi, "Image Indexing and Retrieval Using Color Histograms", *Multimedia Tools and Applications*, vol. 2, pp. 133-156, 1996.

[31] V. N. Gudivada, and V. V. Raghavan, "Design and evaluation of algorithms for image retrieval by spatial similarity", *ACM Trans. on Information Systems*, vol. 13, pp. 115-144, Apr. 1995.

[32] O. Guldogan, E. Guldogan, S. Kiranyaz, K. Caglar, and M. Gabbouj, "Dynamic Integration of Explicit Feature Extraction Algorithms into MUVIS Framework", *Proc. of the 2003 Finnish Signal Processing Symposium, FINSIG'03*, pp. 120-123, Tampere, Finland, May 2003.

[33] R. Haralick, "Statistical and structural approaches to texture", *Proceedings of the IEEE,* vol. 67, pp. 786–804, 1979.

[34] F.L. Hitchcock, "The distribution of a product from several sources to numerous localities", *Journal of Mathematical Physics*, vol. 20, pp. 224–230, 1941.

[35] S. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm", *Journal of the ACM*, vol. 23, pp. 368-388, 1976.

[36] P. Howarth, S. Rüger, "Evaluation of Texture Features for Content-Based Image Retrieval", CIVR 2004, LNCS 3115, pp. 326–334, 2004.

[37] J. Huang; S.R. Kumar, M. Mitra, W. J. Zhu, R. Zabih, "Image indexing using color correlograms", *Proc. of Comp. Vision and Pattern Recog.*, pp. 762-768, Jun. 1997.

[38] D. Huttenlocher, G. Klauderman, W. Rucklidge, “Comparing images using the Hausdorff-distance”, *IEEE Transactions on PAMI*, vol. 15, pp. 850-863, 1993.

[39] G. Iannizzotto and L. Vita, “Fast and Accurate Edge Based Segmentation with no contour smoothing in 2D Real Images”, *IEEE Transactions on Image Processing*, vol. 9, pp. 1232-1237, Jul. 2000.

[40] IEEE, “IEEE standard glossary of image processing and pattern recognition terminology” *IEEE Std. 610.4-1990*, 1990.

[41] ISO/MPEG N4674, “Overview of the MPEG-7 Standard, v 6.0”, J. M. Martínez, ed., MPEG Requirements Group, Jeju, Mar. 2002.

[42] A. K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, 1989.

[43] K. Karu, A.K. Jain, R.M. Bolle, “Is there any texture in the image?”, *Proceedings of the 13th International Conference on Pattern Recog.*, vol. 2, pp. 770-774, 1996.

[44] M.Kazanov, “A new color image segmentation algorithm based on watershed transformation”, *Proceedings of the 17th ICPR*, vol.2, pp. 590-593, Aug. 2004.

[45] S. Kiranyaz, K. Caglar, E. Guldogan, O. Guldogan, and M. Gabbouj, “MUVIS: A Content-Based Multimedia Indexing and Retrieval Framework”, *Proc. of the Seventh International Symposium on Signal Processing and its Applications*,, pp. 1-8, Paris, France, Jul. 2003.

[46] S. Kiranyaz, M. Gabbouj, “A Novel Multimedia Retrieval Technique: Progressive Query (WHY WAIT?)”, *In Proc. of WIAMIS Workshop*, Lisboa, Portugal, 2004.

[47] S. Kiranyaz and M. Gabbouj, “A Dynamic Content-based Indexing Method for Multimedia Databases: Hierarchical Cellular Tree”, *In Proc. of IEEE International Conference on Image Processing*, Genova, Italy, Sep., 2005.

[48] S. Kiranyaz, M. Ferreira, M. Gabbouj, “Automatic Object Extraction over Multi-Scale Edge Field for Multimedia Retrieval”, *IEEE Transactions on Image Processing,* vol. 15, pp. 3759-3772, 2006.

[49] S. Kullback and R. A. Leibler, “On information and sufficiency”, *Annals of Mathematical Statistics,* vol. 22, pp. 79-86, 1951.

[50] D. T. Lee, “Medial Axis transform of a planar shape”, *IEEE Transactions on PAMI*, vol. 4, pp. 363–369, 1982.

[51] S. Y. Lee, and F. H. Hsu, "2D C-string: a new spatial knowledge representation for image database systems," *Pattern Recognition*, vol. 23, pp. 1077-1087, 1990.

[52] S. Y. Lee, M.C. Yang, and J. W. Chen, "2D B-string: a spatial knowledge representation for image database system", *Proc. Second Int. Computer Science Conf.*, pp. 609-615, 1992.

[53] Y. Liu, W. Ma, D. Zhang, G. Lu, "Efficient texture feature extraction algorithm arbitrary-shaped regions", *7th International Conference on Signal Processing*, vol.2, pp.1037-1040, Sep. 2004.

[54] Y. Liu, D. S. Zhang, G. Lu, and W.-Y. Ma, "Region-Based Image Retrieval with High-Level Semantic Color Names", *In Proc. of IEEE 11th International Multi-Media Modelling Conference*, pp. 180-187, Melbourne, Australia, 2005.

[55] V. Luc and P. Soillt, "Watershed in digital spaces: An efficient algorithm based on immersion simulations". *IEEE Transactions on PAMI*, vol. 13, pp. 583-598, 1991.

[56] W.Y. Ma, and B.S. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases", *IEEE Inernationa. Conference. on Image Processin*g, vol. 1, pp. 568-571, Santa Barbara, USA, Oct. 1997.

[57] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations", *Proc. of 5th Berkeley Symposium. On Math. Stati. and Probability*, pp. 281-296, 1967.

[58] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and Texture Descriptors", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 703-715, Jun. 2001.

[59] B. Manjunath, P. Wu, S. Newsam, H. Shin, "A texture descriptor for browsing and similarity retrieval", *Journal of Signal Processing: Image Communication,* vol. 16, pp. 33-43, Sep. 2000.

[60] K. V. Mardia, T. J. Hainsworth, "A spatial thresholding method for image segmentation", *IEEE Transactions on PAMI,* vol. 10, pp. 919-927, Nov. 1988.

[61] V. Mezaris, I. Kompatsiaris and M. G. Strintzis, "Still Image Segmentation Tools for Object-based Multimedia Applications", *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 18,, pp. 701-725, Jun. 2004.

[62] A. N. Moga and M. Gabbouj, “Parallel marker-based image segmentation with watershed transformation”, *Journal of Parallel and Distributed Computing*, vol. 51, pp. 27-45, 1998.

[63] A.Mojsilovic and E. Soljanin, “Color Quantization and Processing by Fibonacci Lattices”, *IEEE Trans. on Image Processing*, vol. 10, pp. 1712-1725, Nov. 2001.

[64] A. Mojsilovic, J. Hu and E. Soljanin, “Extraction of Perceptually Important Colors and Similarity Measurement for Image Matching, Retrieval and Analysis”, *IEEE Trans. on Image Processing*, vol. 11, pp. 1238-1248, Nov. 2002.

[65] F. Mokhtarian, S. Abbasi, and J. Kittler, “Robust and efficient shape indexing through curvature scale space”. *In Proceedings of British Machine Vision Conference*, pp. 53-62, Edinburgh, 1996.

[66] O. J. Morris, M. de J. Lee, and A. G. Constantinides, “Graph theory for image analysis: An approach based on the shortest spanning tree,” *IEE Proc. F., Communications. Radar & Signal Processing,* vol. 133, pp. 146–152, 1986.

[67] MUVIS. http://muvis.cs.tut.fi

[68] A. Natsev, R. Rastogi, K. Shim, “WALRUS: a similarity retrieval algorithm for image databases”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 301-318, 2004.

[69] M. Nabil, J. Shepherd, A. H. H. Ngu, “2D Projection Interval Relationships: A Symbolic Representation of Spatial Relationships”, *Symposium on Large Spatial Databases*, pp. 292-309, 1995.

[70] T. Ojala, M. Pietikainen, D. Harwood, “A comparative study of texture measures with classification based on feature distributions”, *Pattern Recognition*, vol. 29, p. 51-59, 1996.

[71] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang. “Supporting similarity queries in MARS”, *In Proceedings of the 5th ACM International Multimedia Conference*, pp. 403-413, Seattle, Washington, 1997.

[72] G. Pass, and R. Zabith, “Histogram refinement for content-based image retrieval”, *IEEE Workshop on Applications of Computer Vision*, pp. 96-102, 1996.

[73] A. Pentland, R.W. Picard, S. Sclaroff, “Photobook: Tools for Content Based Manipulation of Image Databases”, *In Proc. of SPIE (Storage and Retrieval for Image and Video Databases II),* 2185, pp. 34-37, 1994.

[74] M. Peura and J. Iivarinen, “Efficiency of simple shape descriptors” *In 3rd International Workshop on Visual Form*, Capri, Italy, May 1997.

[75] I. Pratikakis, I. Vanhamel, H. Sahli, B. Gatos, S.J.Perantonis, “Unsupervised watershed-driven region-based image retrieval“, *IEE Proceedings on Vision, Image and Signal Processing*, vol. 153, pp. 313- 322, Jun. 2006.

[76] Y. Rubner, C. Tomasi, L.J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *Int. Journal of Computer Vision*, vol. 40, pp. 99-121, 2000.

[77] H. Samet, “The quadtree and related hierarchical data structures”, *ACM Computing Surveys*, vol.16, pp. 187-260, 1984.

[78] J. Shi, J. Malik. “Normalized cuts and image segmentation”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 731-737, 1997.

[79] K. C. Ravishankar, B. G. Prasad, S. K. Gupta, and K. K. Biswas, “Dominant color region based indexing for CBIR”. *Inter. Conf. on Image Analysis and Processing*, pp. 887-892, 1999.

[80] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, “Content-based image retrieval at the end of the early years”, *IEEE Transactions on PAMI*, vol. 22, pp. 1349-1380, Dec. 2000.

[81] J.R. Smith, S. Chang, “Transform Features for Texture Classification and Discrimination in Large Image Databases”, *Proc. IEEE Inter. Conf. on Image Processing,* vol. 3, pp. 407-411, 1994.

[82] J.R. Smith and S. F. Chang, “VisualSEEk: a fully automated content-based image query system”, *In Proc. of ACM Multimedia*, Boston, Nov. 1996.

[83] M. Stricker and M. Orengo, “Similarity of color images”, *In SPIE Conf. on Storage and Retrieval for Image and Video Databases III*, vol. 2420, pp. 381-392, Feb. 1995.

[84] M. Stricker, and M. Orengo, “Color indexing with weak spatial constraint”, *Proc. SPIE Conf. On Visual Communications*, 1996.

[85] H. Tamura, S. Mori, T. Yamawaki, “Textural features corresponding to visual perception” *IEEE Trans. on Systems, Man and Cybern.*, vol. 8**,** pp**.** 460-472, 1978.

[86] C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images”, *Proceedings of the IEEE Inter. Conf. on Computer Vision*, Bombay, India, 1998.

[87] Virage. [Online] http://www.virage.com

[88] J.Wang, J. Li, G. Wiederhold, “SIMPLicity: Semantics-sensitive integrated matching for picture libraries”, *IEEE Transactions on PAMI*, vol. 23, pp. 947-963, 2000.

[89] S. Wong, W. Leow, ”Color segmentation and figure-ground segregation of natural images”, *Proc. International Conf. on Image Processing*, vol. 2, pp. 120-123, 2000.

[90] A. Yamada, M. Pickering, S. Jeannin, L. Cieplinski, J.R. Ohm, and M. Kim, “MPEG-7 Visual part of eXperimentation Model Version 9.0 ISO/IEC JTC1/SC29/WG11 N3914”, 2001.

[91] YouTube: http://www.youtube.com

[92] D. S. Zhang and G. Lu, "Shape Based Image Retrieval Using Generic Fourier Descriptors", Signal Processing: Image Communication, vol. 17, pp. 825-848, 2002.

[93] Q. Zhou, L. Ma, M. Zhou, D. Chelberg, “Strong Image Segmentation from a Data-driven Perspective: Impossible?”, 6$^{th}$ *IEEE Southwest Symposium on Image Analysis and Interpretation*, 2004.