UNIVERSITY OF SOUTHAMPTON

Preprocessing for Content-Based Image Retrieval

by

Wasara Rodhetbhai

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Engineering, Science and Mathematics

School of Electronics and Computer Science

May 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Wasara Rodhetbhai

The research focuses on image retrieval problems where the query is formed as an image of a specific object of interest. The broad aim is to investigate pre-processing for retrieval of images of objects when an example image containing the object is given. The object may be against a variety of backgrounds. Given the assumption that the object of interest is fairly centrally located in the image, the normalized cut segmentation and region growing segmentation are investigated to segment the object from the background but with limited success. An alternative approach comes from identifying salient regions in the image and extracting local features as a representation of the regions. The experiments show an improvement for retrieval by local features when compared with retrieval using global features from the whole image.

For situations where object retrieval is required and where the foreground and background can be assumed to have different characteristics, it is useful to exclude salient regions which are characteristic of the background if they can be identified before matching is undertaken. This thesis proposes techniques to filter out salient regions believed to be associated with the background area. Background filtering using background clusters is the first technique which is proposed in the situation where only the background information is available for training. The second technique is the K-NN classification based on the foreground and background probability. In the last chapter, the support vector machine (SVM) method with PCA-SIFT descriptors is applied in an attempt to improve classification into foreground and background salient region classes. Retrieval comparisons show that the use of salient region background filtering gives an improvement in performance when compared with the unfiltered method.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. First of all, I would like to thank Professor Paul Lewis for his help, support, encouragement, teachings and supervision. His wise academic advice and ideas have played an extremely important role in the work presented in this thesis. Without Paul's support, this thesis would not have been possible. Besides my advisors, I would like to thank my examiners, Dr Tony Pridmore and Dr Kirk Martinez, who asked me interesting questions and gave insightful comments.

Secondly, I would like to thank my friends and colleagues in the IAM group - in particular, Jonathon Hare, Simon Goodall, Jiayu Tang, - for their academic inspirations and discussions. I would particularly like to thank Aniza Oatman, Ayomi Bundara, Thanyalak Maneewattana and Jaime Cerda Jacobo for their kind help in my study and social life. Outside of the lab, I would like to thank Tasanawan Soonklang and Sangob Taplaintong for their friendship and encouragement.

In addition, I would also like to thank the Royal Thai Government for giving me an opportunity to further my PhD. study.

My grateful thanks also go to my parents and my lovely sisters for their support. Especially, I would like to give my special thank to my beloved husband Sethalat Rodhetbhai who is always beside me.

Chapter 1 Introduction

Nowadays, the new advanced digital technology for capturing and storing media is developing rapidly. Using this electronic equipment is also easier and more comfortable than in the past and is at a much cheaper cost. It is undoubtedly seen that the number of images and other media are increasing dramatically. This is one of the reasons why image retrieval has been of great interest and in great demand.

Image retrieval is becoming more widely developed following the success of text retrieval technology for more than twenty years. In early methods, the multimedia collection is manually annotated with text and textual queries are used as the basis for multimedia retrieval. However, this method has problems due to the high cost of manual text annotation for large collections and also the limited information that can be captured easily in text. Many researchers are exploring the use of the content of multimedia to facilitate or assist the retrieval process. The difficulty of indexing, matching and retrieving multimedia information has led to the proposal of numerous novel techniques.

Image retrieval using image features extracted from the image content is known as Content Based Image Retrieval (CBIR). There are many CBIR systems that have been developed and some of these are discussed briefly in Chapter 2 and more fully in the Appendix A. Moreover, CBIR plays important roles in many fields, for example, the medical field (Shyu, Brodley et al. 1999) and even the arts (Lewis, Martinez et al. 2004). ARTISTE and SCULPTEUR are examples of projects using CBIR to retrieve and browse objects from museums. Further details of these projects are given in

Appendix A.3. With metadata and ontologies the IR systems also provide other useful information and are increasingly accessed via the World Wide Web.

Processes of CBIR system are shown as a high level work flow diagram in Figure 1.1. A process of image retrieval begins with a query which is an input of IR system is extracted features. Content-based visual features such as colour, texture, shape, contour or intensity become the important information called feature vectors in a vector space. They are used to decide similarity or difference between a query and images in the database.

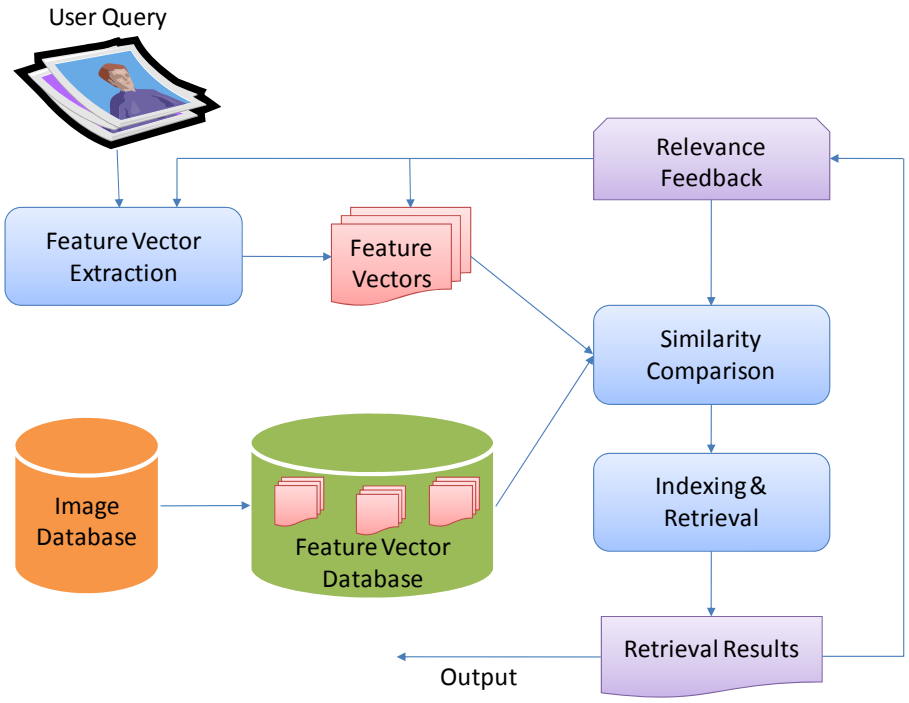


Figure 1.1 High level work flow diagram of CBIR system

The similarity is ordered by the distance between features in a vector space. Images with lower feature distance will be ranked in a higher order (i.e. more similar to the query) than those with greater feature distance.

The main problem in Content-Based Image Retrieval called *semantic gap* corresponds to the mismatch between users' requests and the way a retrieval system try to satisfy

these requests. The semantic gap between the low-level similarity and the high-level user's query leads classical search algorithms to erroneous results.

CBIR researchers have studied various methods to narrow the semantic gap. Relevance feedback (RF) is one of methods in order to bridge this gap by allowing systems to interactively improve their discriminatory capabilities. (Traina, Marques et al. 2006) is one of works using RF in a CBIR application. Although there are a large number of content-based retrieval techniques, the retrieval results have still not reached a satisfactory level of performance for widespread general application and bridging the semantic gap.

Focusing on the class of retrieval problems where the query is formed by an image of an object of interest, users often expect to retrieve images that contain the specified object or region in the top rank. Finding a similar object in the image database is often challenging especially when the object images have possibly been captured from different views or within different scenes or backgrounds. A motivating scenario for the work in this thesis is where a museum may have images of artefacts set against a variety of backgrounds. A searcher may then wish to use one image to retrieve images of other artefacts in the same class.

Even in this particular class of retrieval problems, retrieval results have still not reached a satisfactory and stable level of performance. The images that contain the required object are sometimes ordered at lower rank in the retrieval than some irrelevant images. One of the reasons is that features which are representing the image do not come from the object but from background which typically is not involved with the particular object in the image. In this thesis, approaches based on preprocessing techniques for image retrieval are investigated to improve the overall system performance.

In the first instance, we assume the object retrieval is for particular object collections in which all objects are captured individually in the centre of the image with relatively straightforward backgrounds. The object retrieval system should ideally be based on the central object area only.

To locate the object area and derive the object parts, a number of techniques in the computer vision field have been considered to achieve this objective. For example, in the field of graphics document understanding, Tombre and Lamiroy (Tombre and Lamiroy 2003) considered the concept of *Symbol Spotting* which involves finding graphical symbols in a collection of images. Symbol spotting methods tend to use the most discriminative features to describe the objects to be located. Given a single example of a symbol (usually cropped from a document itself) as the query, the system might return a ranked list of segmented locations where the queried symbol is likely to be found. In (Rusinol and Llados 2008), the spotting method begins with finding keypoints which are extracted from a document image and a local descriptor is computed at each of these points of interest. With this organization, finding an object (symbol) in a certain location and under a certain pose is possible. Normally symbol spotting is applied to graphical documents such as engineering drawings, maps, diagrams etc. For general images which are usually more complex than graphical documents, the symbol spotting idea could possibly be applied but would need more discriminative features to make indexing and searching possible.

Although user interaction could help a system (Hoiem, Sukthankar et al. 2004) to locate the object for CBIR, the thesis aims to investigate approaches which based on the automatic CBIR system. Therefore any helps from users are not required. A number of techniques appeared in (Elhabian, El-Sayed et al. 2008) proposed background removal approaches to locate an object. They are also motivating but all techniques focuses on moving objects in video.

Image segmentation is the process of partitioning an image into multiple regions. Image segmentation algorithms usually do not assume prior knowledge from objects. Moreover, it generally has the ability to operate with natural images. In the first part of the thesis, automatic segmentation is explored as a preprocessing stage for the retrieval. By using image segmentation, all images in the database are segmented to leave only an object region. Then features from the object region are extracted and used as the image representation.

After this first approach, slightly more general image collections are considered. All objects are still taken individually but they could be located anywhere as a major part of the image. Moreover the object may appear against more complex backgrounds such as outdoor scenes. In this case, object extraction using image segmentation as preprocessing for image retrieval may face problems.

Instead of using all parts of an object as the image representation, another approach is based on the idea that an image can be represented by local features from the particular interest points of the image, often referred to as salient regions. Using local features from salient regions as the image representation, some of problems from flawed segmentation are avoided.

Unfortunately, image retrieval by salient regions still sometime faces the irrelevant information problem. In this thesis, there are a number of methods proposed to alleviate this problem. They focus on filtering salient regions. The results of using background filtering as a preprocessing stage for image retrieval are measured. The developed methods would be applied to use in object-based image retrieval systems which focusing in particular object details within an image for example searching or browsing a car in the car image categories.

1.1 Aims and Objectives

The objectives of this research are as follows.

- To investigate and develop preprocessing techniques to extract regions of the image that represent the object as reliably as possible from both uncluttered and cluttered backgrounds.
- To investigate representing the extracted “objects” by feature descriptors and hence retrieval of images by matching the object descriptors.
- To explore the possibility of automatic classification of images given the availability of similar pre-annotated images in the collection from which the association between image features and foreground and background annotations may be learned.
- To implement the proposed approach to improve object-based image retrieval.

- To evaluate the approach and the usability of the experimental system.

1.2 Thesis Structure

This thesis describes the work of the author in attempting to achieve the objectives outlined earlier in this chapter. Chapter 2 documents and describes background and existing research towards these goals. Chapter 3 through Chapter 7 describe the actual research undertaken by the author, and Chapter 8 presents the conclusions of this research together with some views of the author regarding directions for future research. The following list describes the structure and content of the thesis on a chapter by chapter basis.

Chapter 2: Background. In this chapter, there is an introduction to Content-Based Image Retrieval, image segmentation and salient regions. In addition, the performance evaluation methods for CBIR are described.

Chapter 3: Central Object Retrieval using Segmentation. In this chapter, some image segmentation techniques are developed for object segmentation. The normalized cut segmentation; grid segmentation; and region growing segmentation are investigated and compared in the context of object segmentation from background. The approaches are evaluated in the context of image retrieval.

Chapter 4: Image Retrieval using Salient Regions. This chapter introduces the idea of using salient regions as a basis for image retrieval. Neighbourhood pixels around the salient points are extracted and used to generate feature vectors which are then used as the basis for retrieving similar images in the database. Retrieval performance based on local features from salient regions is compared with the retrieval performance based on features extracted from the entire image and also with the segmentation based preprocessing of Chapter 3. A simple synthetic dataset is used in the evaluation. Problems which influence the retrieval results are discussed.

Chapter 5: Background Filtering. A new method to filter background salient regions is proposed in this chapter. The knowledge for the learning system comes from a collection of background images only. This chapter is based on a paper (Rodhetbhai and Lewis 2007) presented at the VISUAL 2007 conference.

Chapter 6: K-Nearest Neighbour Classification for Background and Foreground. This chapter presents an approach to improve the background salient region filtering by utilising both foreground and background information for training the system.

Chapter 7: Foreground-Background Classification using SVM. Here the support vector machine approach to machine learning is applied to the problem of classifying background and foreground using salient regions and the approach is compared with earlier methods.

Chapter 8: Conclusion and Future Work. An overall summary of the thesis and a discussion of future work are presented in this final chapter.

Chapter 2 Background

In this chapter issues, details and techniques relating to the research are introduced. It is divided into 4 parts. Firstly, content-based image retrieval (CBIR) is reviewed. The second part presents a discussion of image segmentation. The third part gives information about salient region detectors and representations. Methods to measure the accuracy of image classification and retrieval are mentioned in the fourth part and the final part of the chapter is about image collections. Information on the SVM machine learning technique is given in the relevant chapter (Chapter 7).

2.1 Content-Based Image Retrieval (CBIR)

Image retrieval using features of the content of the image is known as Content-Based Image Retrieval (CBIR). The features of an image such as those based on colour, shape or texture can be extracted and used for matching with other images.

Many CBIR systems have been developed both as stand-alone systems and on the World-Wide-Web (WWW) for example, QBIC (Flickner, Sawhney et al. 1995) , Photobook (Sclaroff, Pentland et al. 1996), VisualSEEk (Smith and Chang 1996), NeTra (Ma and Manjunath 1999), MARS (Ortega, Rui et al. 1997), Blobworld (Carson, Thomas et al. 1999) and so on (Smeulders, Worring et al. 2000; Deb and Zhang 2004). Many application areas require efficient content-based image retrieval increasingly such as biomedicine (Shyu, Brodley et al. 1999), military, commerce, art and culture (Goodall, Lewis et al. 2004), education, entertainment, etc. Further details of some of these systems are given in Appendix A.

Beside query by keyword, users may query a CBIR system in different ways. *Query by Example (QBE)* means searching images with a query image or a part of the image. *Query by Sketch* is when the users draw a rough approximation of the image they are looking for. For this type of query, the system locates images whose layout matches the sketch.

2.1.1 Feature Extraction

Generally, CBIR systems are able to extract visual features (more details in Section 2.1.3) which are used to represent the contents of an image. Each feature is typically encoded in an n-dimensional vector, called a *feature vector*.

2.1.2 Similarity Measurements

In a vector space, the distance between two feature vectors is computed. The smaller the distance, the greater the similarity. Generally, the Euclidean distance measurement is the most commonly used to calculate the distance. The distance is the root of sum of the squared differences between the pairs of feature vector elements.

$$D_E(x,y) = \|x-y\|_2 = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2}$$

Where $D_E(x,y)$ is the Euclidean distance between two vectors $x, y \in R^n$.

Beside the Euclidean distance, there are many ways to measure a feature distance between two images for example; the Manhattan distance; the Mahalanobis distance (Zhang and Lu 2003); Earth Mover's Distance (EMD) (Stricker and Orengo 1995); and the chord distance. For some feature extraction techniques, a specific similarity measurement may be proposed.

2.1.3 Image Features

Although there is no consensus for feature classification, (Marques and Furht 2002) classified features into three categories:

Low-level features: these features are extracted from the pixel values directly or from simple groupings of pixels, for example; colour, texture, or shape.

Middle-level features: example included blobs or regions gained from image segmentation.

High-level features: semantic information that represents the image, the objects it contains and the categories to which the image belongs.

In the following section, low level features such as colour, texture and shape are described.

Colour Features

Besides the colour histogram which is commonly used, there are many other colour feature representations which have been applied in CBIR. Stricker and Orengo proposed *Colour Moments* (Stricker and Orengo 1995) are the moments of the colour distribution and their use overcomes the quantisation effects of colour histograms. Most information is concentrated in the low order moments and Stricker and Orengo used the first three central moments of the colour components (H, S and V) of the probability of each colour.

However, colour histograms and colour moments lack spatial information about colour location and pixel relation to other pixels. Techniques such as *Colour Sets* (Smith and Chang 1995) and *Colour Coherence Vector* (CCV) (Pass, Zabih et al. 1996) have been proposed to solve this problem.

Texture Features

Texture features capture information about the repeating patterns in the image. In (Marques and Furht 2002), texture features may be categorized into three models: Statistical models; Spectral models and Structural models.

Statistical models: This group includes statistical moments of the gray-level histogram, especially, the second moment (the variance), uniformity and average entropy, descriptors (energy, entropy, homogeneity, contrast, etc.). Haralick et al. (Aksoy and Haralick 1998) proposed the image gray–level co-occurrence matrix which is also classified in this model.

Spectral models: These derive from the spectral density function in the frequency domain. Coefficients of a 2-D transform (e.g., the Wavelet transform), Gabor wavelet transform (Manjunath and Ma 1996).

Structural models: techniques that explain texture in terms of primitive texels (Shapiro and Stockman 2001). This model is popular for artificial, regular patterns.

Shape Features

Generally, shape features require a region identification process. That means an image has to be segmented before extracting most shape features. The shape representation typically is divided into two categories; boundary-based and region-based. Fourier descriptors (Zhang and Lu 2001) are a representative for the boundary-based method while moment invariants are characteristic of the region-based method. Hu (Hu 1962) described a widely used shape representation using seven rotation, scale and translation independent moment combinations which are widely known as the Hu Moments.

2.2 Image Segmentation

Segmentation is used to create regions in an image and is a popular preprocessing stage for many image analysis tasks. Access at object level in images is made possible by image segmentation but the process is problematic as the appropriate level of segmentation for a particular task is usually difficult to ascertain automatically. Some CBIR systems such as IMALBUM (Idrissi, Lavoué et al. 2004) and SIMPLIcity (Wang, Li et al. 2001) have been achieved by segmenting an image into regions with uniform low-level features and the systems provide a graphic user interface to allow a user to select the region of interest.

Segmentation can be classified into two paradigms: *Bottom-Up* (model-based) and *Top-Down* (visual feature-based). The bottom-up approach is segment an image into regions first, and then considers corresponding regions for a single object. A graph representation is commonly used and partitions it into subsets. The normalized cut segmenation (Shi and Malik 1997) is one of well known techniques. The details of this algorithm is in Chapter 3. There are other bottom-up segmentations for example watershed method (Vincent and Soille 1991) and region growing method (Tremeau and Borel 1997) etc.

The main problem of bottom-up methods is that over segmenting may occur and required objects are fragmented into many regions or under segmenting may occur and regions of the object may be merged with the background.

For top-down methods, prior knowledge about an object is required for segmenting. Problems frequently occur from the variety of appearances of objects in a class and the results from the segmentation may not be correct.

Some researchers such as Eran et al. (Borenstein, Sharon et al. 2004) tried to combine top-down and bottom-up segmentation to get a better segmentation. Nowadays, image segmentation is still a difficult topic and also a key open problem in the computer vision field.

Typically users search for the particular object of interest in images. Therefore, the ability to access the image at the object level could be effective for CBIR. There are some techniques proposed to extract an object of interest from an image using segmentation. For example, an interesting approach of automatic object extraction (Kim, Park et al. 2003) aims to extract the object of interest from a complex background based on the assumption that mostly interesting objects are located near the centre of the image. A central object in an image was defined as a set of characteristic regions located near the centre of the image. These regions were characterized as having significant colour distribution compared with surrounding regions. Firstly, the central area of an image and the surrounding region were determined by using the difference between the correlograms (Huang, Kumar et al. 1997) . Then these two types

of core regions were determined to characterize the foreground and the background in the image. The core object region was extended through merging its unlabeled neighbour regions, if they had similar colour distribution to it but were very dissimilar to the core background region. The final merging result, a set of regions connected to each other, was the central object.

2.3 Salient Points and Regions

Since some researchers ((Shao, Svoboda et al. 2003), (Hare and Lewis 2005) etc.) point out that global features which are computed from the entire image may not be good enough to handle heterogeneous images in the database, the use of local image features have been introduced. The notion of interest point detection is used to extract local properties of an image. In (Loupias and Sebe 1999) pixels and groups of pixels that are the most important in an image are called salient points and salient regions. Historically they were called interest points.

In 1997, Schmid and Mohr (Schmid and Mohr 1997) proposed using corners as interest points to match images in the database. Following this, some researchers have tried to find other ways to detect salient points. For example; Lowe (Lowe 1999) used the Difference-of-Gaussians pyramid to calculate the salient point location; Tian et al. (Tian, Sebe et al. 2001) used wavelets such as the Harr and Daubechies4 wavelet function; Tuytelaars and Van Gool (Tuytelaars and Gool 1999) suggested a detector based on intensity extrema.

One of the comparisons of salient point detectors also appears in (Sebe, Tian et al. 2002). In this paper, the local features based on colour and texture information are extracted around the salient points and then matched to retrieve images in the database. Chiou-Ting Hsu (Hsu and Shih 2002) proposed a method in a similar way and designed a voting algorithm to rank the similarity of the retrieved images.

In the salient region field, one of the well known descriptors is the SIFT descriptor (Lowe 1999) (more details in Chapter 4) which are developed into the PCA-SIFT by (Ke and Sukthankar 2004). Mikolajczyk (Mikolajczyk and Schmid 2005) compared the

performance of descriptors and showed that the SIFT-based descriptors outperform other methods.

2.4 Evaluation Methods

Content-based image retrieval systems are evaluated in terms of retrieval effectiveness. The precision and recall which are frequently used in the field of information retrieval are applied to CBIR evaluation.

Image retrieval is measured by precision, which is the number of relevant images retrieved relative to the number of retrieved images, and recall, which is the number of relevant images retrieved, relative to the total number of relevant images in the database.

$$\text{Precision} = \frac{\text{Retrieved Relevant Items}}{\text{Retrieved Items}}$$

$$\text{Recall} = \frac{\text{Retrieved Relevant Items}}{\text{Relevent Items}}$$

2.4.1 Average Precision

The precision and recall are based on the whole list of documents returned by the system. Average precision emphasizes returning more relevant documents earlier. Average precision that combines precision, relevance ranking, and overall recall is defined as follows:

$$\text{Average Precision} = \frac{\sum_{r=1}^{N} (P(r) \text{ x } rel(r))}{\text{number of relevant documents}}$$

where N is the number retrieved, r is the rank, $P(r)$ is precision at a given cut-off rank, $rel(r)$ is a binary function on the relevance of a given rank (1 if the retrieved document at r is relevant and 0 otherwise). That is, average precision is the sum of the precision at each relevant hit divided by the total number of relevant documents in the collection.

2.4.2 Precision-Recall Graph

Precision and recall can be estimated for increasing numbers of retrieved images and the precision and recall values can be displayed in the form of the precision/recall curve. The recall is plotted on the x-axis and the precision is plotted on the y-axis. An ideal goal for retrieval system development is to increase both precision and recall values by making improvements to the system.

2.4.3 Confusion Matrix and Classification

The Confusion Matrix (Kohavi and Provost 1998) of a classification system contains information about the actual and predicted classifications assigned by such system. Performance of classification systems is commonly evaluated using the data in the matrix. Table 2.1 shows the confusion matrix for a two class classifier.

Actual \ Predicted	**Negative**	**Positive**
Negative	*a*	*c*
Positive	*b*	*d*

Table 2.1 Two-dimensional square matrix of actual and prediction

The entries in the confusion matrix have the following meaning:

a is the number of correct predictions that an instance is negative
b is the number of incorrect predictions that an instance is negative
c is the number of incorrect predictions that an instance is positive, and
d is the number of correct predictions that an instance is positive.

There are 6 standard metrics that can be derived from the 2 class confusion matrix.

The accuracy (AC) is the proportion of the total number of predictions that were correct.

$$Accuracy = (a+d) / (a+b+c+d)$$

The true positive rate or recall or sensitivity (TP) is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$True\ positive\ rate\ (Recall) = d / (b+d)$$

The false positive rate (FP) is the proportion of negative cases that were incorrectly classified as positive.

$$False\ positive\ rate = c / (a+c)$$

The true negative rate or specificity (TN) is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$True\ negative\ rate\ (Specificity) = a / (a+c)$$

The false negative rate (FN) is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$False\ negative\ rate = b / (b+d)$$

The precision (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation

$$Precision = d / (c+d)$$

In addition to the confusion matrix, the ROC graph is another way to examine the performance of classifiers. A ROC graph is a plot with the false positive rate on the X axis and the true positive rate on the Y axis. The point (0,1) means the false positive rate is 0 (none), and the true positive rate is 1 (all) so the point (0,1) is the perfect classifier because it classifies all positive cases and negative cases correctly. The point

(1,1) represents a classifier that predicts all cases to be positive, while the point (0,0) corresponds to a classifier that predicts every case to be negative. Point (1,0) is the classifier that is incorrect for all classifications. In many cases, there is a parameter from a classifier that can be adjusted to increase the true positive rate at the cost of an increased the false positive rate or decrease the false positive rate at the cost of a decrease in the true positive rate.

The standard method to compare ROC points in classification space is to compare the Euclidian distance from points to the perfect classifier, point (0,1). Points which are nearer to the point (0,1) can classify better than the more distant points.

2.5 Image Collections

Some websites such as those in the database list of the computer vision bibliography (Price 2009) and the test image lists of the computer vision lab in Carnegie Mellon (Anonymous 2005) provide a number of image collection lists used in different domains of image retrieval. Although there are no general standard test collection or evaluation sets available for image retrieval like Text REtrieval Conference (TREC) (Voorhees and Harman 2005) in the text retrieval field, some image datasets such as the Corel dataset and the Washington dataset (Anonymous 2004) are widely used to evaluate many CBIR systems. However, for those datasets such as the Corel dataset (Müller, Marchand-Maillet et al. 2002) warned users to beware when using them for performance measures in image retrieval.

The object class collections such as the Caltech datasets ((L. Fei-Fei 2004), (Griffin, Holub et al. 2007)) and the PASCAL visual object datasets (Anonymous 2009) which are used in the annual Visual Object Classes Challenge (VOC) have been used for object recognition and classification. In (Ponce, Berg et al. 2006) the recognition and classification performances using such datasets were investigated while (Pinto, Cox et al. 2008) demonstrated that tests based on those datasets may make the comparison of classification systems difficult.

In this research, specific datasets for training and testing are required because the problem in which we are interested require collections of images containing a single object mostly located in the central area of the image. To fulfil the objectives of the research, a number of datasets were created with ground truths labels. Some examples of images used in the experiments are shown in the main part of thesis and also in Appendix B.

2.6 Summary

Some of the basics required for this research have been presented in this chapter. The first part starts with the concept of content-based image retrieval, the different types of features and the important processes in CBIR such as features extraction and similarity measurement. Segmentation schemes and local features represented as salient regions which are the main methods for getting at the contents of objects are mentioned and these are explained in more detail in Chapter 3 and Chapter 4.

Performance measurement is required in both classification and retrieval. Precision and recall graphs are mainly used to measure image retrieval ability. The confusion matrix and ROC graph are also tools to evaluate classification.

Chapter 3 Central Object Retrieval using Segmentation

Finding images containing some particular regions or objects is one of the targets in image retrieval. Although images are captured individually, background information in each image may affect the results of a content based retrieval system. The same object in different images sometimes cannot be retrieved with a good rank when it is on a different background from the query. In order to alleviate this problem, ideally, if it could be identified, the background should not be included in the retrieval process.

Image segmentation could be used to separate regions of interest or objects in the scene. From the segmentation result, all pixels in each region are similar with respect to some characteristics, such as colour, texture, or intensity. These regions may be identified and useful for the subsequent image annotation or image analysis.

Segmentation is widely used in many applications. Some CBIR system such as Blobworld (Carson, Thomas et al. 1999) and NeTra (Ma and Manjunath 1999) allow users to query by image regions of interests derived from image segmentation. The SIMPLIcity system created by (Wang, Li et al. 2001) uses segmented regions to represent images.

In this chapter, we apply image segmentation in an attempt to get rid of background and extract a single salient object in the image which is assumed to be located near the central area of the image. The normalized cut segmentation is introduced in an attempt

to extract the central object. The region growing segmentation is one of the choices to produce a better segmentation result on an uncluttered background or pattern background. The retrieval performance using image segmentation on a simple collection is investigated. In the last section, retrieval with image segmentation and without segmentation is compared.

3.1 Normalized Cut Segmentation

The step of segmentation is required to attempt to isolate the object in an image query and in the database images. A graph cut segmentation called the normalized cut segmentation is investigated because it has been widely used in many image retrieval research projects (Brun 2004) and (Shi and Malik 1998).

In (Barnard, Duygulu et al. 2003), the normalized cut algorithm was compared with other image segmentation algorithms. The result showed that the normalized cut segmentation gave higher precision and recall in their application. It suggested that the normalized cut was possibly a good choice to extract an object from an image.

3.1.1 Normalized Cut

Segmentation by Graph-Theoretic clustering is an idea to model data into a graph and segment it by partitioning the graph (Martinez, Mittrapiyanuruk et al. 2004) and (Shapiro and Stockman 2001). Each data point is represented by a vertex in a weighted graph G = (V, E), where V is a set of nodes in an image and E is a set of edges connecting the nodes. Each edge (*u, v*) has a weight *w* (*u, v*) that represents the similarity between node *u* and *v*. A graph can be partitioned into two disjoint set, A and B by removing edges connecting the two parts. The cut is the degree of dissimilarity between two pieces which can be computed as the total weight of the edges. We can define the cut by equation [3.1].

$$cut\ (A, B) = \sum_{u \in A, v \in B} w(u, v) \qquad [3.1]$$

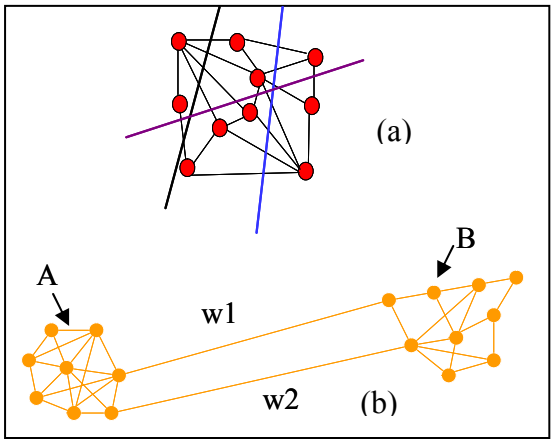


Figure 3.1 (a) Graph model and cuts
(b) Disjoint set A and B which associate with w1 and w2

The optimisation of the graph partitioning is to minimize the cut value. This method is called the *minimum cut* that was proposed by (Wu and Leahy 1993). Generally, the minimum cut that can bipartition the segments will be calculated recursively. However, sometimes the minimum cut can split small sets in the graph. Figure 3.2 shows that the cuts partition out only two nodes because these cuts have smaller values than the ideal cut.

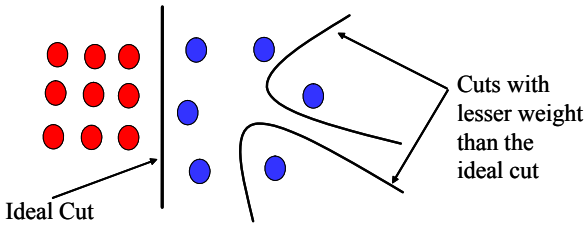


Figure 3.2 Case of a bad partitioning by minimum cut

The *normalized cut (Ncut)* (Shi and Malik 1997) was proposed to solve the above problem. The measure computes the cut cost as a fraction of the total edge connections to all the nodes instead of finding the total edge weight values connecting the two partitions.

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \quad [3.2]$$

where $assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$ is the connection from nodes in A to all nodes and similarly is defined for $assoc(B,V)$.

In fact the partition algorithm tries to minimize the disassociation between groups and maximize the association within the groups. Unfortunately, the minimizing of the normalized cut is an NP-complete problem.

Let x be an $N = |V|$ dimensional vector, $x_i = 1$ if node i is in A and -1, otherwise. Let W be an N x N symmetrical matrix with W(i,j) = w_{ij}. Let D be an N x N diagonal matrix with $D(i,i) = \sum_j W(i,j)$. In (Shi and Malik 1997) and (Shi and Malik 2000), Shi and Malik showed that minimizing Ncut from equation [3.2] can be reduced to minimizing a Rayleigh quotient:

$$\min_x Ncut(x) = \min_y \frac{y^T(D - W)y}{y^T D y} \quad [3.3]$$

with the condition $y(i) \in \{1, -b\}$ and $y^T D1 = 0$. By relaxing y to take on real values, the solution to equation [3.3] can be minimized by solving a generalized eigenvalue system in the form of [3.4].

$$(D - W)y = \lambda Dy \quad [3.4]$$

From the calculation, the optimal solution corresponds to the second smallest eigenvector. Therefore, an image can be subdivided by finding the second smallest eigenvector of equation [3.4] and can subdivide the existing graphs, each time using the eigenvector with the next smallest eigenvalue.

3.1.2 Weight Functions

The steps of the graph construction G = (V, E) are taking each pixel as a node and defining the edge weight function. This is an example of the weight function that was used in (Shi and Malik 1997) and (Shi and Malik 2000).

$$w_{ij} = e^{\frac{-\|F(i)-F(j)\|_2^2}{\sigma_I}} * \begin{cases} e^{\frac{-\|X(i)-X(j)\|_2^2}{\sigma_X}} & \text{if } \|X(i) - X(j)\|_2 < r \\ 0 & \text{otherwise} \end{cases} \quad [3.5]$$

where *X(i)* is the spatial location of node i, and *F(i)* is a feature vector based on intensity, colour, or texture information at that node defined as:

- *F(i)* = 1, in the case of segmenting point sets,
- *F(i) = I(i),* the intensity value, for segmenting brightness images,
- *F(i)* = [*v,v.s.sin(h),v.s.cos(h)*]*(i)*, where h; s; v are the HSV values, for colour segmentation,
- *F(i)* = [$|I * f_i|$, ..., $|I * f_n|$]*(i)* , where the f_i are Difference of offset Gaussian (DOOG) (A.Young, Lesperance et al. 2001) filters at various scales and orientations as used in the case of texture segmentation.

From (Hu 1962), the weight w_{ij} = 0 for any pair of nodes i and j that are more than r pixels apart.

Figure 3.3 shows some experiment results appearing in (Shi and Malik 2000) which are evidence that the normalized cut algorithm works reasonably well in image segmentation.

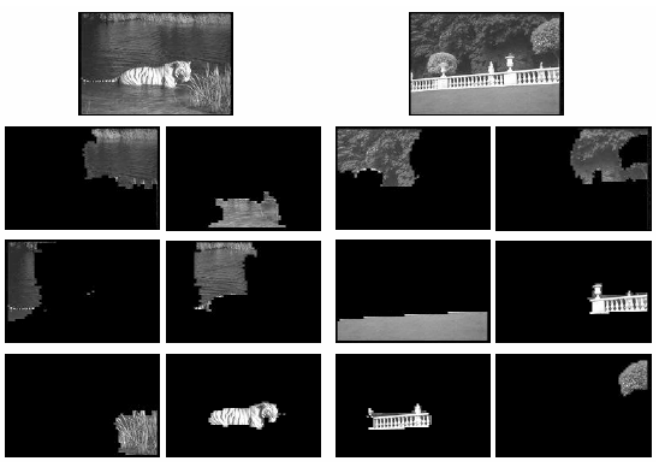


Figure 3.3 Some results of experiments which appeared in (Shi and Malik 2000)

3.1.3 Normalized Cut Segmentation Experiments

The normalized cut segmentation is used to separate an object and background into different segments. In this experiment, the results on various kinds of images: (1) a synthetic image with one colour background; (2) a real image are demonstrated. The steps of segmentation are as follows

1. Set up a weighted graph G = (V, E) and set the weight on the edge.
2. Solve $(D - W)x = \lambda Dx$ for eigenvectors with the smallest eigenvalues.
3. To bipartition the graph, use the eigenvector with the second smallest value.
4. Decide if the current partition should be subdivided and recursively repartition if necessary.

To investigate segmentation results, segmentation which is based on three features; colour intensity and intervening contour; are compared. Image segmentation by the intensity is weighted from the intensity value (0-255) of each pixel while the image segmentation by colour is calculated from values of H (Hue), S (Saturation), and V (Value). To obtain the H, S, and V values, an input RGB image is changed to an HSV image before segmentation. The colour feature vector from each pixel can be calculated as

$$F(i) = [v, v.s.\sin(h), v.s.\cos(h)](i) \qquad [3.6]$$

For intervening contour features, the weight is calculated from the maximum energy of edge information. Edge information is derived from an edge detector. Information about the strength of a contour can be obtained through orientation energy (OE) from elongated quadrature filter pairs.

The orientation energy is normally high at a sharp contour. In the experiment, the Canny algorithm (Canny 1986) is used as an edge detector. If the orientation energy between nodes is strong, it means that those nodes belong to two different partitions. The maximum of edge response is a peak in contour orientation energy. The weight matrix by intervening contour features is calculated as

$$W_{ij} = \exp(-\text{Max}\{\text{EdgeResponse}(u)\}) \quad [3.7]$$

u is the edge between $[(x_i,y_i) ; (x_j,y_j)]$.

Sampling Points

In principle each pixel is compared with all others. That means if the image size is 100 by 100, space for a big matrix 10,000 x 10,000 must be reserved to keep the weighted matrix from the comparison of all pixels. To reduce the calculation time, a sample radius (r) and a sample rate are used to sample the points that we want to compare around each pixel. The sample radius shows how far the area extends from the pixel. The sample rate sets the fraction of selected points within the sample radius. In experiments, the sample radius is set to 10 and sample rate is set to 0.3. Figure 3.4 illustrates numbers of points varied by sample radius (r).

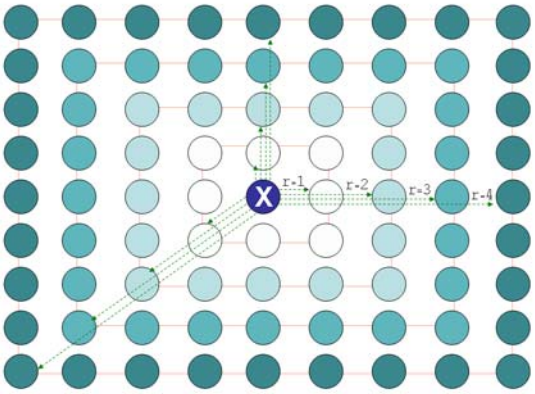


Figure 3.4 The number of points from different values of sample radius (r)

3.1.4 Results

In the first experiment, the normalized cut algorithm is applied to a simple 100 x 100 black and white image with $\sigma_I = 0.01, \sigma_X = 4, r = 10$. In Figure 3.5, the red line shows the boundary of an object in the image. It can be noticed that the extraction using different features on an uncomplicated image seems similar.

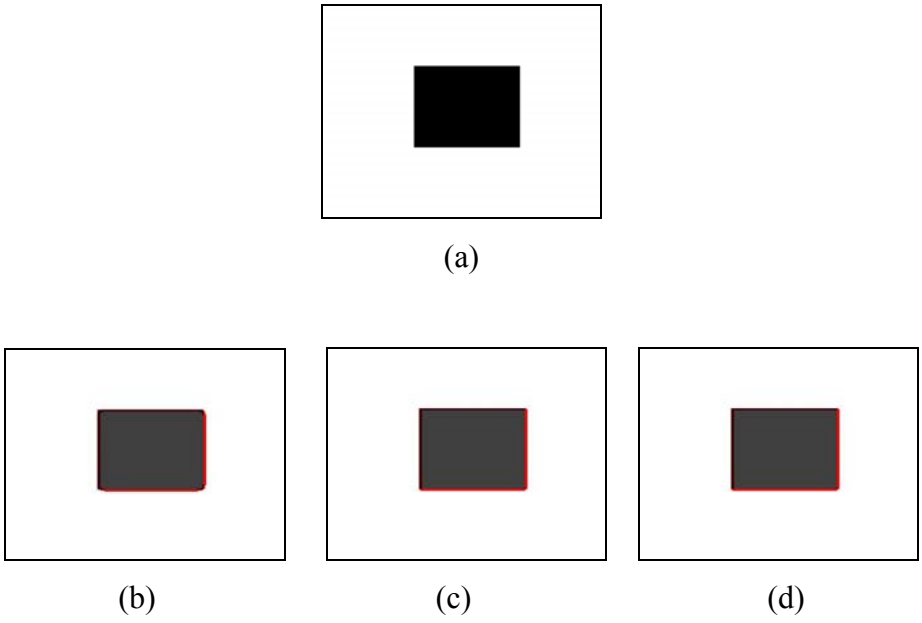
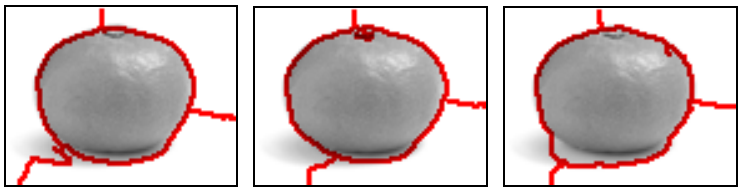


Figure 3.5 The result of segmentation. (a) an original image,
(b) using intervening contours, (c) using colour and (d) using intensity

From Figure 3.5, there are specific examples for which all three features can produce satisfactory results. With the same parameters, Figure 3.6 shows an orange image that is segmented into 4 segments. Segmentation results based on different features now shows the object falls in a central single region.

(a)



(b) (c) (d)

Figure 3.6 The result of segmentation. (a) an original image,
(b) using intervening contours, (c) using colour and (d) using intensity

Then, four more complex images from the Victoria and Albert museum collection are tested. Figure 3.7 shows the test images and the results with six regions. Red lines show segmented regions of each query image.

For the first three query images (1Q-3Q), the results of segmentation by all three features look good but over-segmentation appears in some result images such as 1E, 2E, 3C and 3I. For the fourth query image (4Q), under-segmentation has occurred. The top of the object is not included with the main central region in each image. This happens because background and object are a very similar colour in places.

In the case of images in which an object have almost only one shade, such as 1Q and 3Q, the segmentation by colour (see 1C and 3C) obtains better results than the segmentation for the object that contains many colours such as 2Q (see 2C).

The appearances of segmentation results using different features are not always similar depending on the types of features used. The over-segmentation and under-segmentation can happen during automatic segmentation. The number of segments requested and other parameter settings also cause different segmentation results. Another important point is that data driven segmentation of objects is impossible in the general case unless assumptions can be made or prior knowledge about the objects is available. It can be seen that the normalized cut segmentation has many parameters to set, is computationally intensive and has large memory requirement.

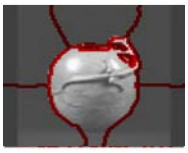
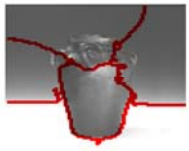
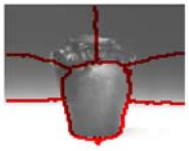
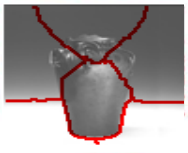
Query Image	**By Intervening Contour**	**By Colour**	**By Intensity**
 1Q	 1E	 1C	 1I
 2Q	 2E	 2C	 2I
 3Q	 3E	 3C	 3I
 4Q	 4E	 4C	 4I

Figure 3.7 Examples of image segmentation using different features

3.2 Segmentation using Region Growing Technique

In some images, the object in the image can be easily detected when it is located on an uncluttered background, for example for which most of the background is based on just one or two colours. In such cases a region growing algorithm for segmentation may be appropriate. Region growing algorithms begin with seed pixels and neighbouring pixels are aggregated into the region if they are similar to the seed. In the same way, this segmentation finds background pixels with similar value and masks them out as background. To let the system know the characteristics of the background, a sample pixel set from the background is selected. By assuming the object of interest mostly locates near the central area of the image, the area near the border of the image can be assumed to be pixels from the background.

The segmentation algorithm in this section is applied from a segmentation algorithm used with a flash and non-flash images (Braun and Petschnigg 2002). It begins by converting an input image from RGB colour space to YCbCr colour space because YCbCr has less covariance. Then, a number of pixels around the four borders, which are assumed to be background pixels, create a rectangular frame.

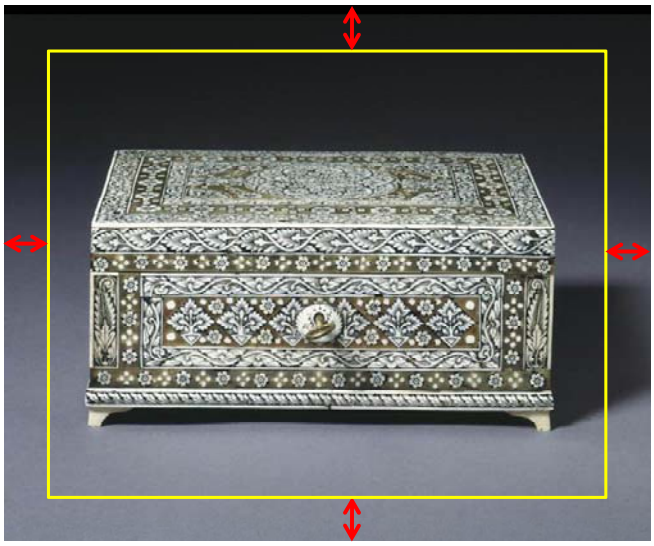


Figure 3.8 A rectangular frame as a background set around an image

Figure 3.8 shows a rectangular frame area which is assumed to be background dataset. A background "cube" is created from this background dataset. The cube dimensions are padded with a tolerance value (tol) of selected pixel. Each pixel from the entire image is classified as lying within the cube or not. All pixels within the cube are assumed to be background pixels and then set to 1's while other pixels are set to 0's. With this information, a binary object detection mask is created. The background initially is defined as the location of all 1's connected to the first sampling pixel while the object is defined as the largest group of 0's.

For a background with more than one colour, multiple cubes are created instead of one cube. If a pixel does not fall into the previously created cube, the L1 distance is calculated to the nearest cube. If the distance is less than a tolerance value (tol), the cube is expanded to contain that pixel. Otherwise, a new cube is created. If the tolerance value is set low, many cubes will be created.

In the experiment, pixels within 5 percent of width and height of the edge around an image become a set of training points for the background.

3.2.1 Region Growing Segmentation Experiments

Segmentation is first tested with synthetic images with tolerance threshold at 0.1. Figure 3.9 shows three original images on the left side and images after segmentation on the right side.

In Figure 3.9, 1(a) and 2(a) is a blue cross on a white background and on a green textured background respectively. The segmentation is perfect (the black area in all (b) images is the correct background area). 3(a) demonstrates a green object on a two pattern background. Here the segmentation can extract the object from textured background. However, there are a few pixels around the edge of object in 3(b) which have the colours of the background and are residual background pixels.

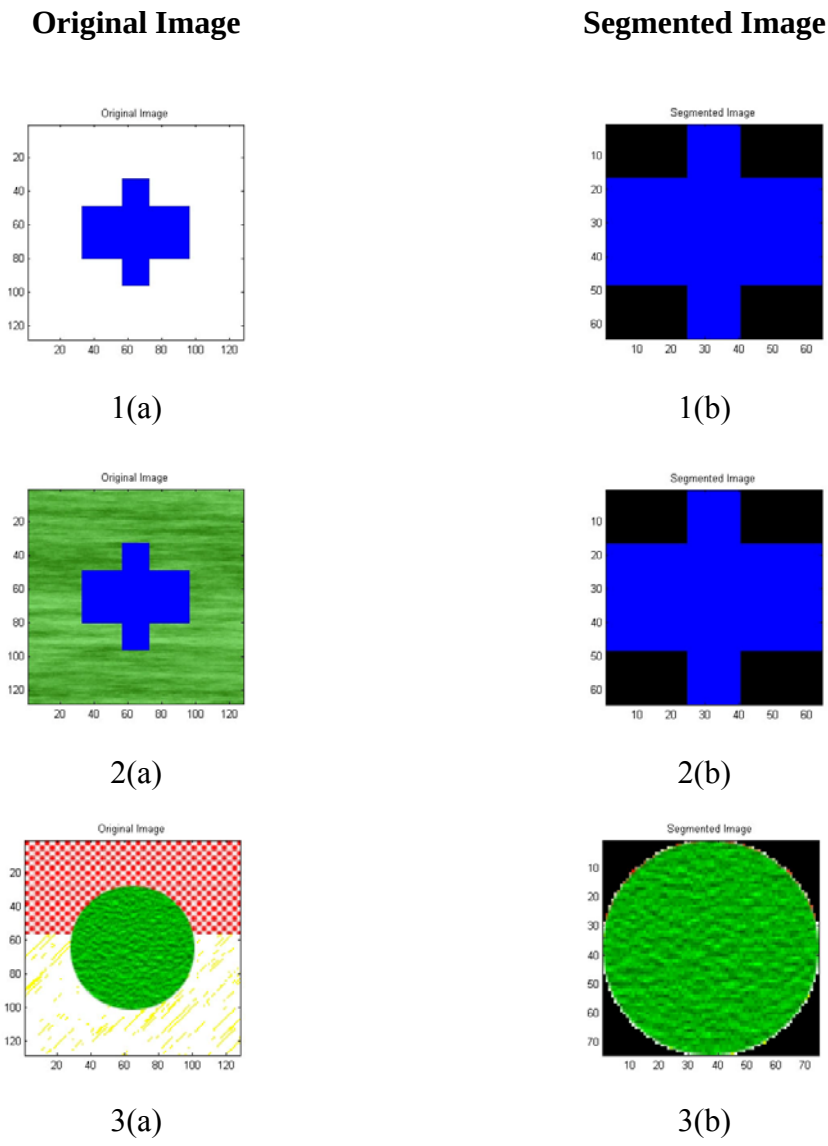


Figure 3.9 Some image examples before and after segmentation

As the segmentation result is reasonable on the synthetic images, a real image dataset is used next. The results are shown in Figure 3.10 with three different tolerance threshold (tol) settings.

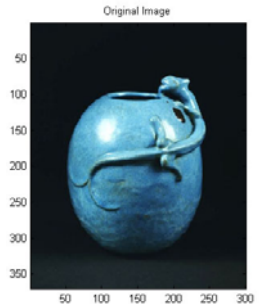
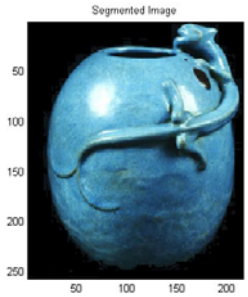
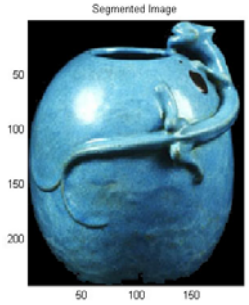
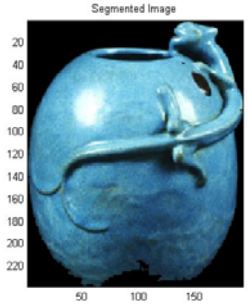
Original Image	Tol=0.05	Tol=0.10	Tol=0.20
			
1(A)	1(B)	1(C)	1(D)
			
2(A)	2(B)	2(C)	2(D)

Figure 3.10 Some image examples before and after segmentation

In the first image (1(A)-(D)), the background is removed but the shadow of object is left. The segmentation returns a better result as we increase the tolerance value. However, there are some pixels eliminated from the part of the object when the tolerance value is high. In image 2(C) and 2(D), more pixels at the bottom of the object are removed when the tolerance value is increased.

The segmentation result varies when tol is varied. To get a perfect segmentation an adjustable threshold or threshold optimization would be required.

Another important consideration for image retrieval is the processing time per image. Table 3.1 shows the computing time of the normalized cut segmentation by the intervening contour and the region growing segmentation on the image 2(A) in Figure 3.10.

Methods \ Image Size (pixels)	**100 x 123**	**150 x 185**
Normalized Cut Segmentation	6.49 sec	31.56 sec
Region growing Segmentation	0.28 sec	0.57 sec

Table 3.1 Computing times in second unit from two techniques

It can be seen that the normalized cut segmentation consumes substantially more computing time than the region growing segmentation. For this reason and the fact that many parameters must be set and the number of regions must be known, the normalized cut segmentation is not used in the rest of the experiments.

3.3 Image Retrieval with a Small Collection

The aim of this experiment is to investigate the effect of background removal on image retrieval. A set of images with similar objects on different backgrounds was created for the experiment. Three different methods were compared. The first method extracted features from a whole image (Noseg method). The second method extracted features after background removal using the region growing segmentation (Mseg segmentation) and the last method extracted features from the central area of the image derived from dividing the image equally into nine rectangular regions. We referred to the last method as “the grid segmentation” or Gseg segmentation method.

Figure 3.11 The 8 objects in 3 different backgrounds

3.3.1 Dataset

There are 24 images created from 8 objects with 3 different backgrounds (see Figure 3.11). Each image is 250 x 166 pixels. The images are captured with small changes in viewing angle.

3.3.2 Feature Extraction

We considered four features- colour histogram, colour coherence vectors, Hu moments and Gabor filters - which are extracted from each image.

Colour Histogram and Colour Coherence Vectors

In this experiment, the RGB colour space was divided into 64 cells (4x4x4) and the histogram gave the fraction of the total number of pixels falling in each cell.

The CCV technique is a colour-based method that takes spatial information into account. Pixels in an image are separated to be coherent or not coherent pixels. Colour coherent pixels are part of some sizable contiguous colour region, while incoherent pixels are not (Pass, Zabih et al. 1996). A pixel is coherent if the size of its connected colour region exceeds a fixed value threshold; otherwise, the pixel is incoherent.

A colour coherence vector *(G)* is generated and used to match with other images. The vector is a composite of the colour histogram for the coherent pixels and a colour histogram for the incoherent pixels. The CCV technique avoids matching between coherent pixels in one image and incoherent pixels in another image.

Let the number of coherent pixels of colour i be α_i and the number of incoherent pixels β_i. The colour coherence vector consists of $\langle(\alpha_i, \beta_i),...,(\alpha_n, \beta_n)\rangle$ where (α_i, β_i) is the coherence pair for colour i.

To compare the CCV feature distance between two images, we used absolute differences. The absolute value is described by the following equation.

$$\Delta G = \sum_{i=1}^{n} \left| (\alpha_i - \alpha'_i) \right| + \left| (\beta_i - \beta'_i) \right|$$

When α_i and β_i are coherent and incoherent vectors of the image *I*; α'_i and β_i' are coherent and incoherent vectors of the image I'.

In the experiment, the 64 bin histogram was generated for coherence and incoherence and matched separately. The percentage of pixels threshold to separate coherent and incoherent regions is 1 percent of the total number of pixels. In other words, the fixed value threshold equals 164 pixels from the image size 16384 pixels (128 x 128). Only regions larger than 164 pixels were regarded as coherent.

Hu Moments

Hu Moments are moment invariants which can be used as a set of image descriptors. Hu (Hu 1962) proposed this series of moments as an invariant binary shape representation to classify handwritten characters.

Central moment μ_{pq} are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_C)^p (y - y_c)^q f(x, y) dx dy$$

And the normalized moments ν_{pq} are defined as $\nu_{pq} = \dfrac{\mu_{pq}}{\mu_{00}^{\omega}}$

where the coordinates (x_c, y_c) denote the centroid of *f(x, y)* and $\omega = (p + q + 2)/2$

The seven Hu rotation, translation and scale invariant moment functions are calculated using these normalised central moments as follows:

$$
\begin{aligned}
\phi_1 &= \mu_{20} + \mu_{02}, \\
\phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, \\
\phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2, \\
\phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2, \\
\phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + \\
&\quad (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2), \\
\phi_6 &= (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}), \\
\phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) - \\
&\quad (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2).
\end{aligned}
$$

Gabor Filters

Gabor filters are used widely in the field of computer vision and pattern recognition for example in the research and application of recognition, texture segmentation (Shioyama, Wu et al. 1999) and identification of fingerprints (Jain and Pankanti 2000).

Gabor filters are applied to extract texture features because they are efficient in reducing image redundancy and are robust to noise. A region around a pixel is described by the responses of a set of Gabor filters of different frequencies and orientations.

An input image *I(x,y)* is convolved with a two-dimensional Gabor function *g(x,y)* to get a Gabor feature image *r(x,y)* as follows.

$$r(x, y) = \iint I(\xi, \eta) g(x - \xi, y - \eta) d\xi d\eta$$

The following family of Gabor functions is used:

$$g_{\lambda,\Theta,\varphi}(x, y) = e^{-\frac{(x'^2 + \gamma^2 y'^2)}{2\sigma^2}} \cos(2\pi \frac{x'}{\lambda} + \varphi)$$

$$x' = x\cos\Theta + y\sin\Theta$$

$$y' = -x\sin\Theta + y\cos\Theta$$

λ is the wavelength of the cosine factor and $\frac{1}{\lambda}$ is the spatial frequency of the harmonic factor. Θ is the orientation of the normal to the parallel stripes of a Gabor function. The phase offset φ is specified in degrees. σ is the standard deviation of the Gaussian factor of the Gabor function and γ is called the spatial aspect ratio.

An image's red, green, and blue channels are filtered separately; alternatively the RGB image is converted to the gray scale image (0-255) before calculation. The collection of the Gabor filter responses is achieved by iteration over angles and frequencies, producing a response vector the length of the product of the sum of angles and the sum of frequencies. It will also generate filtered images for each angle-frequency combination. Using many different orientations and scales ensures invariance; objects can be recognized at various different orientations, scales and translations. In the experiment, the orientation value Θ is set at 4.0 and frequency value is 6.0.

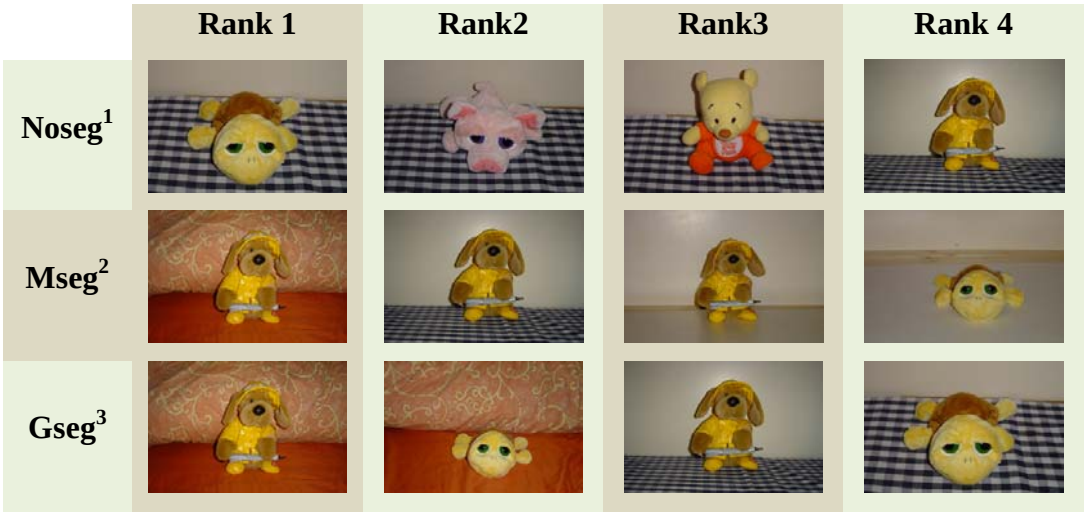
The above four features were extracted from all images in the database collection and stored in separate files. The similarity was compared using the Euclidian distance.

3.3.3 Experiments

Images with different background from images in the database were used as the query to search the database. Figure 3.12 shows the four best matches using the RGB histogram of a query which is not in the image database.

The result shows an example of how the region growing segmentation produces better results (The matching images are ranked 1, 2 and 3) when compared with the whole image and grid segmentation for RGB based retrieval. However, segmentation by region growing does not always give better retrieval rank.

Query: 



Note: the query is not in the dataset

1: Noseg = Without Segmentation (whole image)

2: Mseg = Region growing segmentation

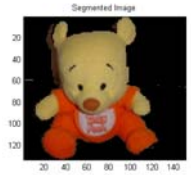
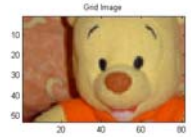
3: Gseg = Grid segmentation

Figure 3.12 The retrieval by the query image

In Figure 3.13, the results are shown for retrieval using one of the images from the database as a query and the RGB colour histogram for matching. All three images are returned within the top 5 by the grid segmentation and RGB histogram retrieval while the last image retrieved by region growing segmentation is on rank 7. Examining the regions after segmentation shows that the segmentation by Gseg gets enough important parts of the object to retrieve all three images within the top 5 positions.

Query: 

Mode \ Image			
Noseg[1]	Rank 1	Rank 11	Rank 13
Mseg[2]	Rank 1	Rank 7	Rank 3
Gseg[3]	Rank 1	Rank 3	Rank 4

Mode	**Regions after segmentation**		
Mseg[2]			
Gseg[3]			

Note: the query is an image in the dataset

1: Noseg = Without Segmentation (whole image)

2: Mseg = Region growing segmentation

3: Gseg = Grid segmentation

Figure 3.13 Results of retrieval with and without segmentation

3.3.4 Results

Table 3.2 shows the average precision (see Section 2.4.2) computed from all images in this collection using different features and different approaches to segmenting out the background. The Mseg segmentation outperforms all the Noseg and Gseg cases except for Gabor filter features with the Gseg segmentation which has a slightly higher precision than the Mseg segmentation. Both the Mseg segmentation and the Gseg segmentation have higher precision than Noseg on all features.

	Noseg	**Mseg**	**Gseg**
RGB Histogram	0.356	0.809	0.732
CCV	0.403	0.800	0.733
Hu Moments	0.269	0.322	0.289
Gabor Filters	0.184	0.241	0.257

Table 3.2 Average precision of different features

For this dataset, the RGB histogram and CCV are the features that for all three methods give better precision than Hu moments and Gabor filters. Using the CCV is slightly better than using the RGB histogram on average.

3.4 Summary

In this chapter, three segmentation methods are introduced as a way of trying to reduce the effect of background on image retrieval tasks. They are the normalized cut segmentation, the region growing segmentation and the grid segmentation. Since segmentation is the preprocessing step before retrieval and the efficiency of retrieval can depend on how well the segmentation is performed, it is helpful to understand the segmentation and find ways to improve the results of this step.

From experiments with the normalized cut segmentation, using the edge information features in the segmentation is better than using colour and intensity. However, it may be possible to achieve better segmentation results by combining several features

together and this has not yet been explored. Another way to improve segmentation is by parameter adjustment. Several experiments would be required to find the best parameters and this is not a desirable feature for automatic techniques. To use automatic segmentation, the question of how many segments to produce has to be solved for the normalized cut approach. Moreover, the cut needs to be more reliable and closer to the exact shape of objects. Furthermore, because of the computational complexity of the algorithm, the normalized cuts algorithm consumes significant computing time.

Region growing segmentation on uncluttered backgrounds returns good results within a short period. The grid segmentation, which obtains the features from the central area, also returns good results in a short period because the extracted area has much of the significant information from the object of interest and the method to find the area is simple.

The last section in this chapter shows an image retrieval experiment on a small dataset. It demonstrates that the region growing and grid segmentation approaches give better retrieval performance than without segmentation and the region growing approach performs slightly better than the grid segmentation method.

In summary, this object-based image retrieval retrieves similar object images from a database based on the appearance of the physical objects in those images. The principle of only using data from the object is demonstrated to be better than using data from the whole images.

From experiments on our image collection, it can be noticed that the image retrieval system is successful for objects that can be easily separated from the background and that have distinctive features of objects in those images. In addition, the performance of a CBIR system is affected by the results of object extraction by image segmentation.

In some situations, our segmentation method, which is based on colour features may have difficulty in segmenting an object. Figure 3.14 shows one of the image examples segmented by the region growing approach described in Section 3.2. The major part of

the object after segmentation is lost because the background growing expands to include the object area. These kinds of images cannot be segmented well with only the colour feature. The segmentation may achieve better results if it includes information from other features that make the background and object distinguishable.

Figure 3.14 An original object image (left) and after segmentation (right)

Automatic image segmentation is a hard task to achieve satisfactory results where object and background are difficult to distinguish. Furthermore, it will become more complex to select appropriate features and set adaptable parameters for a wide variety of images. To avoid the need for potentially unreliable segmentation, people have recently explored other methods for effective CBIR.

As we are interested in retrieval where users are particularly interested in searching for specific object in images, we move forward to study further about local characteristics to form the image representation for content-based image retrieval. In the next chapter, CBIR systems using salient regions-based representations will be investigated.

Chapter 4 Image Retrieval using Salient Regions

Generally, the features which are used to match images are computed from the whole image. However, global features cannot deal successfully with similar objects on substantially different backgrounds.

In the last chapter, CBIR used information from the object only to retrieve images containing the object which were similar to an object query image. From this approach, image segmentation became an important task to segment the object from the background. Although segmentation worked reasonably for much of the image collection in the previous experiments, it faced problems when dealing with images which were noisy and complicated. Unsatisfactory segmentation results can cause CBIR to fail badly.

Without relying on segmentation, one of the alternative approaches is to identify salient regions as the image representation. To solve the problem, one approach is to attempt to detect “interesting” regions. Local features around salient areas are calculated to describe the image.

Salient points or salient regions became popular as local descriptors relatively recently and have been suggested for use in image classification, image recognition and content-based retrieval to locate the important points. Some research papers such as (Sebe, Tian

et al. 2002) and (Hare and Lewis 2004) have shown that local features from salient regions provide good results for CBIR.

For comparing images, this approach works by comparing features around salient points. Features around salient points of the similar objects are frequently more similar than for different objects. Therefore, CBIR based on salient regions may be useful for our problem. Moreover, CBIR by salient regions can give the benefits of dealing with more complex background which helps CBIR to avoid problems which may happen when using segmentation.

This chapter starts by introducing a way to find the salient points in an image. A new method to match salient regions is proposed. Retrieval of images using salient regions is investigated and compared with the segmentation approach. Results of image retrieval performance are described.

4.1 Salient Point Detection by Difference-of-Gaussian

4.1.1 Detection of Scale-space Extrema

There are several methods to find salient points. One of them is the idea of using peaks in a Difference-of-Gaussian pyramid proposed by Lowe (Lowe 1999). A function L is produced from a variable scale Gaussian, $G(x, y, \sigma)$ and the image $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where $*$ means the convolution operation in x and y, and the 2D Gaussian kernel is

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

The Difference-of-Gaussian function $D(x,y, \sigma)$ of the convoluted image $L(x, y, \sigma)$ is the subtraction of two adjacent scales separated by a constant scaling factor k:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

The intervals in scale-space are sampled by increasing the scale parameter σ by a constant amount. In Lowe's paper (Lowe 2004), the image is down sampled by a factor of two when the scale parameter σ doubles. Therefore, the later octave has half the dimensions of the previous one. Figure 4.1 shows an approach to the construction of $D(x, y, \sigma)$. For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images separated by a constant factor k, shown on the left column. Each octave of scale space (i.e., doubling of σ) is divided into an integer number, s, of intervals, so k equals $2^{1/s}$. In order that the final extrema detection covers a complete octave, $s + 3$ images in the stack of blurred images must be required per octave.

Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images shown in the right column of Figure 4.1. Once a complete octave has been processed, we resample the Gaussian image that has twice the initial value of σ (it will be 2 images from the top of the stack) by taking every second pixel in each row and column. The accuracy of sampling relative to σ is no different than for the start of the previous octave, while computation is reduced considerably.

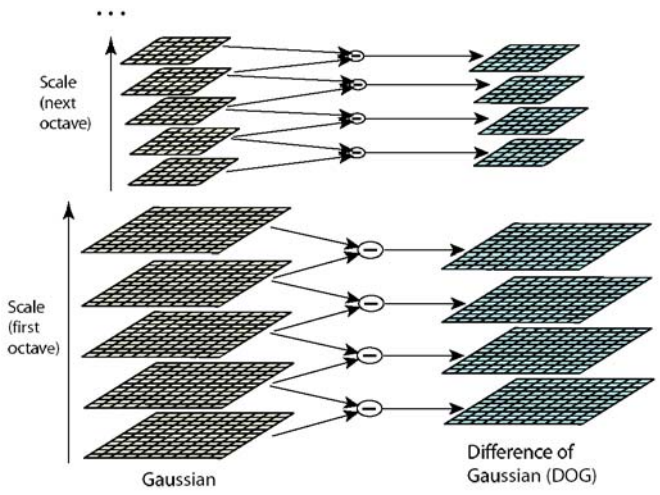


Figure 4.1 Difference-of-Gaussian (DOG) at different scales (Lowe 2004)

To detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbours in the current image and nine neighbours in the scale above and below. It is selected only if it is a minimum or a maximum of the scale-space

neighbours. Figure 4.2 shows a comparison of a pixel to its 26 neighbours in 3x3 regions at the current and adjacent scales.

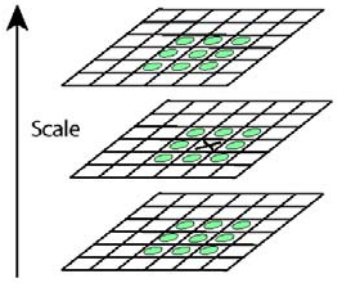


Figure 4.2 Comparison between a pixel (marked with X) and its 26 neighbours (marked with circles) to find maxima and minima of the Difference-of-Gaussian images. (Lowe 2004)

4.1.2 Keypoint Localisation

The first item to be checked for stability of keypoints is contrast. The contrast threshold constant is used to determine the stability to noise. The values of $D(x,y,\sigma)$ at the keypoint location which is less than a contrast threshold are discarded as unstable and susceptible to noise.

In the next step, keypoints that are located along edges are rejected because these keypoints are defined poorly and are likely to be susceptible to small amount of noise. Keypoints that are along the edge have a small curvature while keypoints that are across the edge have a large curvature.

The elimination of these points can be achieved using the ratio of the principle curvatures. The ratios that are too large will be rejected. The principle curvatures can be found by computing the Hessian matrix, **H** at the keypoint location.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Let α be the eigenvalue with the largest magnitude and β be the smaller one. The trace Tr(H) and determinant Det(H) is calculated to find the ratio of the principle curvatures.

$$\mathrm{Tr(H)} = D_{xx} + D_{yy} = \alpha + \beta$$
$$\mathrm{Det(H)} = D_{xx}D_{yy} - D_{xy}^2 = \alpha\beta$$

Let r be the ratio between the largest magnitude eigenvalue and the smaller one, so that $\alpha = r\beta$

$$\frac{\mathrm{Tr(H)}^2}{\mathrm{Det(H)}} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

Since $\frac{(r+1)^2}{r}$ is at a minimum when the two eigenvalues are equal and it increases with r. To check that the ratio of principal curvatures is below some threshold, r, we check only

$$\frac{\mathrm{Tr(H)}^2}{\mathrm{Det(H)}} < \frac{(r+1)^2}{r}$$

In Lowe's paper (Lowe 2004), keypoints with ratios greater than $r = 10$ are rejected.

4.2 Experiments

4.2.1 Salient Point Detection

All Images were changed to gray scale before detecting keypoints. The algorithm to find keypoints by peaks of the Difference-of-Gaussian was applied to the dataset described in Chapter 3 (see the details in Section 3.3.1). In this experiment, parameters were set as for the experiment in (Lowe 2004).

For all images, up to fifty points were extracted which was a reasonable number to represent images according to (Tian, Sebe et al. 2001). Figure 4.3 shows some images with keypoints and salient regions. The first column shows the original images. The

yellow masks in the images of the second columns are salient points. Salient regions in the last column are represented by a number of black squares.

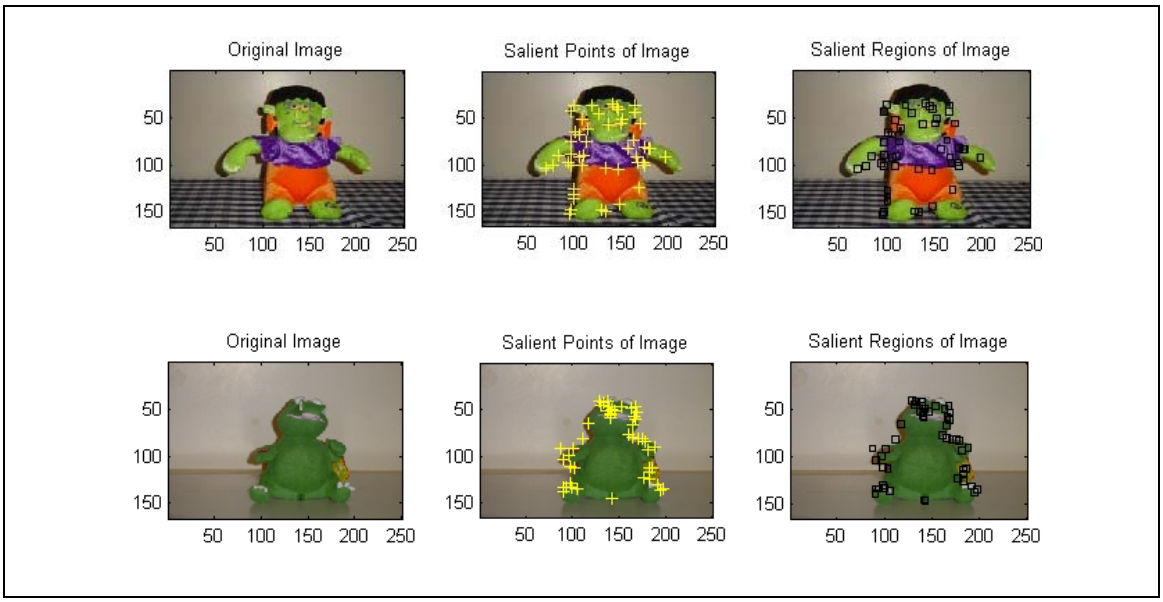


Figure 4.3 Two examples of salient regions around salient points

4.2.2 Local Features and Indexing

After keypoints were located, the rectangular regions size (2*d*+1) x (2*d*+1) pixels were created around the keypoints (where *d* is the distance from the centroid of the keypoint). The number of regions depends on the number of points that are found in each image. In this experiment, the region has a size 7 x 7 when $d = 3$. If any keypoints are detected far from the four image borders less than distance *d*, the regions size of those keypoints would be smaller than the normal size. The reason is the size cannot create the normal region size (2*d*+1) x (2*d*+1) around keypoints.

In the feature extraction step for images in the database, features in all regions around all keypoints in all images were extracted. Features from the query image were also extracted from all regions and compared with other images in the database.

Matching Algorithm

Generally, the number of keypoints used in each image has been made equal because it is then easy to compare one by one. However, in some cases controlling the number of keypoints in each image to be equal is hard to do. For example, in some uncluttered images only a few keypoints can be detected while some cluttered images generate many keypoints. To deal with this case, a new approach to matching for different numbers of regions is proposed.

Let image A and image B have sets of salient regions: $R^A = \{R_1^A, R_{2,\ldots,}^A R_M^A\}$ and $R^B = \{R_1^B, R_{2,\ldots,}^B R_N^B\}$ respectively where M is the number of salient regions in image A and N is the number of salient regions in image B.

Given $Fv(R_i)$ as a d dimensional feature vector of a region R_i and $\delta(R_i, R_j)$ is a similarity distance between $Fv(R_i)$ and $Fv(R_j)$, then

$$\delta(R_i, R_j) = \left\| Fv(R_i) - Fv(R_j) \right\|_2 = \sqrt{\sum_{k=1}^{d} [Fv(R_i)_k - Fv(R_j)_k]^2}$$

The algorithm to match between regions is as follows:

Let

Q = min (M, N)

For k =1 to Q

{

$P(A,B)_k$ = min(δ(*,*))

Remove $\delta(i,*)$ and $\delta(*,j)$ i, j: index when $\delta(i,j)$ is the minimum value

Note: * = All

}

Assign C_{AB} as the number of matching regions between image A and image B. The pair $P(A,B)_k$ is counted as matching if $P(A,B)_k$ is less than or equal the matching_threshold (μ). In this experiment, μ = 1.0.

$$C_{AB} = Numberof\ \{\forall_i P(A,B)_i \mid P(A,B)_i \le \mu\}$$

Define S_{AB} is a score of matching between image A and B .

$$S_{AB} = \{\sum_{i=1}^{C_{AB}} P(A,B)_i \mid P(A,B)_i \le \mu\}$$

For matching steps of a query image U and n images V_i in the database ($i \in \{1,..,n\}$), the above parameters are calculated. The retrieve rank is first ordered descending by C_{UV_i} and sorted ascending by S_{UV_i} if any images have the number of matching (C) equally.

It is noted that the maximum number of keypoints used in each image is set to be fifty points which follows the experiments which have been done in (Tian, Sebe et al. 2001). Their experiment showed that the improvement in accuracy did not justify the computational effort involved when using more than fifty points.

4.3 Results

In this experiment, fixed size salient regions are used as described above and to compare with the results of the retrieval experiments in chapter 3 various feature vectors are extracted from each of the salient regions.

Figure 4.4 shows the result of image retrieval using the CCV from each of the salient regions compared with the use of CCV in other methods. In this example the salient regions approach retrieves all three images with higher rank than using the whole image and also at higher rank than with grid based segmentation. However, in this instance the salient region method does not outperform retrieval by the region growing segmentation with CCV.

Query: 

	Rank 1	Rank 2	Rank 3	Rank 4
Noseg[1]				
Mseg[2]				
Gseg[3]				
DoG Region[4]				

Note: the query is not in the dataset

1: Noseg = Without segmentation (whole image)

2: Mseg = Region growing segmentation

3: Gseg = Grid segmentation

4: DoG Region = Regions by Difference-of-Gaussian

Figure 4.4 The retrieval by the query image

The experiments in Section 3.3 are repeated using the salient regions from the DoG rather than the regions resulting from central region segmentation. The average precision is calculated for retrievals using each of the images in the database in turn as the query. Table 4.1 shows the average precision (see 2.4.1) for this collection using each of the different features for the different segmentation approaches compared with

the DoG salient region approach. The salient region approach performs better than using the whole image and when using Gabor filters it outperforms the other two segmentation techniques. However, on the whole, best results are obtained with the region growing segmentation (Mseg) approach and the colour histogram or CCV features.

	Noseg	**Mseg**	**Gseg**	**DoG Region**
RGB Histogram	0.356	0.809	0.732	0.585
CCV	0.403	0.800	0.733	0.585
Hu Moments	0.269	0.322	0.289	0.233
Gabor Filters	0.184	0.241	0.257	0.316

Table 4.1 Average precision of different features

In this part, salient regions have been applied to image retrieval. Salient regions are generated as the patch areas around salient points. With the new matching method proposed, feature vectors extracted from each region are compared and the images are ranked. The results tend to support the experiments from (Sebe, Tian et al. 2002) and (Hare and Lewis 2004) that the image retrieval by local features in salient regions has a potential to provide better retrieval when compared with the global features from the whole image. However, it is important to note that when trying to retrieve object images, the background may well be contributing salient regions which confuse the object matching.

4.4 Image Retrieval from a Synthetic Database

In this section, we investigated image retrieval with and without preprocessing (segmentation) using a synthetic image database.

4.4.1 The Synthetic Dataset

There are 72 images created from four catalogues with a combination of varied characteristics. Each image measures 128 x 128 pixels. Table 4.2 shows the

characteristics to make the combinations. Figure 4.5 shows 18 of the 72 images in the Blue-Cross catalogue.

Catalogue	Rotate (degree)	Background	Position
A (Blue-Cross)	0	White (ffffff)	Central
B (Red-Cross)	45	Green (00ff00)	Up-Right
C (Blue-Step)		Texture No.1	Down-Left
D (Red-Step)			

Table 4.2 The characteristics of images

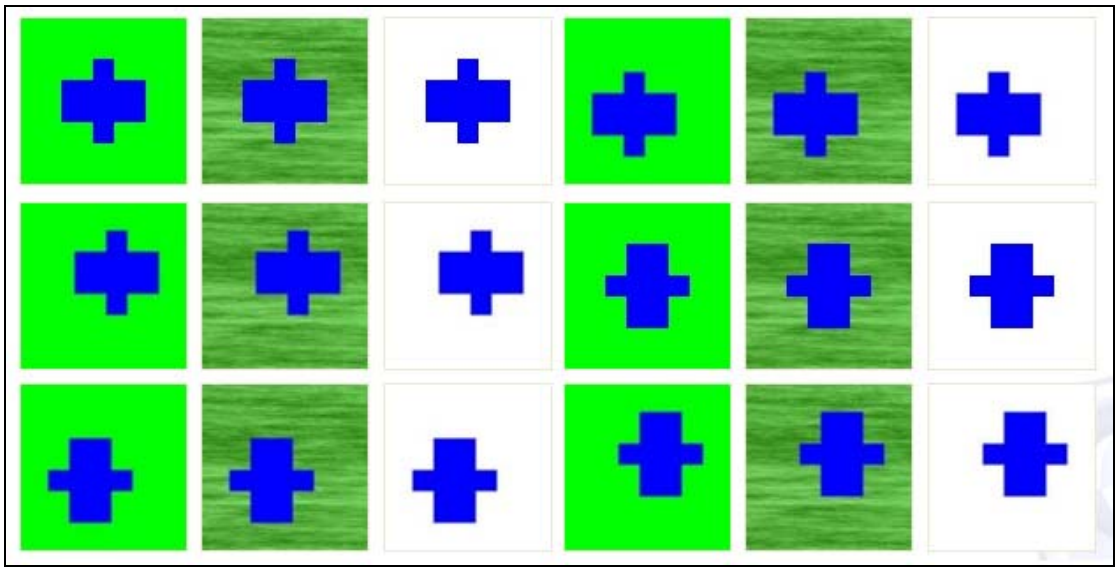


Figure 4.5 The Blue-Cross Catalogue

4.4.2 Retrieval Methods and Feature Vectors

Four retrieval methods were compared. One was using the whole image with no preprocessing (NOSEG). The second type used the region growing segmentation as the preprocessing step (MSEG). The third used rectangular grid segmentation as the preprocessing step (GSEG) which the image was divided into 9 equal rectangles and the central region was used. The fourth method (DOG) used salient regions detected using the Difference-of-Gaussian pyramid and each of the different feature vector types was then used to represent the salient regions detected.

The use of four different feature vectors was compared; RGB Histogram, CCV, Hu moments and Gabor filters.

4.4.3 The Experimental Procedure

Taking each category at a time, each of the images in the category was used as the query image and the total database of 72 synthetic images was searched. This procedure was repeated for each of the four feature vectors considered and for each of the four retrieval methods making 16 retrieval experiments for each of the four search catalogues.

4.4.4 Results

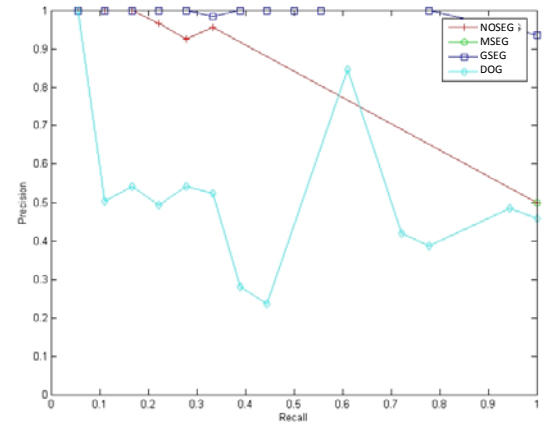
For each of the four data catalogues and each feature vector type a precision-recall graph was produced comparing the four retrieval methods. The trends for RGB histogram and CCV were very similar so only the results for CCV were considered below. The graphs for CCV are shown in Figure 4.6. For Hu moments they are shown in Figure 4.9 and for Gabor filters they are shown in Figure 4.10.

Retrieval Using Coherence Vector (CCV) Features

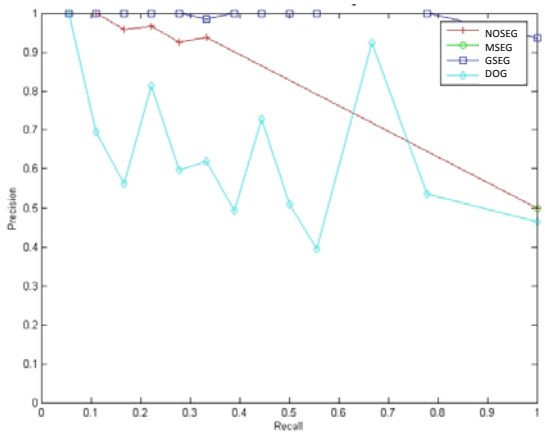
The RGB histogram and CCV gave similar trends in the precision-recall graphs so only the results for CCV are presented here. From Figure 4.6, it can be seen that retrieval using the local features extracted from salient regions does not outperform the other approaches in catalogues A, B and D. On the other hand, the retrieval from the central

piece given from grid segmentation performs well on catalogue A and B but cannot beat the precision from the whole image in catalogue C and D after recall around 0.25.

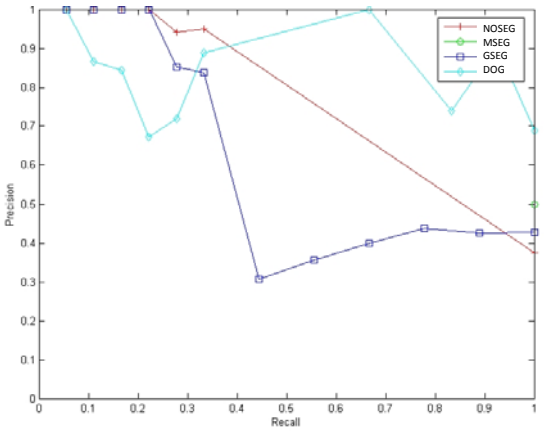
It is possible that using a central piece (GSEG) of each image can return the highest precision in the catalogue A and B because the central rectangular area contains much more information; enough to represent the object in the image.



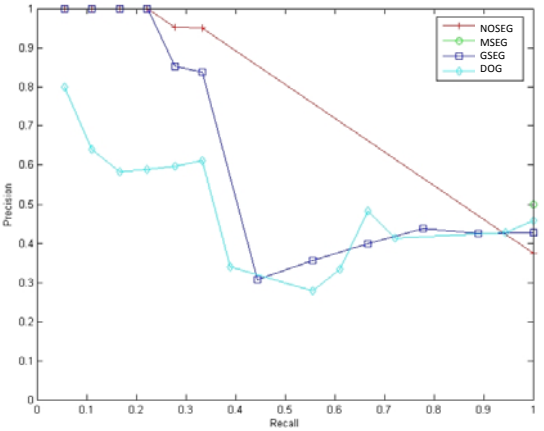
(a) Catalogue A



(b) Catalogue B



(c) Catalogue C



(d) Catalogue D

Figure 4.6 (a-d) Graphs comparison of 4 segmentation methods using CCV features

Figure 4.7 shows the retrieval rank by the region growing segmentation method from two different objects. The format of ranking appearing in figure is (rank:name of image). The query is represented on the top left of the figure. The 4 lines of objects are ordered by the similarity distance of features between the query and images in the

database. The nearest distance is in the first rank and increasing distance is ordered left to right and then top to bottom. Any images that have the difference of distance not more than the threshold value of ranking are given the same rank. In this experiment, the threshold value is 0.001.

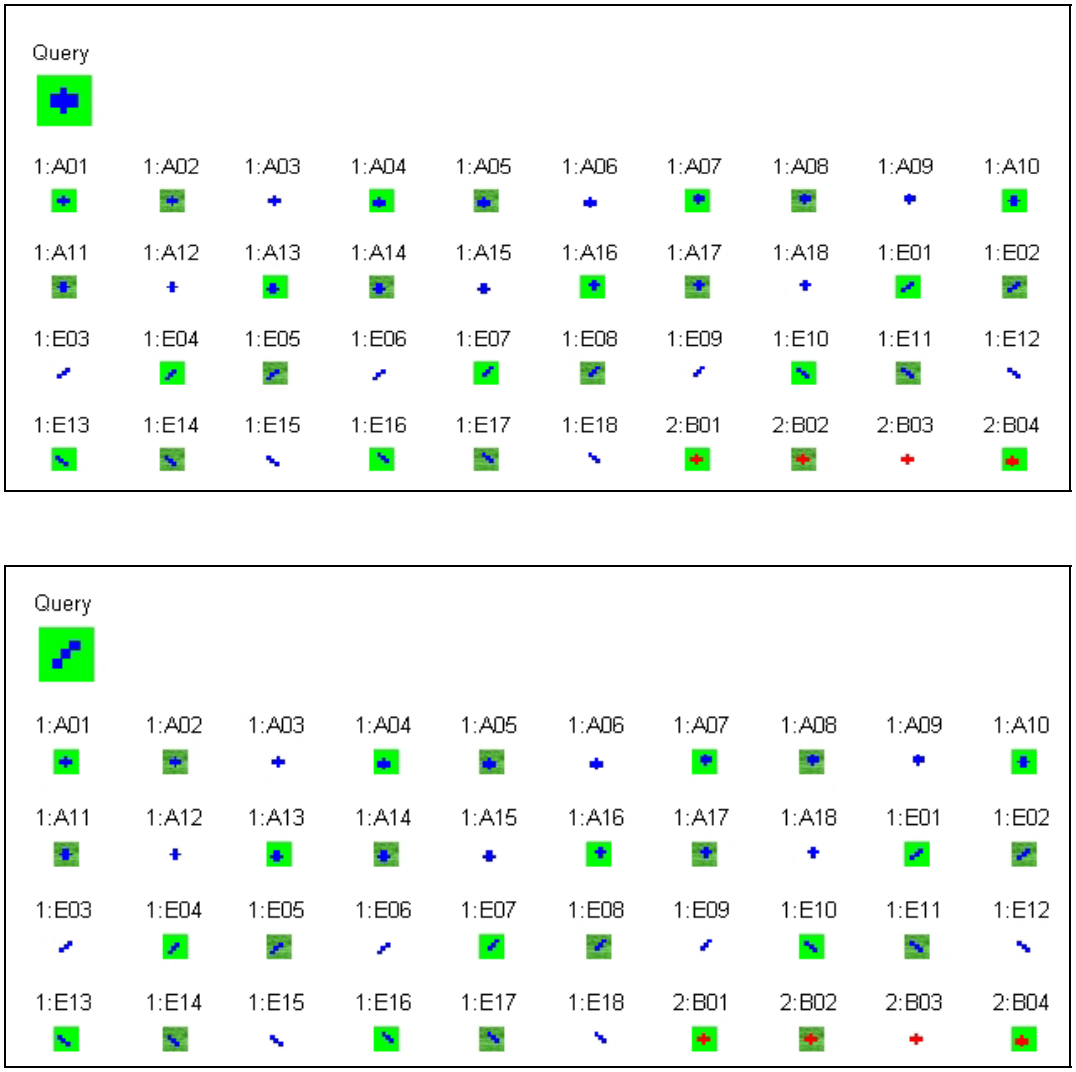


Figure 4.7 The top 40 images from two different queries but the retrieval results are all identical.

In this case, there are 36 images that have the nearest distance for which the difference of distance is less than the threshold value of ranking, so all of them are arranged in the first rank. This is an interesting case to be noted about the object segmentation and the colour features on this dataset.

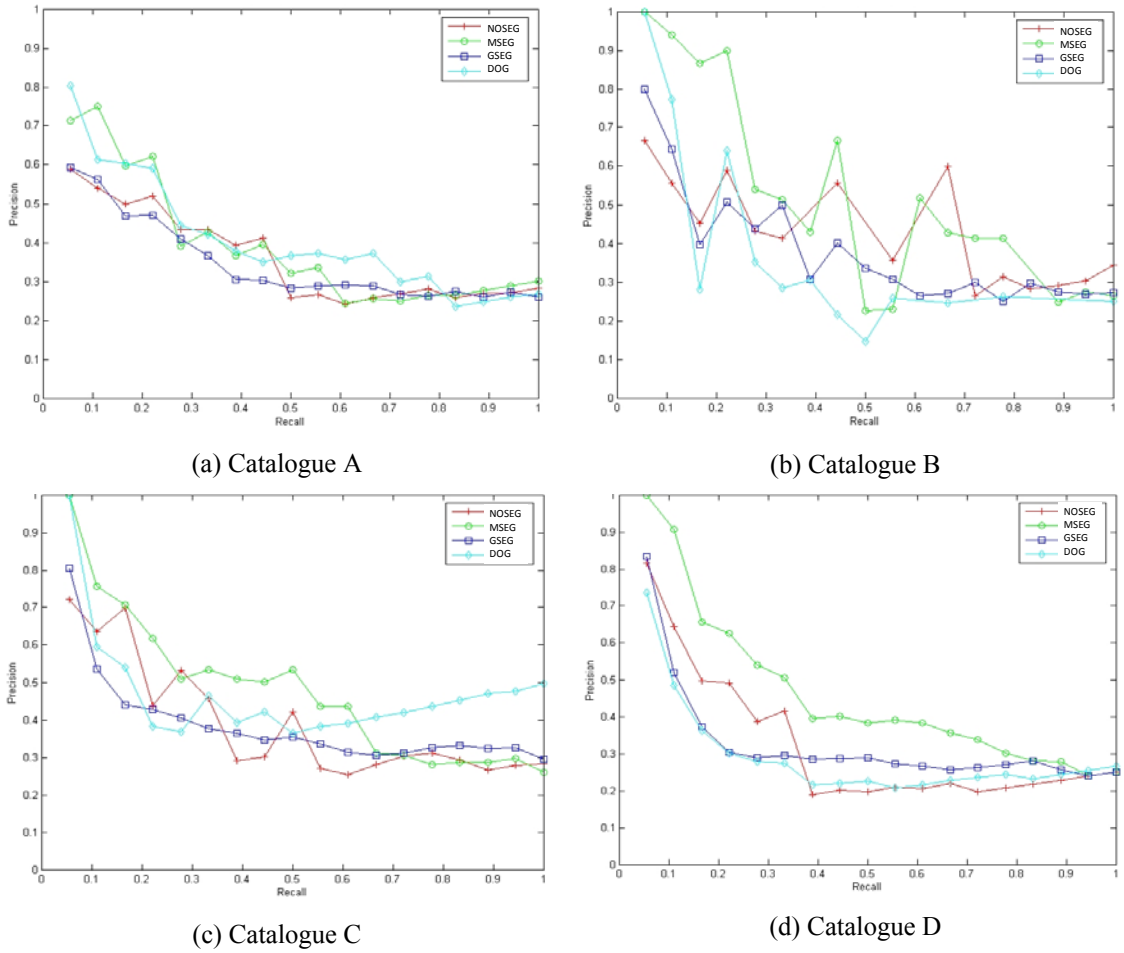
Figure 4.8 Pieces of objects from two images which are segmented by the region growing segmentation

With the perfect segmentation of the region growing segmentation on this collection, a problem appears when two objects have the same colour. Figure 4.8 shows two objects different in shape but similar in colour. The colour features cannot separate these two objects. Hence both have retrieved 36 images in the same rank (see Figure 4.7).

In precision-recall graphs, if there are some images which get the same rank, the precision is calculated at the last position of the image that has the same rank. So the line of the region growing segmentation (MSEG) in Figure 4.6 shows the precision only at recall = 1.

Retrieval Using Hu Moments Features

Figure 4.9 presents the four average precision-recall graphs for each catalogue. By Hu moments feature, the results by all methods are quite similar for catalogue A. The region growing method retrieves the best rank in catalogue D. The DoG method retrieves the lower precision than other methods in catalogue B and D but it outperforms in catalogue C after recall 0.5.



(a) Catalogue A

(b) Catalogue B

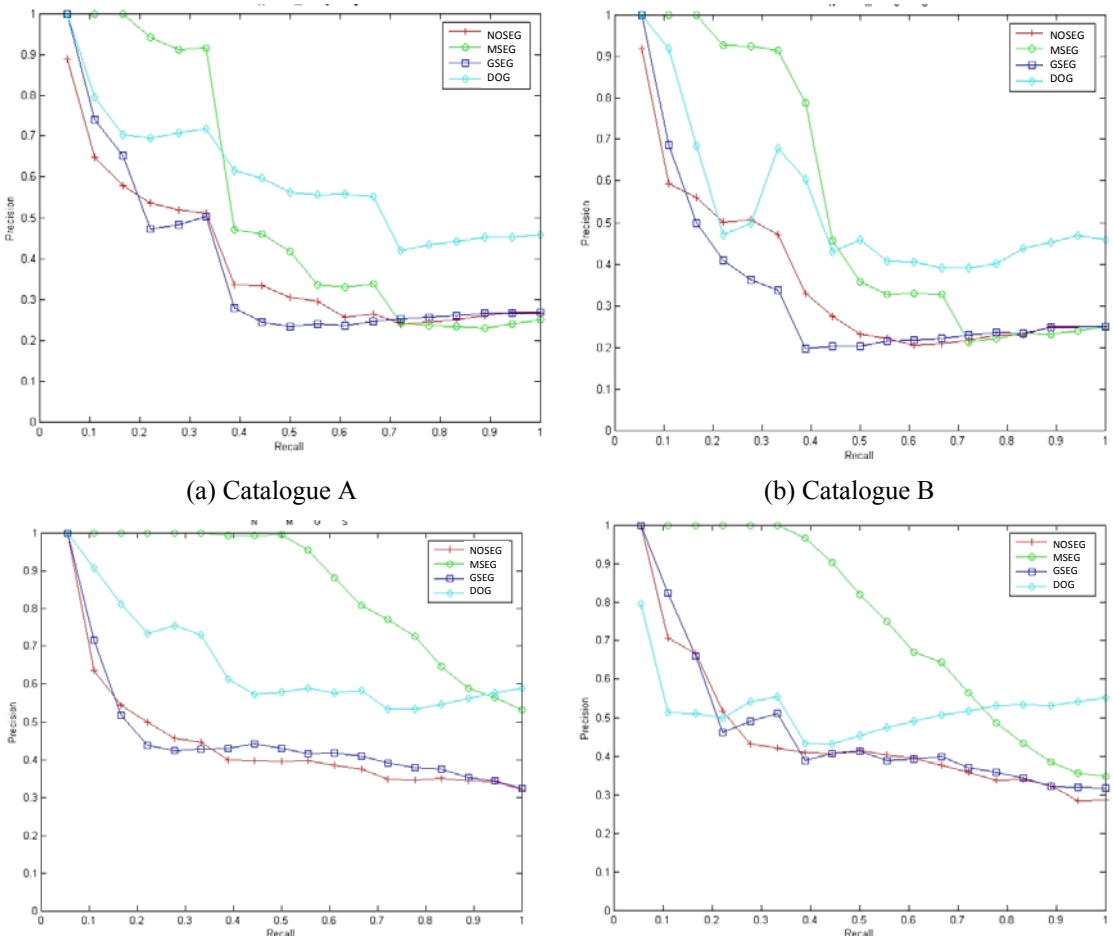
(c) Catalogue C

(d) Catalogue D

Figure 4.9 (a)-(d) Graphs comparison of 4 methods by Hu moments

Retrieval Using Gabor Filters Features

The region growing segmentation retrieves relevant images in the higher rank than other methods especially in the early recall up to around 0.4 in catalogue A; nearly 0.45 in catalogue B; 0.95 in catalogue C and around 0.75 in catalogue D. For the higher recall value in each catalogue, the salient region method which gets the second higher rank becomes the highest precision.



(a) Catalogue A (b) Catalogue B

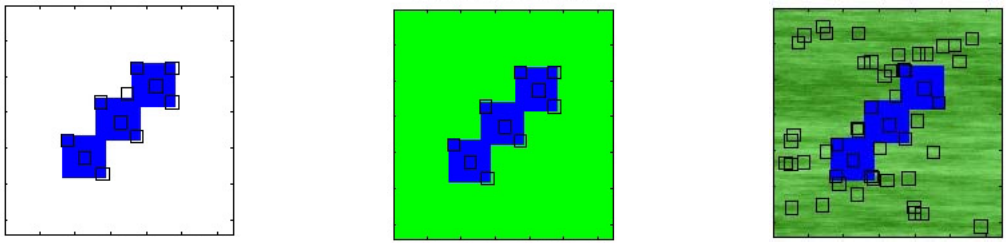
(c) Catalogue C (d) Catalogue D

Figure 4.10 (a)-(d) Graphs comparison of 4 methods by Gabor filters

Except for recall more than 0.7 in catalogue A and B, the region growing segmentation has nearly the same precision as using the whole image and the central rectangle by grid, both the region growing segmentation method and the salient region method have the higher precision than using the whole image and the central rectangle by grid.

4.4.5 Discussion

There are some things to notice in this collection about local features from salient regions. First the number of keypoints is different in some images. Some images have a few keypoints while the maximum number of keypoints can be found in other images.

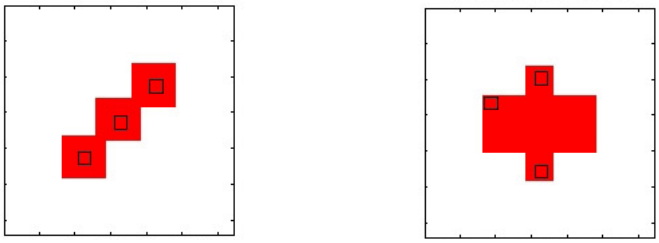


(a) (b) (c)

Figure 4.11 Images with different number of keypoints

In Figure 4.11(c) there is more than 50% of the number of keypoints located within the area that is assigned to be background. It is because the algorithm decides that the background has more salient information giving rise to the keypoints. This problem may be reduced if the background can be identified and its importance reduced.

In Figure 4.12; although the number of keypoints between image A and B are equal, they cannot discriminate between two objects.



(A) (B)

Figure 4.12 Equal numbers of keypoints, but dissimilar images

This case happens because the salient region around keypoints does not represent the image well enough to separate these two images. To relieve this problem, there are some possible ways to get more information from the image for example an increase of the region size around keypoints and a method to detect keypoints.

4.5 Summary

Retrieval experiments on simple synthetic images have been described in this chapter. The dataset has been examined using four methods and four different features. As the results in the previous chapter showed, the use of preprocessing using segmentation for example by region growing can be used to get better precision compared with the use of the whole image or only the central piece of the image. The retrieval by salient regions also does well except for the colour features in catalogue A, B and D which we believe may happen for the reasons presented in the conclusion section.

The results on this dataset suggest that query by some specific regions or a part of an object using some features do not always deliver the best result because sometimes the feature vectors that are extracted from these regions or the piece of the object may not be a reliable representation of the full object.

4.6 Conclusions

The research aims to deal with the retrieval of images containing the same object of interest as that of the query. The "global" methods capture information not only of the object of interest but also the context. The contextual information may be useful in recognizing some object classes; boat in the water; car in the street; etc. However, the risk is that the context may dominate and the system may fail to distinguish the object from the context. Hence a poor result for object retrieval is obtained. So far, we have attempted to investigate two approaches; segmentation based techniques and local features from salient regions.

Two image segmentation techniques were presented in Chapter 3. Segmentation into essential regions can be successful if objects can be separated from the background and have distinctive physical structure. For general images, more reliable segmentation is required because the segmentation process affects directly the retrieval results. Failures of the segmentation may decrease the retrieval performance.

Since perfect segmentation is difficult, the second approach using salient regions is proposed as a way to avoid segmentation. In this approach, the idea is that an image

can be represented by the local features from the areas of the image that are most interesting.

The retrieval using object segmentation and local features from salient regions in the image have been investigated and compared. The experiments show that both approaches achieve better results than the approaches using features from the whole image. However, so far the salient regions have been selected from anywhere in the image and some will be from the foreground and some from the background. If it is possible to identify salient regions which are likely to be background salient regions, it may be possible to filter these out and improve the retrieval quality. This approach will be investigated in the next chapter.

We are also interested to study more natural datasets of object images. However, for more challenging image collection which consists of more complex and cluttered images, automatic image segmentation is more difficult and normally becomes computationally more expensive. Therefore, we continue the investigation on CBIR using only the salient region approach.

Chapter 5 Background Filtering

Salient regions are regions in an image where there is a significant variation with respect to one or several image features. In content-based image retrieval (CBIR), salient points and regions are used to represent images or parts of images using local feature descriptions. In (Sebe, Tian et al. 2002; Hare and Lewis 2004) the salient region approach has been shown to outperform the global approach. Many researchers have proposed different techniques based on salient points and regions. For example, Schmid and Mohr (Schmid and Mohr 1997) proposed using salient points derived from corner information as salient regions for image retrieval, whilst Q.Tian et al. (Tian, Sebe et al. 2001) used a salient point detector based on the wavelet transform.

Salient regions are also applied to the problem of object retrieval, for example, in the case where a specific object in a query image is required to be retrieved from the image database. Traditional CBIR based on salient regions begins with salient region detection. Each salient region is then typically represented by a feature vector extracted from the region. In the query step, there is matching between salient regions from the query image and those from images in the collection and similar images are ranked according to the quality of match.

However, one of the reasons that the accuracy of object retrieval may be less than optimal is the presence of salient regions in the retrieval process which are not located on the object of interest.

Attempts to reduce the influence of irrelevant regions have appeared in some research projects. Ling Shao and Michael Brady (Ling Shao 2006) classified the selected regions

into four types before the use of correlations with the neighbouring region to retrieve specific objects. Hui Zhang at el. (Zhang, Rahmani et al. 2006) pruned salient points using segmentation as a filter.

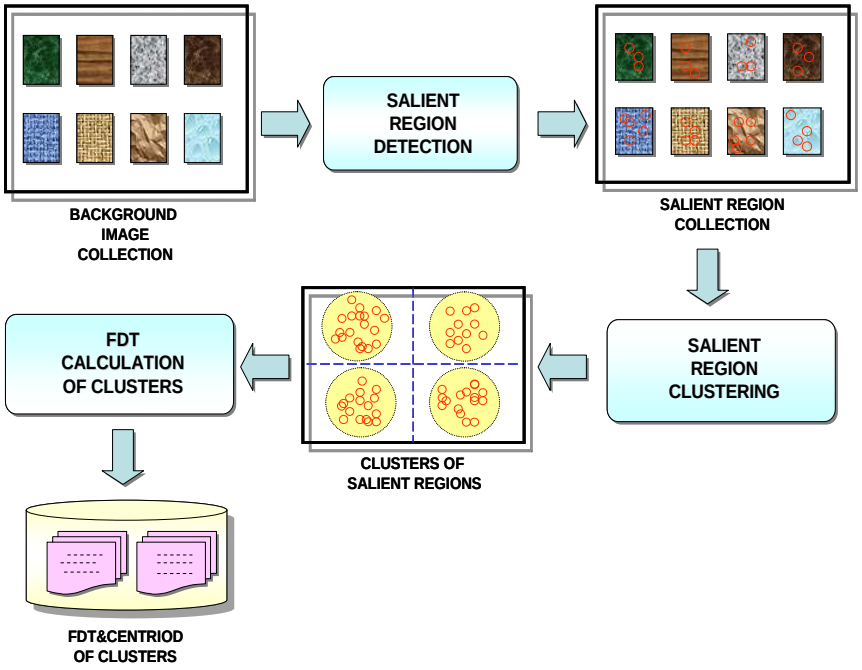


Figure 5.1 Background clusters for filtering salient regions

In this chapter we propose a method to filter salient regions using background information. Situations where the technique may be particularly appropriate are those where the image backgrounds are not completely arbitrary but can be characterized by a limited number of prototypes. Identifying particular objects in indoor office scenes is an example.

In our approach, the system begins by creating clusters of salient regions from a collection of background only images. Thereafter, when processing images containing objects, salient regions with a high probability of belonging to a background cluster are removed before further processing. The process will be described in Section 5.1. It is illustrated schematically in Figure 5.1 and uses a distance threshold from the centre of each cluster called the fractional distance threshold (FDT).

For image retrieval, the background filtering step is applied after salient regions have been extracted. The process is illustrated schematically in Figure 5.2.

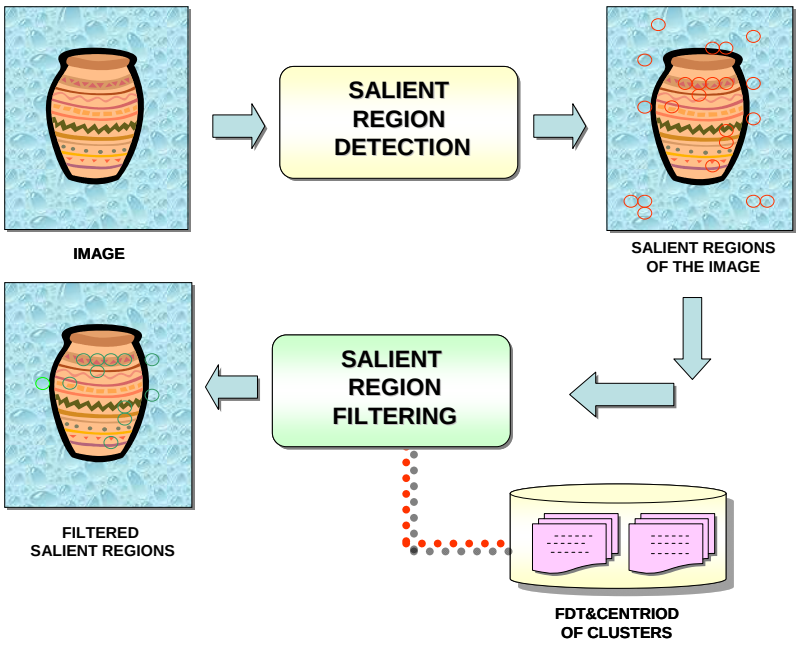


Figure 5.2 The salient region filtering process

In the following sections we show that by using salient region filtering it is possible to reduce the number of unwanted salient regions and improve the precision of the retrieval process.

The chapter is organised as follows. In Section 5.1, the methods of background clustering and calculation of the fractional distance threshold are introduced. The experimental procedure is described in Section 5.2. Results and discussion are presented in Section 5.3 and finally, Section 5.4 presents the conclusions and a brief discussion of future work.

5.1 Background Clustering

5.1.1 Salient Region Detection and Feature Extraction

Recently, many local detectors which can identify salient regions in an image have been described and evaluated (Fraundorfer and Bischof 2005; Moreels and Perona 2005). In Chapter 4 we introduced one of the popular approaches to salient region detection and representation which uses the multi-scale Difference-of-Gaussian (DoG) pyramid for region detection and the SIFT (Scale Invariant Feature Transform) from Lowe (Lowe 2004) to represent the detected salient regions. For each salient region, a 3D histogram of gradient locations and orientations is calculated. The SIFT descriptor has been evaluated by Mikolajczyk and Schmid in (Mikolajczyk and Schmid 2005) to be one of the best performing local descriptors. The DoG and the SIFT approaches to salient region detection and representation are those adopted in our work here.

5.1.2 Background Cluster Construction

One assumption of the method presented here is that salient regions from the foreground objects are reasonably distinct from background salient regions, or that any similarities involve a sufficiently small proportion of the total object salient regions to make their removal negligible.

The method begins with the detection of salient regions in a collection of background images. Since large numbers of salient regions may typically be detected in a single image, a random sample of salient regions are selected from all the background images and feature descriptors are extracted and used to cluster the salient regions into k clusters using the k-means clustering algorithm.

Since the clusters are derived from salient regions on background only images, these clusters are identified as the *background clusters*. The centroid of each cluster is calculated, essentially as a 128 element SIFT descriptor. Members in the same cluster are background salient regions that are similar to each other and dissimilar to the salient regions of other groups.

Many of the clusters are quite small in number so deriving a valid statistical model of the background clusters is not possible but to discriminate between salient regions on foreground and background, we determine an appropriate percentile distance from each cluster centroid, which we call the fractional distance threshold (FDT) for each of the background clusters. The FDT of a cluster is the distance between a cluster member at a particular percentile and the centroid of that cluster. Thus FDT (90) is the distance from the centroid to a cluster member for which 90% of cluster members are nearer the centroid. The same percentile is used for all clusters and the appropriate percentile value found by experiment (see Section 5.2). The actual FDT and centroid for each cluster is retained for use in the retrieval process.

In the salient region filtering step, salient regions are detected and the features extracted. Any salient region (*S*) which has a feature distance (*D*) to the centroid (*C*) greater than the FDT (*i*) value for all (*n*) clusters is assumed to be a salient region on the foreground (S_{fg}). Otherwise, it is assumed to belong to a background region (S_{bg}) as is represented by the following formula.

$$S = \begin{cases} S_{fg} & , \text{if } \forall k \left[D(S, C_k) > FDT(i)_k \right] \\ S_{bg} & , \text{otherwise} \end{cases} \qquad \text{where } k \in \{1, 2, ..., n\} \text{ and } i \in \{1, 2, ..., 100\} \qquad [5\text{-}1]$$

5.2 Experiments

We separated the experimentation into 2 parts. The first part is to establish appropriate parameters for the clustering and fractional distance threshold estimation and the second is to evaluate the retrieval performance using background salient region filtering.

5.2.1 FDT Percentile Estimation

A background only image collection, composed of 120 background only images (400 x 300 pixels) was created for 12 different backgrounds (10 images per background). Salient regions were extracted from each of the images and the number of salient

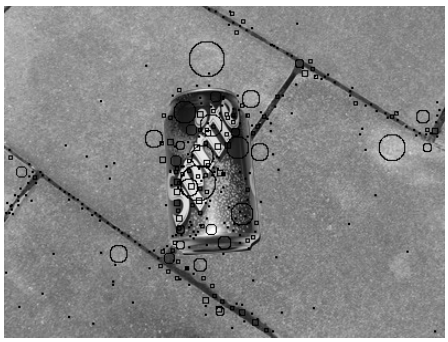
regions found in each image varied between 8 and 3,503 depending on image content. Figure 5.3 shows some examples of background images from the dataset.

Figure 5.3 Samples of background images

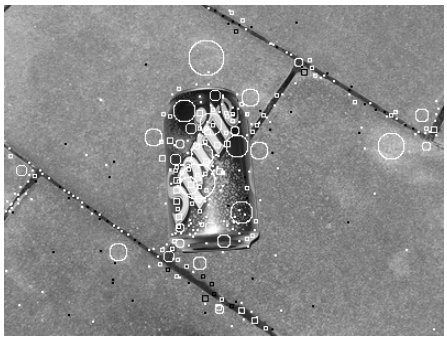
In order to find appropriate values for the number of clusters in the k-means clustering, k, the number of randomly selected salient regions to use, S, and the percentile setting for the FDT calculation, a range of k and S combinations was used for clustering and the FDT estimated at each percentile from 50 to 100 in steps of 5. Each of eleven different k and S combinations were used. The resulting FDTs were used to check the percentage of correctly assigned foreground and background salient regions on a collection of object and background images. For these images, the ground truth was established by manually delineating the area covered by the object and if the centre of a salient region (SR) fell in the object area it was taken as an object SR. Otherwise, it was taken as a background SR.

Figure 5.4 shows examples of the decisions made by the system using some of the different FDT values. Salient regions in white circles represent foreground SRs and those with black circles represent background SRs.

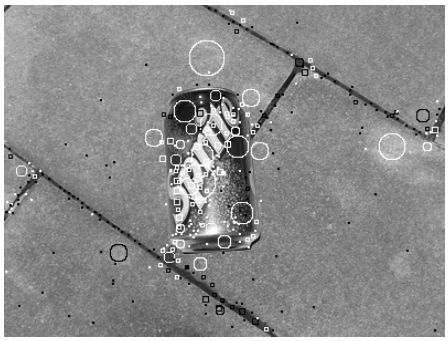
(A) An original image



(B) The image with salient regions



(C) The image with filter at FDT = 50



(D) The image with filter at FDT = 80

Figure 5.4 Foreground (white circle) and Background (black circle) salient regions at FDT = 50 and FDT = 80

The performance of the decisions for a range of k, S and FDT values is measured via the receiver operating characteristic (ROC) space (Fawcett 2004). A ROC space represents the relationship of true positive rates (TP) and false positive rates (FP). Each classification produces a (TP and FP) pair corresponding to a single point in ROC space. We define

w as the number of **correct** predictions that an instance is **Foreground SR**
x as the number of **incorrect** predictions that an instance is **Background SR**
y as the number of **incorrect** predictions that an instance is **Foreground SR**
z as the number of **correct** predictions that an instance is **Background SR**

The recall or true positive rate (TP) determines the proportion of background SRs that were correctly identified, as calculated using the equation:

$$TP = \frac{z}{y+z} \quad [5\text{-}2]$$

The false positive rate (FP) defines the proportion of foreground SRs that were incorrectly classified as background SRs, as calculated using the equation:

$$FP = \frac{x}{w+x} \quad [5\text{-}3]$$

Figure 5.5 shows the ROC curve (TP against FP) as the FDT percentile is varied from 50 to 100.

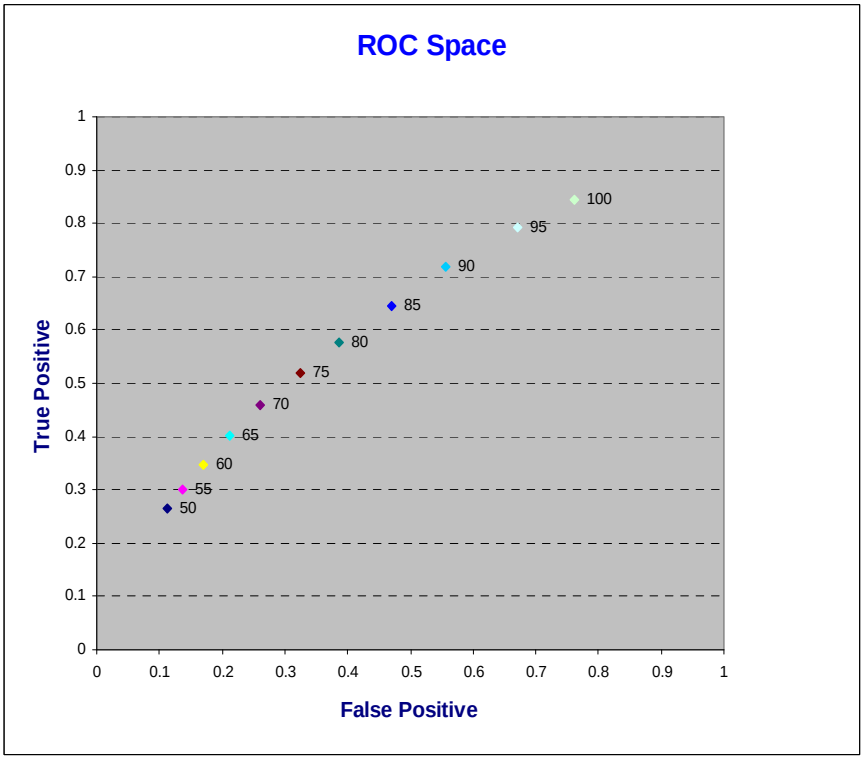


Figure 5.5 The TP and FP coordinates of FDT at 50 to 100 on the ROC space

To compare the prediction performance, distances are calculated from all points to the perfect classifier point in ROC space which is the point (0, 1). The point (0, 1) means all regions are classified correctly.

The overall results are presented in Table 5.1 where, for each of the *k* and *S* combinations, the table shows the distance from all of the TP and FP pairs to the point (0, 1). It illustrates how the percentage correct varies with the percentile for the FDT. It can be seen that in general a percentile of 85 gives the best results and that this is achieved with a *k* value of 5,000 and an *S* value of 50,000. These values are used in the retrieval experiments in the following section.

Background Cluster (k - cluster, S - sample)	FDT										
	50	55	60	65	70	75	80	85	90	95	100
A (k500,S5000)	0.7528	0.7332	0.6997	0.6661	0.6336	0.6134	0.6038	0.6121	0.6408	0.7325	0.7893
B (k500,S10000)	0.6747	0.6460	0.6104	0.5874	0.5647	0.5675	0.5987	0.6381	0.6976	0.7547	0.8831
C (k500,S50000)	0.6048	0.5725	0.5474	0.5469	0.5600	0.5880	0.6332	0.7012	0.7755	0.8759	0.9778
D (k500,S100000)	0.5925	0.5614	0.5388	0.5335	0.5488	0.5809	0.6403	0.7145	0.7951	0.8967	0.9859
E (k1000,S5000)	0.8536	0.8431	0.8155	0.7786	0.7204	0.6799	0.6574	0.6285	0.5791	0.6279	0.6334
F (k1000,S10000)	0.7871	0.7635	0.7200	0.6646	0.6253	0.5931	0.5669	0.5840	0.5968	0.6954	0.7600
G (k1000,S50000)	0.6637	0.6198	0.5797	0.5466	0.5305	0.5456	0.5691	0.6295	0.7074	0.8094	0.9363
H (k1000,S100000)	0.6494	0.6040	0.5645	0.5315	0.5200	0.5374	0.5799	0.6621	0.7448	0.8612	0.9658
I (k5000,S10000)	0.9731	0.9730	0.9687	0.9556	0.9473	0.9437	0.9372	0.8842	0.7693	0.7304	0.7291
J (k5000,S50000)	0.8637	0.8439	0.7962	0.7437	0.6926	0.6214	0.5667	0.5149	0.5304	0.6075	0.6864
K (k5000,S100000)	0.8206	0.7886	0.7470	0.6902	0.6254	0.5733	0.5179	0.5194	0.5610	0.6440	0.7763

Table 5.1 The distance to (0,1) of 11 background cluster types (A-K) at the different FDT value (50 – 100). The lower the distance, the better the classifier.

5.2.2 Object Retrieval

In order to test the effectiveness of background filtering, two test datasets, each of 120 individual object images, were created from 10 objects using 12 different background patterns. These patterns are the same as the background patterns of the training set but different images of the patterns are used (i.e. different scales and camera orientations).

In dataset 1, the number of salient regions in these images is between 98 and 1,728 regions. There are no scale and orientation change in each object. In dataset 2, the number of salient regions per image is between 174 and 2,349 regions. The scale and orientation is varied. From the results of the clustering experiments described earlier, the 5,000 background clusters from 50,000 salient points were used as the background

clusters in the retrieval experiment and the chosen FDT value for all clusters was set to 85. Each object image was used in turn as the query image. The salient regions were extracted and background salient regions were filtered out from both the query image and the remaining object dataset images. After pruning, the strongest 50 salient regions from the remaining SRs were used to calculate the similarity between the query and dataset images and precision and recall results were obtained. The experiment was repeated without background filtering.

5.3 Results and Discussion

The precision and recall graphs with and without background filtering are shown in Figure 5.6 for dataset 1. From the graph it can be seen that the object retrieval system with background filtering outperforms the system without background filtering with an improvement in precision. The average precision with background filtering is 0.25 and without background filtering average precision is 0.18.

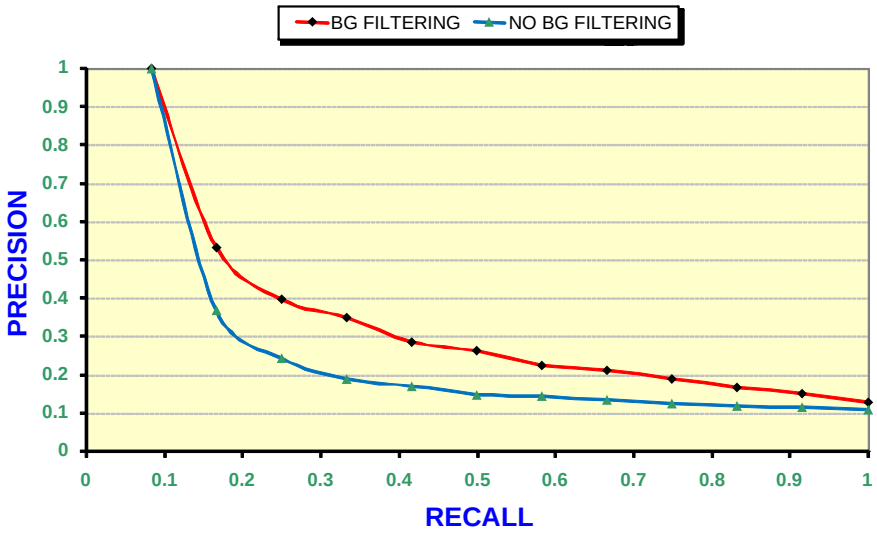


Figure 5.6 Dataset 1. Precision and recall with and without background filtering

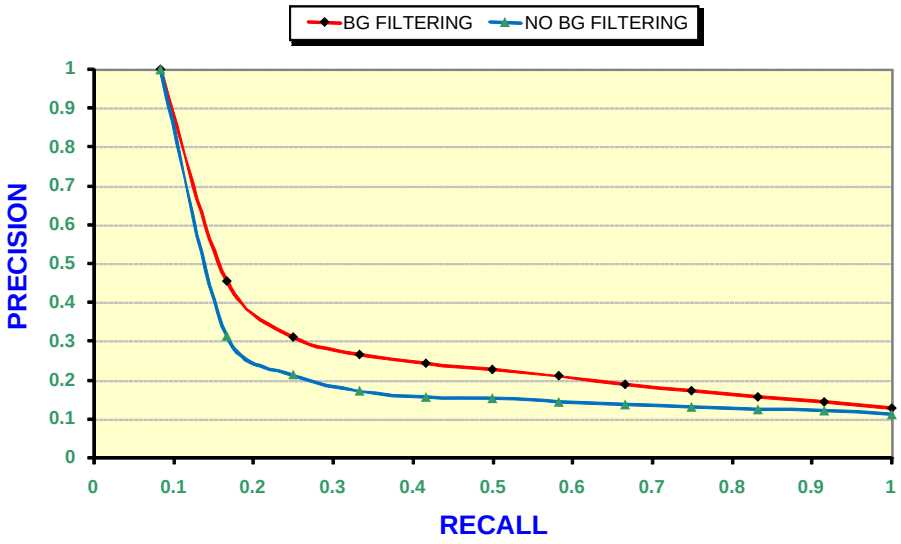


Figure 5.7 Dataset 2. Precision and recall with and without background filtering

For the more challenging dataset 2, the precision and recall graph is shown in Figure 5.7. Again the performance is improved by using background filtering. The average precision is 0.24 without background subtraction and is 0.27 with background subtraction.

Looking back to Figure 5.5, it can be seen that best salient region classification performance is still far from the perfect classifier. The salient region filtering uses the particular differences between object areas and background areas to discriminate these regions and this is clearly not very robust.

5.4 Conclusion

A method of filtering background salient regions for object retrieval is developed and implemented. A comparison has been made between retrieval with and without background salient region filtering and the filtering process is found to give improvements in precision. This is a rather preliminary evaluation of the technique but is sufficient to suggest that a more robust way of modelling the backgrounds in terms of salient regions might be sought.

In summary, the background filtering method is an attempt to distinguish between the objects and the surrounding areas. Since certain types of query can benefit from using background information to filter irrelevant regions, further attempts are being made to improve performance of this technique.

In the next chapter a more robust way of estimating the probabilities of salient regions being background or foreground regions is investigated and is shown to lead to an improvement in retrieval performance.

Chapter 6 K-Nearest Neighbour Classification for Background and Foreground

In the last chapter, pruning background salient regions based on a background collection has been studied. It was shown that the image retrieval precision could be increased when some background salient regions were filtered. In this chapter we continue to explore background salient region pruning by using an approach based on K-Nearest Neighbour (K-NN) classification.

In this case we assume that both foreground and background information is provided in the training step. The classification into background and foreground classes is possible to achieve. To identify which salient region is then the background class or the foreground class, the statistical information of the salient region is calculated from its neighbours.

After classification, all salient regions assigned to the background class are filtered out and then salient regions which have the highest probability of being salient regions located on foreground are selected as the image representation.

The result from the experiment shows that the performance of image retrieval with pruning of salient regions is improved when compared with image retrieval without pruning.

6.1 K-Nearest Neighbour (K-NN) and the Probability

The K-Nearest Neighbour technique is a method for classifying unknown objects based on the K closest training examples in the feature vector space. In this space, the nearer the vectors in terms of Euclidean distance, the more similar the "objects" represented. Here we applied the K-NN method to predict salient regions which are described by feature vectors.

The training set D is composed of salient region examples which are labelled by their class, foreground or background. The salient region examples in the training set are in the background class when the centre of a salient region is located in the background area and the salient region is in the foreground class if its centre is located within the foreground object region. For the test set, the class of a salient region x is predicted by computing the similarity between x and its k nearest neighbours in D and then either assigning x to the class most commonly occurring amongst its k nearest neighbours or estimating the probability of being in the background or foreground as follows.

Let P_{bg} be the probability that a salient region is in the background class. Consider the k nearest neighbours from the "training set". If b of these k nearest neighbours are background, an estimation of P_{bg} is

$$P_{bg} = \frac{b}{k} \qquad [6.1]$$

Also if P_{fg} is the probability that a salient region is foreground then $P_{fg} = 1 - P_{bg}$

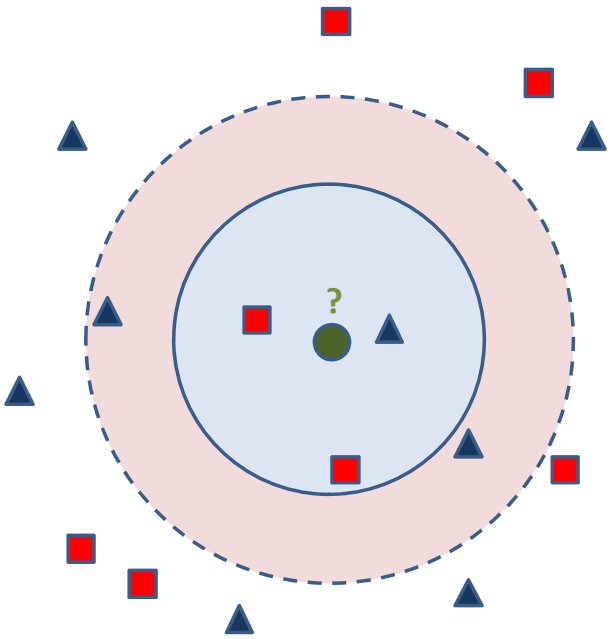


Figure 6.1 The test patch (green circle) should be classified either to the background class of blue triangles or to the foreground class of red squares. If k = 3, the P_{bg} of the green circle equals 0.33 and the P_{fg} of the green circle equals 0.67. If k = 5 it has the P_{bg} of the green circle equals 0.6 (3 triangles inside the outer circle) and the P_{fg} = 0.4.

In the experiments, the classification of a testing region is based on estimates of its probability of being background and foreground. The reliability of the nearest neighbour method depends crucially on the separability of the classes in the feature space and an appropriate choice of *k* which are illustrated in Figure 6.1.

6.2 The Experiments

The experiments are divided into two parts: finding the optimum background probability cut and image retrieval. The first part focuses on the background probability. This step aims to identify which salient regions are in the background. The background probability is used to try to perform reliable background filtering.

The training set was composed of examples of foreground and background salient regions. All salient regions in images from the training set were detected by the DoG method and represented by SIFT descriptors (see Chapter 4). In this experiment 57,048 background salient regions and 9,585 foreground salient regions were extracted from 120 images of 10 objects. Each object image was taken on 12 different backgrounds.

Then all salient regions in each image were categorized to be either background or foreground examples depending on the location of the centroid in the mask image. All images have their own mask images which presents the area of foreground and background.

To label a salient region in the test set, the Euclidian distance from the salient region to all the salient regions in the training set is calculated to find the k nearest salient regions. Each salient region from the test set has a background probability (P_{bg}) which is calculated as described above.

In order to classify a salient region (S) to be a salient region in the background class (S_{bg}) or the foreground class (S_{fg}), we define a threshold called "Being Background" (BB). Any salient region which has P_{bg} greater than or equal to BB is allocated to the background class.

$$S = \begin{cases} S_{bg} & \text{, if } P_{bg} \geq BB \\ S_{fg} & \text{, otherwise} \end{cases} \qquad [6.2]$$

6.2.1 Finding the Optimum Background Probability Cut

In this experiment, the objective is to find the best parameter estimation to classify salient regions into the background and foreground classes. From equation [6.1] and [6.2], BB and k are parameters that effect directly the classification. BB can vary from 0.0 to 1.0.

In order to find the optimum value for k, a range of k values from 1 to 10 was used. Each salient region was picked to calculate P_{bg} when k was varied from 1 to 10. The average accuracy (AVG.AC); the average true positive (AVG.TP) and the average false positive (AVG.FP) of being background (see details in Section 2.4.3) were measured to compare the salient region classification performance. It is noted that the salient region under test is excluded from the k nearest neighbours as it would always match itself first. The results of the tests are shown in
Table 6.1. Each salient region in the training set was used in each test.

k	BB	AVG.AC	DISTANCE	AVG.TP	AVG.FP
1	1.0000	0.9576	0.2018	0.9757	0.2003
	0.0000	0.7877	1.0000	1.0000	1.0000
2	1.0000	0.9448	0.1702	0.9520	0.1633
	0.5000	0.9530	0.2966	0.9901	0.2964
	0.0000	0.7877	1.0000	1.0000	1.0000
3	1.0000	0.9296	0.1557	0.9284	0.1383
	0.6667	0.9504	0.2550	0.9784	0.2541
	0.3333	0.9433	0.3677	0.9933	0.3676
	0.0000	0.7877	1.0000	1.0000	1.0000
4	1.0000	0.9119	0.1560	0.9022	0.1215
	0.7500	0.9462	0.2269	0.9669	0.2245
	0.5000	0.9450	0.3197	0.9855	0.3193
	0.2500	0.9072	0.5290	0.9945	0.5289
	0.0000	0.7877	1.0000	1.0000	1.0000
5	1.0000	0.8933	0.1648	0.8764	0.1091
	0.8000	0.9381	0.2067	0.9524	0.2012
	0.6000	0.9429	0.2887	0.9765	0.2878
	0.4000	0.9151	0.4651	0.9879	0.4649
	0.2000	0.9004	0.5750	0.9976	0.5750
	0.0000	0.7877	1.0000	1.0000	1.0000
6	1.0000	0.8778	0.1749	0.8545	0.0970
	0.8333	0.9286	0.1917	0.9375	0.1812
	0.6667	0.9286	0.1917	0.9375	0.1812
	0.5000	0.9183	0.4170	0.9811	0.4166
	0.3333	0.9116	0.5142	0.9945	0.5141
	0.1667	0.8947	0.6077	0.9980	0.6077
	0.0000	0.7877	1.0000	1.0000	1.0000
7	1.0000	0.8605	0.1907	0.8309	0.0882
	0.8571	0.9193	0.1835	0.9228	0.1665
	0.7143	0.9328	0.2436	0.9549	0.2394
	0.5714	0.9185	0.3812	0.9720	0.3802
	0.4286	0.9171	0.4717	0.9908	0.4716
	0.2857	0.9171	0.4717	0.9908	0.4716
	0.1429	0.8900	0.6352	0.9983	0.6352
	0.0000	0.7877	1.0000	1.0000	1.0000
8	1.0000	0.8438	0.2090	0.8085	0.0837
	0.8750	0.9096	0.1798	0.9079	0.1544
	0.7500	0.9272	0.2266	0.9441	0.2196

k	BB	AVG.AC	DISTANCE	AVG.TP	AVG.FP
	0.6250	0.9155	0.3557	0.9632	0.3538
	0.5000	0.9185	0.4411	0.9847	0.4409
	0.3750	0.9124	0.5074	0.9925	0.5073
	0.2500	0.9013	0.5755	0.9959	0.5755
	0.1250	0.8662	0.7281	0.9985	0.7281
	0.0000	0.7877	1.0000	1.0000	1.0000
9	1.0000	0.8284	0.2255	0.7878	0.0763
	0.8889	0.8996	0.1784	0.8926	0.1425
	0.7778	0.9201	0.2161	0.9323	0.2052
	0.6667	0.9125	0.3342	0.9544	0.3311
	0.5556	0.9185	0.4149	0.9794	0.4143
	0.4444	0.9151	0.4739	0.9877	0.4738
	0.3333	0.9068	0.5408	0.9933	0.5407
	0.2222	0.8782	0.6684	0.9962	0.6684
	0.1111	0.8627	0.7496	0.9992	0.7496
	0.0000	0.7877	1.0000	1.0000	1.0000
10	1.0000	0.8139	0.2420	0.7688	0.0715
	0.9000	0.8892	0.1804	0.8777	0.1326
	0.8000	0.9128	0.2075	0.9201	0.1915
	0.7000	0.9087	0.3131	0.9449	0.3082
	0.6000	0.9187	0.3885	0.9739	0.3876
	0.5000	0.9160	0.4474	0.9832	0.4471
	0.4000	0.9107	0.5071	0.9891	0.5070
	0.3000	0.8851	0.6306	0.9940	0.6306
	0.2000	0.8743	0.6950	0.9980	0.6950
	0.1000	0.8601	0.7627	0.9993	0.7627
	0.0000	0.7877	1.0000	1.0000	1.0000

Table 6.1 Average AC, TP and FP values with varied *BB* at k = 1-10

From the table it can be seen that the best accuracy is when k = 1 and *BB* value is 1.0. The data can also be explored using a ROC graph showing true positives against false positives. This is shown in Figure 6.2.

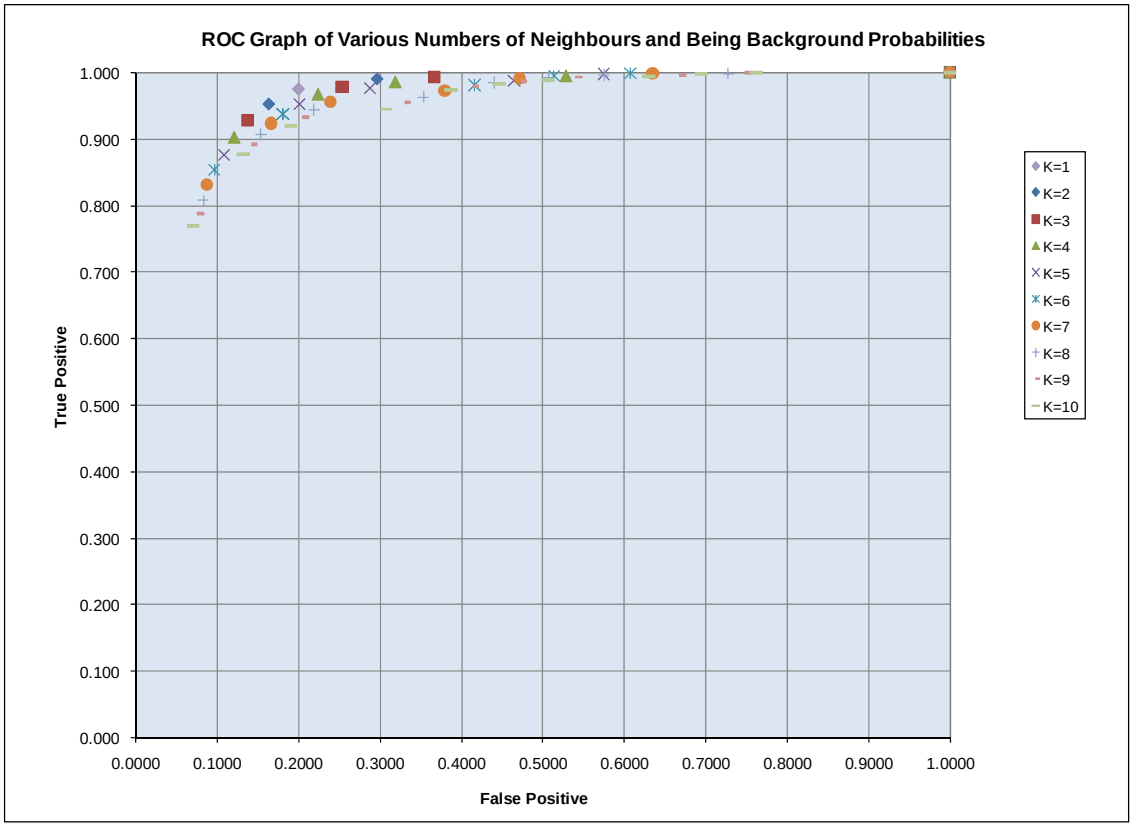


Figure 6.2 ROC graph of various numbers of neighbours k = 1-10

To establish which k value and threshold BB produce the best background classification, the distance from all points on the ROC curve to the best classifier point (0,1) is calculated.

Column 4 in Table 6.1 shows the distance of the points in ROC space to the perfect point (0,1). Using this approach the best background classification is when $BB = 1.0$ and $k = 3$.

6.2.2 Image Retrieval

The image database is composed of 120 images (see Appendix B). There are 10 objects. Each object image was taken on 12 different backgrounds with different positions and orientations but in the same plane.

To prepare a representation of images from the database, salient regions were extracted from each image and the background probability P_{bg} was calculated for each. The salient regions for which the background probability was greater or equal than BB were filtered out. The remaining salient regions were classified as foreground salient regions. Up to fifty salient regions which had the highest foreground probability P_{fg} were selected to represent the image. In the cases that there was more than one salient region which had the same value of P_{fg} , the selection was based on the curvature value (see Section 4.1.2) of salient regions.

For the querying step, the query image was represented by the SIFT descriptors from fifty salient regions. The process of the calculation was the same as for all images in the database. The ranking was based on the Euclidian distance of the SIFT descriptors between the query image and all images in the database.

Since there are two methods for the performance measurement, a comparison is made for the system which is believed to reach the best performance on image retrieval. A system setting $k = 3$ with $BB = 1.0$ and the system setting $k = 1$ and $BB = 1.0$ were also compared with the system without filtering. The result shows on the precision and recall graph in Figure 6.3.

It could be seen from Figure 6.3 that the best retrieval on the recall from 0.0 - 0.75 is the setting $BB = 1.0$ and $k = 3$ which achieve the best classification in the ROC graph. The setting $BB = 1.0$ and $k = 1$ is slightly better than the other settings on recall 0.75 - 1.0.

From this figure, two parameter settings with salient region filtering have higher precision than the retrieval without filtering. This indicates that the filtering process has improved the retrieval performance on this dataset.

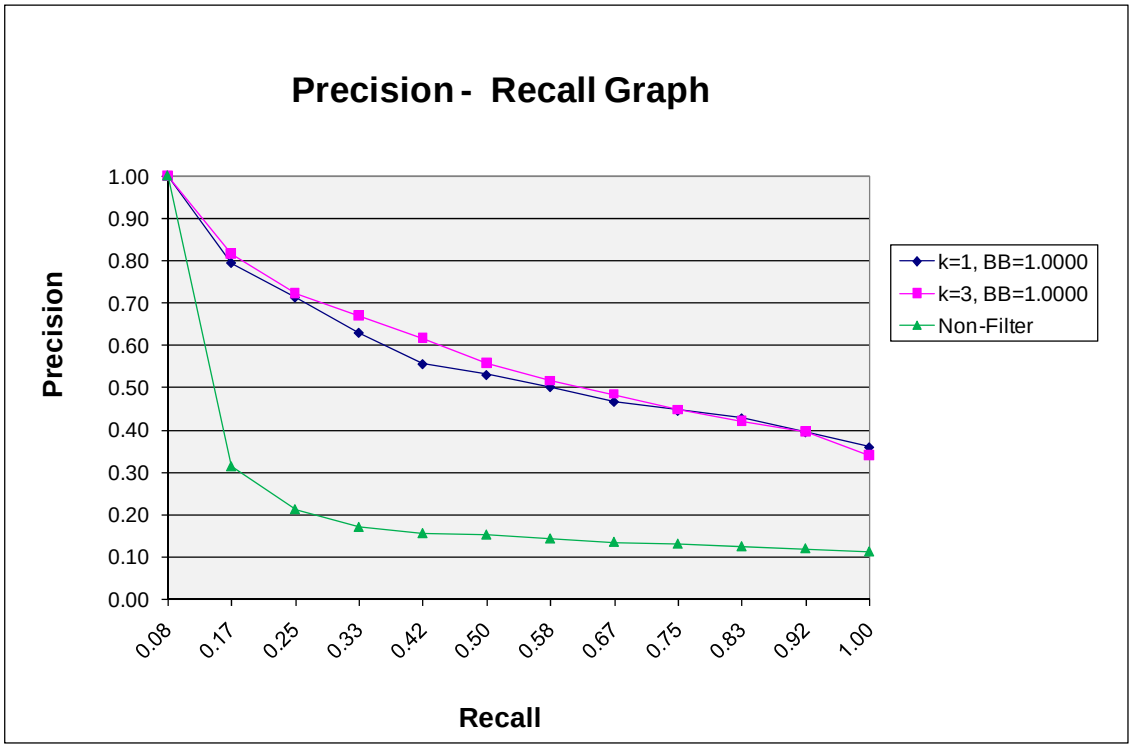


Figure 6.3 A precision and recall graph shows the performance of 3 setting.

6.3 Discussion and Summary

In this chapter, the K-Nearest Neighbour technique is used to classify salient regions into the background and foreground classes. The result of the classification is decided by a being background probability and a being foreground probability of each salient region.

To achieve the best classification, the parameter optimization was examined. A number of the nearest neighbours, k, is the first parameter that affects the result of classification. Being Background (*BB*) is set to be a threshold for the probability being background. It is used to separate between the foreground and background class. The performance of the parameter optimization was measured by computing the average accuracy (AVG.AC) of classification and finding the best settings when they were compared by measuring the distance on the ROC graph. The best parameter setting was used in the image retrieval application.

For image retrieval, all images in a database are represented by the top group of salient regions which have a high probability of being foreground. To increase the accuracy of representation of the object in the image, all salient regions which have the probability of being background greater than or equal to the threshold the BB are filtered out.

The experiment shows that K-NN classification has the potential to improve the performance of image retrieval. In the next chapter, the Support Vector Machine (SVM) classification is explored as a possibly superior classification technique.

Chapter 7 Foreground-Background Classification using SVM

In the previous chapter, K-nearest neighbour classification based on the probability of being foreground and background was applied to filter out salient regions located on the background and select only the salient regions with high probability of being foreground as the salient regions to represent the images. In this chapter, one of the most popular techniques of classification, Support Vector Machines (SVM) is investigated and applied to the problem of learning the class of salient regions, foreground or background.

Unlike the previous chapter which uses scale-invariant feature transform (SIFT) as the region descriptor, here PCA-SIFT descriptors are used to represent an image. The PCA-SIFT is computationally more efficient and reduces the influence of noise in the data.

7.1 Support Vector Machine (SVM)

The Support Vector Machine is currently one of the most popular techniques in machine learning. It has been applied to a broad class of computer vision problems, including object recognition e.g. (Pontil and Verri 1998), image retrieval e.g. (Zhang, Lin et al. 2001), (Kim, Song et al. 2007) and image classification e.g. (Ren, Shen et al. 2004) .

SVM classification constructs a hyperplane in the vector space to manage the input data as two groups of vectors in an n-dimensional space. This hyperplane optimally separates the data into two categories. A good separation is achieved by the hyperplane that has the largest distance to the neighbouring data points of both classes.

The aim of SVM modelling is to find the optimal hyperplane that separates clusters of vectors in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the *support vectors*. Figure 7.1 shows the maximum margin decision hyperplane in SVM which can separate data into 2 classes (green circles and red triangles).

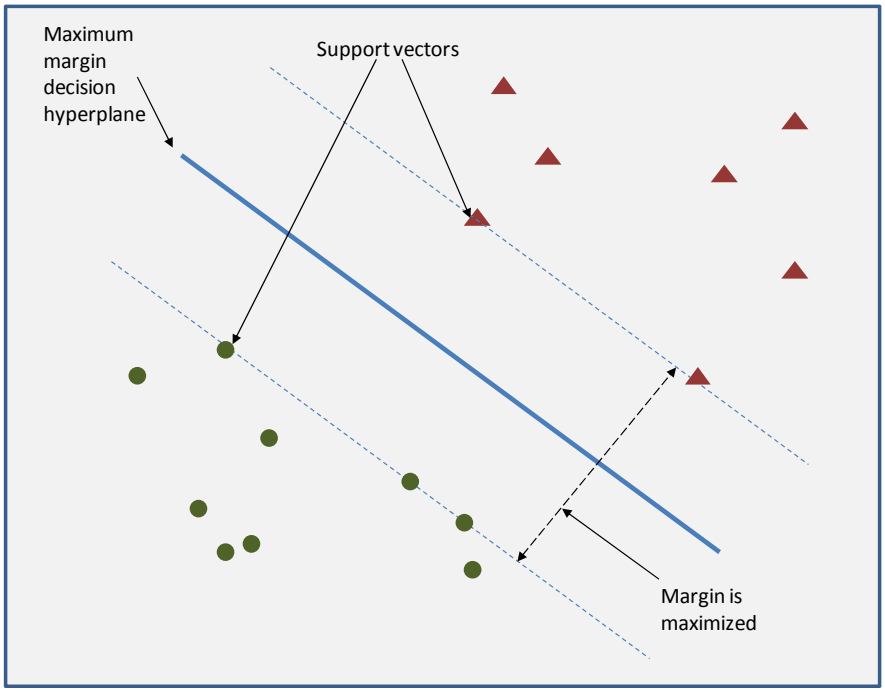


Figure 7.1 Maximum margin decision hyperplane (green and red triangles are training data)

The SVM learns how to classify from a training set of feature vectors, whose expected outputs are already known. The training step enables a binary classifying SVM to define a hyperplane in the feature space, which optimally separates the training vectors of two classes. When there is a new feature vector as an input, its class is predicted on the basis of which side of the plane it maps to.

Sometimes, the training step of SVM takes a much longer time when dealing with the larger n-dimensional vector spaces. In this chapter, one of the solutions to alleviate this problem is dimensional feature reduction. In the next part, the PCA-SIFT descriptor is introduced.

7.2 The PCA-SIFT descriptor

The popular Scale Invariant Feature Transform (SIFT) descriptors introduced by Lowe (Lowe 2004), were used in the previous experiments described above. As described in Chapter 4, the keypoint descriptor represented by the SIFT is built by a local descriptor which is based on the image gradients from a patch of pixels in its own neighbourhood. In each patch, the orientation 4 x 4 histograms with 8 orientation bins each are created to capture the important aspects

In 2004, Yan Ke and Rahul Sukthankar (Ke and Sukthankar 2004) applied Principal Component Analysis (PCA) (Joliffe 1986) which is a technique used here to reduce the dimensions of the SIFT. Although the dimension is reduced from a 128-element vector descriptor, the new descriptor under the name "PCA-SIFT", is still well-suited to representing keypoint patches. In (Ke and Sukthankar 2004), there are experiments comparing the performance between SIFT and PCA-SIFT. The result shows that the PCA-SIFT is more distinctive, more robust to image deformations, and more compact than the standard SIFT representation.

PCA-SIFT descriptors encode the salient aspects of the image gradient in the feature point's neighbourhood; however, instead of using the SIFT's smoothed weighted histograms, Principal Component Analysis (PCA) is applied to the normalized gradient patch.

Unlike previous experiments, the 36 dimensional PCA-SIFT descriptors are used to represent salient regions. Because the PCA-SIFT descriptor is composed with a smaller number of element vectors than the SIFT descriptor, using the PCA-SIFT can get the benefit of space saving.

Not only does PCA-SIFT save the space cost, but it saves the time processing as well. Table 7.1 shows the training time for a training set (in Section7.3.1) on a 2 dual core 3GHz Xeons. The 128 dimensional SIFT descriptor consumes significantly more training time than the 36 dimensional PCA-SIFT.

Region Descriptor	Training Time (seconds)
SIFT	22,456
PCA-SIFT	11,085

Table 7.1 Training time comparison between SIFT and PCA-SIFT

Figure 7.2 shows results of image retrieval by the K-NN classification in Chapter 6 using k=1 and BB=1.0. There is a comparison between the image retrieval using SIFT and PCA-SIFT as the descriptors. In this dataset, the PCA-SIFT performed better than the SIFT. The parameter setting k=3 and BB=1.0 in Figure 7.3 shows the same trend as Figure 7.2.

For these reasons, PCA-SIFT is preferred to represent salient regions in order to save processing times and space in the next experiments.

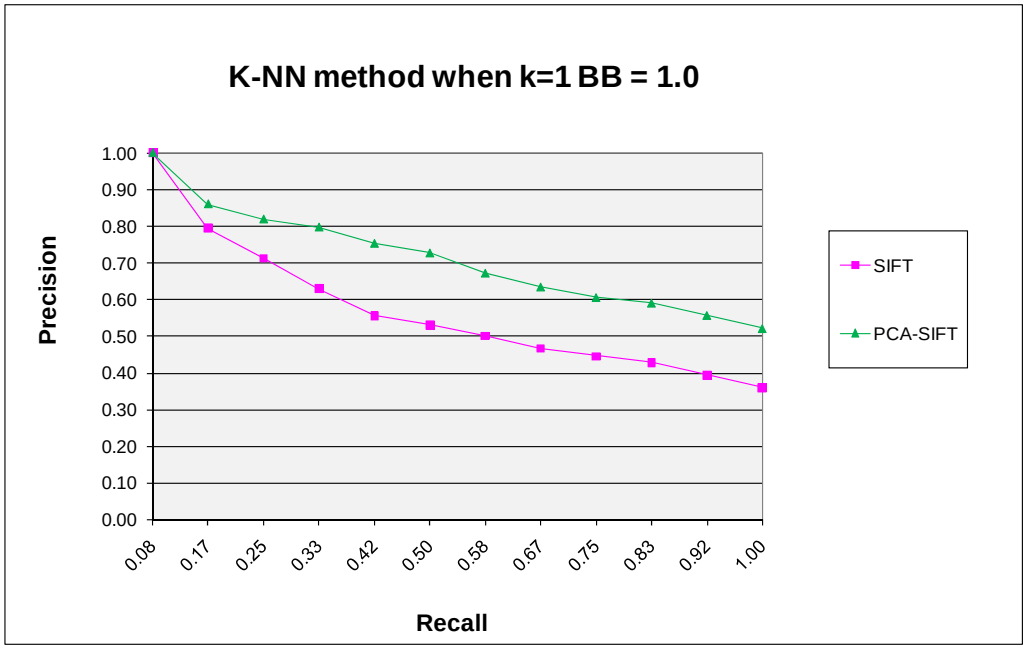


Figure 7.2 Image retrieval performance of K-NN method when k = 1 and BB = 1.0 comparing between SIFT descriptor and PCA-SIFT descriptor.

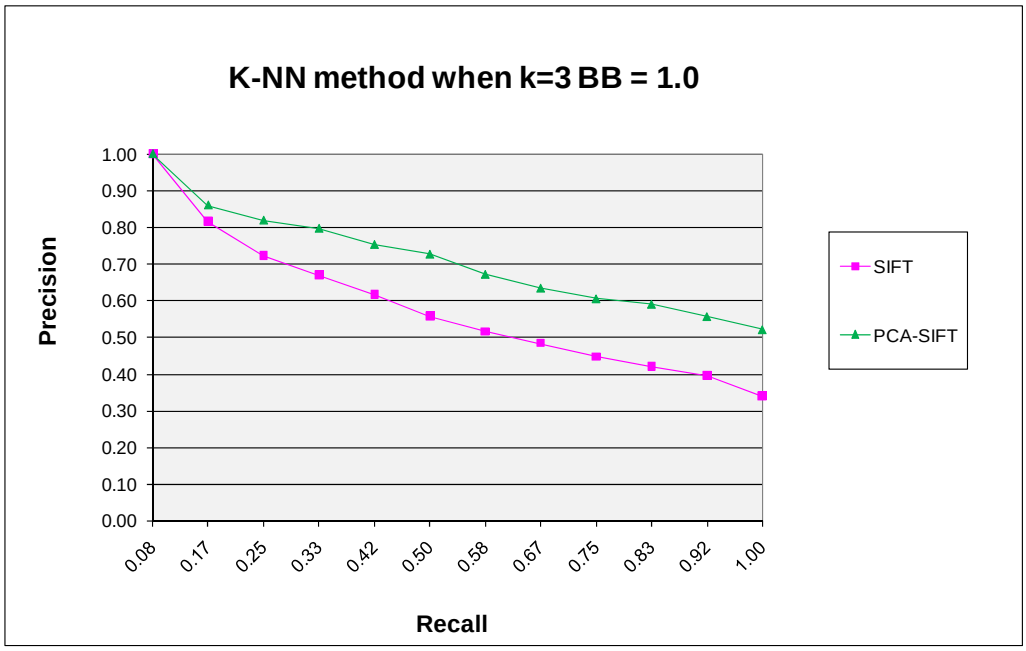


Figure 7.3 Image retrieval performance of K-NN method when k = 3 and BB = 1.0 comparing between SIFT descriptor and PCA-SIFT descriptor

7.3 Experiments

The goal of the experiment is to use SVM to classify salient regions into the background or foreground class. Experiments are separated into 3 steps. The first step is the training step which is followed by the prediction step. The last step is using the results of prediction to evaluate the image retrieval. The training and testing set is the same as the experiments in Chapter 6.

The libraries from LIBSVM (Chang and Lin 2008) which was developed by Chih-Chung Chang and Chih-Jen Lin (Chang, Hsu et al. 1999) are used to support the SVM classification in the experiment.

7.3.1 Training Step and Prediction Step

To prepare the training space, all images in the training set were used and their salient regions detected by using the DoG detector (described in Chapter 5 and Chapter 6). All regions were defined as background or foreground based on the centroid of the salient regions. A manually defined mask indicated where the object was located and salient regions with centroids on the object were marked as foreground salient regions and those not on the object were marked as background. Firstly, the SVM classifier was trained using 66,633 salient regions from 120 images to construct the training model.

Since sometimes, the relation between class labels and attributes is nonlinear. SVM uses a kernel function to handle it. Rather than fitting nonlinear curves to the data, the kernel function in SVM maps the data into a different space where a hyperplane can be used to do the separation. In this experiment, the radial basis function (RBF) (Hsu, Chang et al. 2003) is used as the kernel function. There are two parameters when using the RBF function: the penalty parameter C and the kernel parameter γ (Chang and Lin 2008).

Cross-validation is the process used to optimize these two parameters so that the classifier can accurately predict unknown data. Basically pairs of (C, γ) are tried and the one with the best cross-validation accuracy is picked. (Chang and Lin 2008) recommended that applying exponentially growing sequences of C and γ. (for

example, C = ; 2^{-5},2^{-3},…, 2^{15} , $\gamma = 2^{-15}$,2^{-13},…, 2^{3}) is a practical method to identify good parameters.

The cross-validation accuracy is the percentage of correctly classified salient regions in the cross-validation set. With the cross-validation method examined on the variety of parameter value, the best cross-validation accuracy on our dataset is at C = 8 and $\gamma = 2$ which gives the maximum accuracy rate at 96.65%. Table 7.2 shows some samples of the accuracy based on the parameter C and γ.

We picked C = 8.0 and $\gamma = 2.0$ to use in the RBF kernel to predict salient regions in the test set because this setting gave the highest cross-validation accuracy rates. The output of the foreground and background prediction step is used further in the image retrieval stage.

C	γ	**Cross-Validation Accuracy Rates (%)**
32.0	0.0078	85.61
0.5	0.0078	85.61
32.0	0.5	95.50
32.0	2.0	96.59
8.0	2.0	96.65

Table 7.2 Some examples of the cross-validation accuracy at the varied parameter C and γ

The class of the unknown salient regions from the testing data is predicted. The prediction is based on the training model which is derived from the training step. For the test set, there are 57,884 salient regions to determine from 120 images. The accuracy of the prediction is 93.45%

7.3.2 Image Retrieval

After the prediction by SVM, salient region filtering and selection was processed in the similar way as in Chapter 6. All salient regions in the background class were filtered out. Only up to fifty foreground regions with the lowest curvature value in ordering (see Section 4.1.2) were selected to represent the object in the image. The steps of image retrieval were the same processes as the experiment in Chapter 6.

In Figure 7.4, there is a performance comparison between the K-Nearest Neighbour method from Chapter 6, the SVM classification, the non-filtered method and the ideal method. All images using in all methods were represented by PCA-SIFT descriptors. It is noted that the ideal method is created to show the best performance of image retrieval if all salient regions of images are on the foreground. The program picked salient regions from the foreground location only. To confirm all salient regions located on the foreground area, the ground truth from the mask images were used.

7.4 Results

From the precision and recall graph, the performance of image retrieval by K-Nearest Neighbour and SVM classification is not different. Although both methods have precision less than the ideal method, but they are substantially better than the non-filter image retrieval.

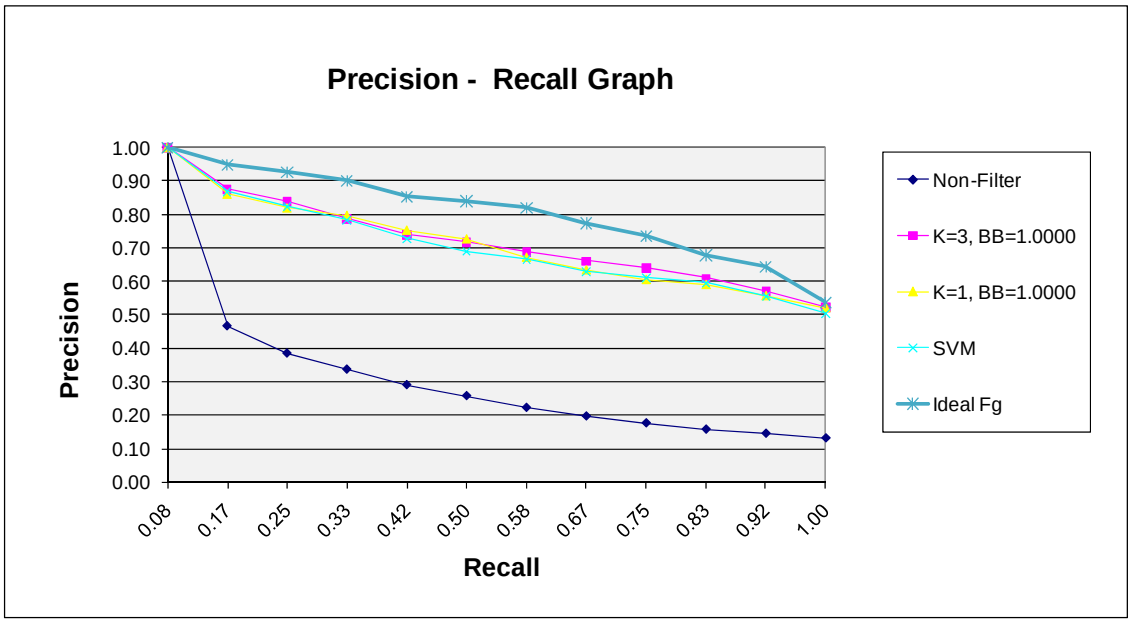


Figure 7.4 The precision and recall graph comparing the SVM method with other methods

7.5 Summary

In this chapter, the foreground-background classification by SVM has been investigated. In addition, using 36 dimensions of PCA-SIFT rather than the basic SIFT saves the calculation time for the training process and performs better performance on our dataset.

On this dataset, classification by SVM does not lead to retrieval performance which is significantly different to the K-NN classification shown in Chapter 6. However, the use of PCA-SIFT as the chosen feature vector does lead to a substantial improvement over the use of SIFT for filtering background salient regions for image retrieval.

Chapter 8 Conclusion and Future Work

This final chapter attempts to draw together and summarise the main conclusions of the previous chapters. There are a number of preprocessing approaches and techniques which are investigated in this thesis and the new approaches have been evaluated in the context of image retrieval applications. Finally, avenues for future research are presented following on from the ideas presented in the thesis.

8.1 Summary and Conclusions

In recent years, many problems of Content Based Image Retrieval (CBIR) have attracted the interest of researchers, especially on image descriptors, similarity measures and search optimization through databases. This thesis specifically focuses on the preprocessing of images for image retrieval with the aim of trying to improve the performance of retrieval. Using additional processes could augment the retrieval potential.

The thesis focuses on the problem of object retrieval when background influence over the object may occur. This leads to a decrease in the retrieval performance. On the assumption that image retrieval should be based on the specific object which users are interested in, we tried to extract the main area of an object in the image without the background information.

Automatic Image segmentation is the first aspect of the investigation. Segmentation algorithms including normalized cut segmentation, grid segmentation and region growing segmentation were used in an attempt to extract a central object in an image.

Image retrieval based on central object regions was compared using various image features; colour histogram, CCV, Hu moments and Gabor filters.

In the experiment, normalized cut segmentation was investigated first. However, region growing segmentation was selected to be the preprocessing for our application as it was shown that region growing segmentation performed more quickly and with fewer parameters than normalized cut segmentation.

Image Segmentation was able to extract most of the object especially on images with simple backgrounds. It was confirmed that image retrieval on the specific object with the reduction of the background influence was better than image retrieval on the whole image. Unfortunately, current automatic image segmentation was still not able to perfectly associate segmented regions to the actual objects that were being described.

Further investigations on image retrieval based on salient regions were introduced in Chapter 4. All images in the database were represented by local features from salient regions. To detect salient regions in images, the Difference-of-Gaussian (DoG) detector described in (Lowe 2004) was used. Performance of retrieval based on salient regions was compared with retrieval based on image segmentation. Precision and recall graphs were used for the comparison. In this section no attempt was made to filter out salient regions which were associated with the background. But the use of salient regions gave an improvement over retrieval using the whole image. Image retrieval based on salient regions was the main focus in the rest of the thesis.

In an image, salient regions can be detected in any locations. In the thesis we show that filtering background salient regions can be used to benefit image retrieval. The number of salient regions is reduced and leaves only relevant regions.

Chapter 5 proposed the first filtering approach based on background information. In the training process, a group of background clusters of salient regions was created. It was used to predict the background salient regions in the test images. Since only background samples were required for the training, it was not necessary for the object

in the query to be known. The experiment showed that image retrieval with filtering of salient regions was better than without filtering of salient regions.

In the case where foreground information was provided in the training set, the class of unknown salient regions could be determined by the class of their neighbours in the training set. After salient regions were categorized into background or foreground class, all salient regions classified as background were filtered and a number of salient regions with the highest probability of being on the foreground were selected as the representation of the image. The details of this method appeared in Chapter 6.

Chapter 7 presented the SVM technique in an attempt to improve the performance of the class prediction. The time of training was also reduced by changing the 128 dimensional SIFT descriptors to the 36 dimensional PCA-SIFT descriptors. In this approach the retrieval was performed in a similar way to that when using the K-NN classification in the previous chapter. The K-NN and SVM based classification led to very similar performance in image retrieval but the change to PCA-SIFT led to improvements over the use of the basic SIFT for background filtering.

The main contributions in this thesis are proposing a number of preprocessing approaches to the image retrieval community. Attempting to reduce the influence of the irrelevant information from images in the database has been achieved. We began with preprocessing by image segmentation in an attempt to extract the object from background. Then image retrieval based on salient regions was investigated and the preprocessing was focused on classifying salient regions into background or foreground classes. When only the background examples were available, the background cluster was constructed to predict background salient regions. K-NN classification and SVM classification were also applied to filter out background.

8.2 Future Work

The main future work would be to develop a better method for discriminating effectively between the object and background salient regions. There are a number of works that can be extended from the thesis which are listed by chapters.

8.2.1 Central Object Retrieval using Segmentation

To avoid background dominance, image segmentation was used to separate an object region from the background. In Chapter 3, results of image segmentation based on different features such as colour or edge information have been shown and compared. Currently there are an increasing number of techniques to improve the accuracy of image segmentation. In addition to using only one feature, some image segmentation algorithms combine multiple image features and include some techniques to improve segmentation results. For example, (Roth and Lange 2004) proposed the feature selection technique for image segmentation.

Improvements in split and merge techniques have reduced the chance to have over-segmentation and under-segmentation in images. This means split and merge techniques might bring an improvement to the object extraction result.

8.2.2 Image Retrieval using Salient Points

In order to retrieve regions from the image, a salient detector finds the contents of the prominent object in the image. In object-based image retrieval, the object is represented by parts of the object or regions of objects. Improving the success of the salient detection could improve the retrieval performance.

In all experiments in Chapter 4 - Chapter 7, salient points or salient regions are detected by the Difference-of-Gaussian (DoG) method described in (Lowe 2004). It would be interesting to investigate how other salient detectors such as Harris or MSER (Mikolajczyk, Tuytelaars et al. 2005) affect the performance of image retrieval. In (Vinyals, Ramisa et al. 2007) the combination of different region detectors improves object recognition. Therefore, it might also be expected to improve image retrieval on our dataset.

8.2.3 Background Filtering by Background Clusters

This method uses a group of background clusters from a number of background examples to define a salient region as foreground or background. As it was mentioned

in Chapter 5, the background clusters were created from the K-Means clustering method, and the automatic selection for a choice of K would be interesting to investigate further.

Feature descriptors were incorporated to represent salient regions once identified. Using more powerful feature descriptors may be another way to improve the performance of image retrieval.

8.2.4 Foreground-Background Classification by K-NN and SVM

As for the problem from the background clusters; the K-NN classification must find a choice of K for the K-NN method. Therefore a technique to automatically optimize K would be beneficial here too.

SVM cross-validation consumes a vast amount of time. A novel proposed method such as (An, Liu et al. 2007) possibly to alleviate waiting for processing. For a large dataset, to evaluate the cross-validation approximately by applying the well-known incomplete Cholesky decomposition technique would be interesting.

8.2.5 The Discriminative Regions

Another way to improve the performance of salient region filtering is to introduce techniques for modelling the object classes, in cases where these are known, rather than or in addition to the modelling of the background. The constellation model is an example to model an object of interest by ‘parts’ and ‘shape’. The automatic part selection proposed by Weber (Weber 2000) and Fergus (R. Fergus 2003) to use for learning the object class categorization would be interesting to apply with our research in the future.

8.3 Challenges in a Real World

The number of digital images and other media such as video clips and movies is increasing dramatically. Hence, the importance of information retrieval on very large datasets is becoming significant. Image retrieval in the real world often needs to deal

with a variety of complex objects and cluttered background images. It is still a challenging open problem for the IR community.

The preprocessing approaches presented in this thesis have been shown to enhance the IR performance on image datasets in which one foreground object is presented on a relatively straightforward background. For the complicated and very large image data mining, our preprocessing approaches might be extended by more advance techniques to make the approaches more sophisticated.

We believe that the combining of the preprocessing techniques developed here and more intelligent salient region detectors and descriptors including indexing and matching techniques should bring CBIR closer to the user's satisfaction in the near future.

Appendix A CBIR Systems

A.1 QBIC (TM) -- IBM's Query by Image Content

QBIC (Flickner, Sawhney et al. 1995) was the first significant content based retrieval system which has influenced many other CBIR systems. QBIC can use example images, user-constructed sketches and drawings, selected colour and texture patterns, camera and object motion and other graphical information to make queries on large image and video databases. It uses colour, texture, shape features, text descriptors or combinations of these to retrieve. QBIC has two main components, the database population and the query mechanism.

A.1.1 Database Population

The still Images

The Process of creating the database is called Database Population. The still images are reduced to thumbnails and annotated with any text information. The system has successfully used fully automatic unsupervised segmentation methods along with a foreground/background model for object identification. Moreover it provides semi-automatic tools for identifying objects by the user. The flood-fill method is one of techniques that were applied as an advanced tool. Users can click to allow fast object identification. Another tool which is based on the snake concept can help users to track the edges of objects. The curves that maximize the image gradient magnitude are the curves that the snake based tool finds.

Video Data

For video, three major components for database population are as follows,

- shot detection
- representative frame
- derivation of a layered representation of shots

First step is short detection. In the QBIC system, two methods proposed are shot detection based on global representations without any spatial information and the other one is shot detection based on measuring differences between spatially registered features like intensity differences. Using a robust normalized correlation measure it is possible to detect small motions and combine this with a histogram distance measure.

Finding changes in camera operation can be used to detect shots. General camera transformations (pan, zoom and illumination changes) can be represented as unknown affine 2 x 2 matrix transformations of the 2D image coordinate system and of the image intensities themselves.

For Shot boundaries, they can be defined on the basis of events such as appearance/disappearance of an object, diverse change in the motion of an object or similar events.

In the step of frame generation representation, each shot is represented by r-frames which can be used as follows:

1. Being treated as still images in which objects can be identified during database population.
2. Being the basic units initially returned in a video query during query step

The r-frame selection could be simple by selecting the first, the last or the middle shot. The system uses a synthesized r-frame created by seamlessly mosaicking all the frames in a given shot using the computed motion transformation of the dominant background.

This is used for some situations such as a long panning shot. For step of layered representation, the algorithm divides a shot into a number of layers. Each layer has its own 2D affine motion parameters and region of support in each frame.

A.1.2 Database Query

QBIC supports queries based on example images, user-constructed sketches and drawings, and selected colour and texture patterns.

Query by colour

User can select colour from a colour wheel. Users can not only select one colour, but percentages of two colours by adjusting sliders as well. The colour feature is a 3D vector of Munsell colour coordinates. When performing a query, the query MTM image histogram is matched to the image histograms in database. The difference histogram Z is computed, and a similarity measure is given by, $\|Z\| = Z^T AZ$ where A is a symmetrical matrix with A(i,j) measuring the similarity of colours i and j.

Query by Texture

User can specify texture by selecting from set of pre-stored example images called a sampler. The texture feature can be represented in term of coarseness, contrast and directionality. Coarseness measures texture scale (average size of regions that have the same intensity), contrast measures vividness of the texture (depends on the variance of the grey-level histogram), and directionality gives the main favoured direction of the image texture (depends on the number and shape of peaks of the distribution of gradient directions).

Query by Shape

The query specified by the drawn shape uses many features to match and characterize. These features are area (number of pixels in shape body), eccentricity (the ratio of the smallest and the largest eigenvalue), circularity (perimeter^2/Area), major axis orientation (the second order covariance matrix is computed using boundary pixels.

Major axis orientation is the direction of the matrix largest eigenvector), a set of algebraic moment invariants, and a set of tangent angles around the perimeter.

Query by Sketch

Sketch is a freehand drawing of the dominant lines and edges in the image. The sketch feature is an automatically extracted, reduced-resolution "edge map". A template-matching technique is used for sketch matching.

For multi-object queries, the combining of feature distances is used for matching.

A.2 BLOBWORLD

One of famous CBIR systems called Blobworld (Carson, Thomas et al. 1999) is created by the computer science division, University of California, Berkeley. It is a system for image retrieval based on finding coherent image regions (blobs). Since automatic image segmentation is a difficult problem, Blobworld models the joint distribution of colour, texture, and position features with a mixture of Gaussians. The Expectation Maximization (EM) algorithm is used to estimate the parameters. After the image is separated into regions, a description of each region's colour and texture is produced.

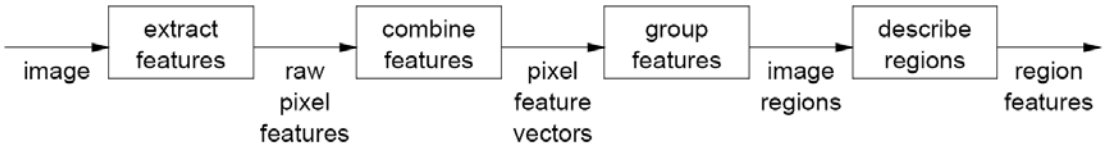


Figure A.1 The stages of Blobworld processing: From pixels to region descriptions

Figure A.1 shows the steps used in creating the Blobworld representation. Firstly, a proper scale is selected for each pixel, and the features for the pixel are extracted features. Secondly, pixels are grouped into regions by modelling the distribution of pixel features with a mixture of Gaussians using EM. Finally, regions are described by the colour distribution and texture.

A.2.1 Feature Extraction

Colour Feature Extraction

The colour feature is represented by a three-dimensional L*a*b colour space. Smoothing the colour features is useful to avoid over segmenting regions.

Texture Feature Extraction

First, texture extraction must provide a proper description of the texture parameters, and it must be computed in a neighbourhood. This can be solved by multi-orientation filter banks. The other requirement is concerned with scale selection. Unfortunately, texture descriptors computed at the wrong scale confuse the issue.

The proposed scale selection method is based on edge/bar polarity stabilization and the texture descriptors start from the windowed second moment matrix. Blobworld uses *polarity* (The measure of the extent to which the gradient vectors in a certain neighbourhood all point in the same direction)*, normalized texture contrast* and *anisotropy* over the region, as the 2D coordinate (contrast, contrast x anisotropy). The normalized texture contrast and anisotropy are derived from an orientation tensor defined as, $M(x,y) = G(x,y) * (\nabla I)(\nabla I)^T$ where $G(x,y)$ is a Gaussian low-pass filter and (∇I) is the image intensity. The anisotropy is defined as $a = 1 - \lambda_2 / \lambda_1$ and the normalized texture contrast, defined as $c = 2\sqrt{\lambda_1 + \lambda_2}$, where λ_1, λ_2 are the eigenvalues of *M*.

A.2.2 Grouping Pixels into Regions

The system groups the 8-dimensional feature space into regions by modelling the feature vector distribution with a mixture of Gaussians using an EM-algorithm. To improve the segmentation, it uses some post-processing.

A.2.3 Blobworld Interface and Matching

The user selects a blob from an image and specifies the relative importance of the blob features ('not', 'somewhat', 'very'). Since the user can select the internal representation, the feature weights are made in an easy and proper way.

To match between two regions by the colour histogram, the distance between their histogram is calculated.

$$d(h_1, h_2) = (h_1\text{-}h_2)^T A(h_1\text{-}h_2)$$

where h_1 and h_2 are colour histogram and $A = (a_{ij})$ is a symmetric matrix of weights representing the similarity between colour bins i and j.

For texture information, the distance between two texture descriptors is the Euclidean distance between their coordinates in representation space (contrast and anisotropy x contrast). The indexing of the Blobworld system is based on that of the QBIC system. However, it cannot select weight features automatically.

A.3 ARTISTE /SCULPTEUR

A.3.1 ARTISTE

Since there is a requirement for systems and techniques to provide effective remote access to European collections from museums and galleries, the European project called ARTISTE (An integrated Art Analysis and Navigation Environment) was established (Allen, Vaccaro et al. 2000).

The objective of the project is to develop and demonstrate the value of an integrated art analysis and navigation environment. Although collections are stored in separate databases and all have their own unique scheme for the metadata, ARTISTE makes it possible to search quickly and transparently by translating local metadata schemas to

common standards so that the individual collections are searched as though they were all a single entity.

Cross Collection

ARTISTE undertakes cross-collection searching between multiple databases or between museums. For the cross collection searching architecture, the abilities of a Search and Retrieve Web Service (SRW) was enlarged. An API was implemented to let the server access the ORDBMS and a UDM (user defined module) was used to produce feature vectors which were held as binary large objects (BLOBS). The resource description format (RDF) was used to define the syntax and semantics of standard metadata terms.

Query similar images by colour

A CCV (Colour Coherence Vector) method is used when the user is looking for similar images based on colour. Alternatively, the algorithm used will be the 'Mono-Histogram', if the user is either searching for colour images but submits a black and white image, or is searching for black and white images, regardless of whether a colour or black and white image is submitted.

Query similar images by pattern or texture

To find images that have a similar pattern to the image submitted, the algorithm called PWT (Pyramid Wavelet Transform) is used.

Query larger images from a sub-image

Sometime a query may be a sub-image. ARTISTE is able to identify the parent image and locate position. The method of Sub-Image Matching begins by partitioning images into a pyramid of image patched, extracting feature vectors and matching using the standard algorithms for each patch to find the best match. When the query is colour image, the algorithm will be Multi-Scalar Colour Coherence Vector (MCCV). On the other hand, if query is a black and white image or the user wants to search black and white images, the algorithm used will be Multi-Scalar Mono-Histogram (Chan, Martinez et al. 2001).

Moreover, ARTISTE has the ability to retrieve images that match with low quality monochrome query image such as fax images (Fauzi and Lewis 2002). To solve this problem, it uses the slow query by low-quality image (slow QBLI) method or the fast QBLI (uses Pyramid Wavelet Transform-based method).

A.3.2 SCULPTEUR

This project uses semantic web technologies and content and metadata based retrieval techniques for searching and exploiting the digital collections in museums. SCULPTEUR builds on the prior EU funded project, ARTISTE.

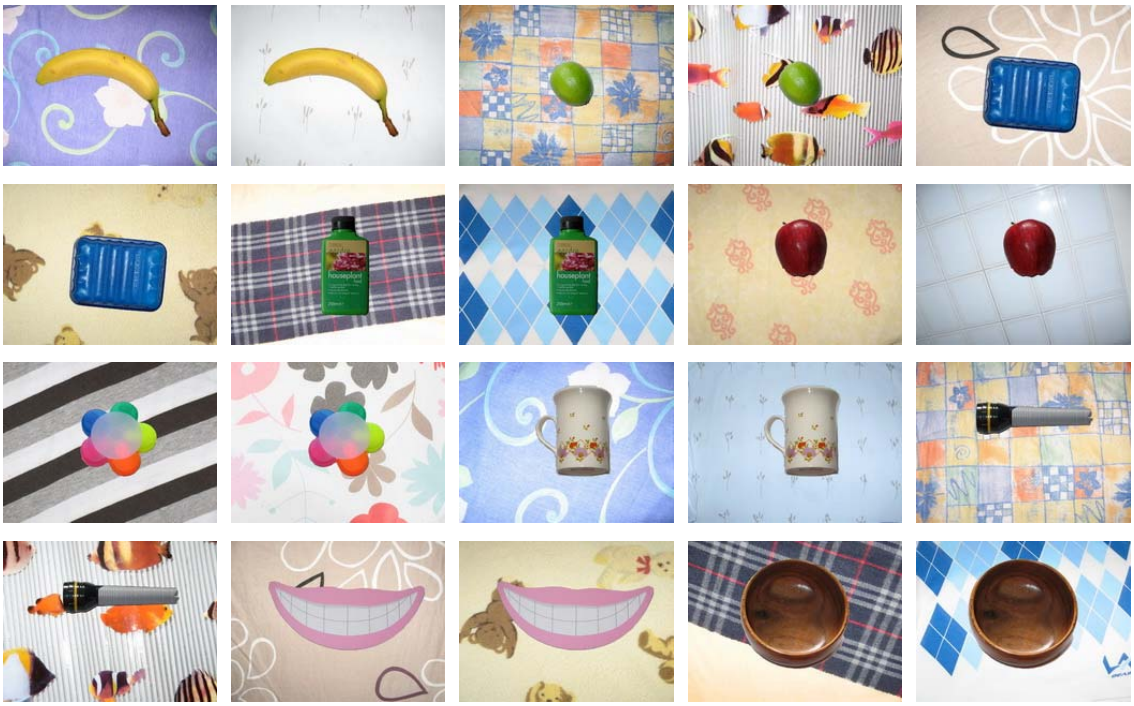
ARTISTE is the starting point for SCULPTEUR. SCULPTEUR increases the scope of the system to include 3-D digital representations of artefacts. The system is implemented as a web based client-server application. Retrieval from the system uses content-based retrieval; textual metadata based retrieval or semantic retrieval using ontologies which are implemented using Protege and held as RDF.

The ontology, which is built up as a semantic layer, is based on the conceptual reference model (CRM) developed by CIDOC. Semantic Web technologies are used to generate structure and link the data. Agents are being created to extract missing data instances from the web to fill the semantic layer. NLP and IR tools are used to identify information and structure sources. With the semantics, interoperability protocols are needed to let the system interoperate and provide seamless cross-collection searching.

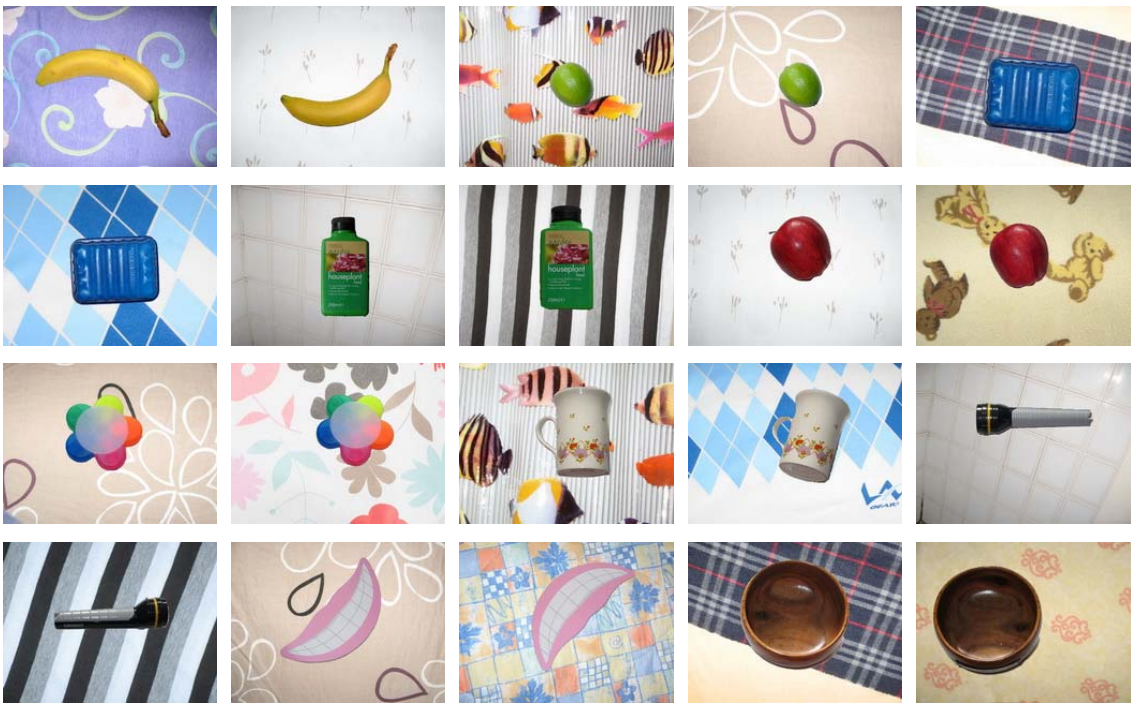
Appendix B Examples of Image Datasets

B.1 Examples of background images from the background collection in Chapter 5

B.2 Examples of images from dataset 1 in Chapter 5



B.3 Examples of images from dataset 2 in Chapter 5 and the test set in Chapter 6 and Chapter 7



Bibliography

A.Young, R., R. M. Lesperance and W. W. Meyer (2001). The Gaussian Derivative Model for Spatial-Temporal Vision: I. Cortical Model. *Spatial Vision (VSP)* 14(3): 261-319.

Aksoy, S. and R. M. Haralick (1998). Textural Features for Image Database Retrieval. *IEEE Workshop on Content-Based Access of Image and Video Libraries, in conjunction with CVPR'98*, Santa Barbara, CA, June,21 1998, pages 45-49.

Allen, P., R. Vaccaro and G. Presutti (2000). Artiste: An integrated Art Analysis and Navigation Environment. *Cultivate Interactive* (1).

An, S., W. Liu and S. Venkatesh (2007). Fast Cross-Validation Algorithms for Least Squares Support Vector Machine and Kernel Ridge Regression. *Pattern Recognition* 40: 2154 - 2162.

Anonymous. (2004). Groundtruth Database. Retrieved 24 JUNE, 2008, from http://www.cs.washington.edu/research/imagedatabase/groundtruth/.

Anonymous. (2005, 30 June 2005). Computer Vision Test Images. Retrieved 26 MAR, 2009, from http://www.cs.cmu.edu/~cil/v-images.html.

Anonymous. (2009). PASCAL Object Recognition Database Collection, Visual Object Classes Challenge. Retrieved 25 DEC, 2008, from http://www.pascal-network.org/challenges/VOC

Barnard, K., P. Duygulu, N. Freitas, Forsyth, D., D. Blei and M. I. Jordan (2003). Matching Words and Pictures. *Journal of Machine Learning Research* 3: 1107-1135.

Borenstein, E., E. Sharon and S. Ullman (2004). Combining Top-down and Bottom-up Segmentation. *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*. 4**:** 46.

Braun, M. and G. Petschnigg. (2002). Information Fusion of Flash and Non-Flash Images. Retrieved 10 July, 2004, from http://graphics.stanford.edu/~georgp/vision.htm.

Brun, A., Knutsson, H., Park, H., Shenton, M.E. and Westin, C.F. (2004). Clustering Fiber Traces Using Normalized Cuts. *MICCAI, 2004*.

Canny, J. (1986). A Computational Approach To Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions* 8(679-714).

Carson, C., M. Thomas, S. Belongie, J. M. Hellerstein and J. Malik (1999). Blobworld: A System for Region-Based Image Indexing and Retrieval. *Third International Conference On Visual Information and Information Systems*, Amsterdam, pages 509-516.

Chan, S., K. Martinez, P. Lewis, C. Lahanier and J. Stevenson (2001). Handling Sub-Image Queries in Content-Based Retrieval of High Resolution Art Images. *International Cultural Heritage Informatics Meeting*, pages 157-163.

Chang, C.-c., C.-w. Hsu and C.-J. Lin (1999). The Analysis of Decomposition Methods for Support Vector Machines. *the Workshop on Support Vector Machines, 16th International Joint Conference on Artificial Intelligence (IJCAI 99)*.

Chang, C. C. and C. J. Lin. (2008). LIBSVM - A Library for Support Vector Machines. from http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Deb, S. and Y. Zhang (2004). An Overview of Content-Based Image Retrieval Techniques. *The 18th International Conference on Advanced Information Networking and Applications (AINA)*, pages 59-64.

Elhabian, S. Y., K. M. El-Sayed and S. H. Ahmed (2008). Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art. *Recent Patents on Computer Science*. 1**:** 32-54.

Fauzi, M. and P. Lewis (2002). Query by Fax for Content Based Image Retrieval, for Challenge of Image and Video Retrieval. *International Conference on the Challenge of Image and Video Retrieval* London, United Kingdom, July 2002, Springer, pages 91-99.

Fawcett, T. (2004). ROC Graphs: Notes and Practical Considerations for Researchers. Palo Alto, USA, HP Laboratories.

Flickner, M., H. Sawhney, W. Niblack and J. Ashley (1995). Query by Image and Video Content: The QBIC System. *IEEE Computer* 28(9): 23-32.

Fraundorfer, F. and H. Bischof (2005). A Novel Performance Evaluation Method of Local Detectors on Non-planar Scenes. *Computer Vision and Pattern Recognition (CVPR)*, 20-26 June 2005, pages 33.

Goodall, S., P. H. Lewis, K. Martinez, P. A. S. Sinclair, F. Giorgini, M. J. Addis, M. J. Boniface, C. Lahanier and J. Stevenson (2004). SCULPTEUR: Multimedia Retrieval for Museums. Image and Video Retrieval. *The Third International Conference on Image and Video Retrieval (CIVR'04)*, Dublin, Ireland, July 21-23, 2004., pages 638-646.

Griffin, G., A. Holub and P. Perona (2007). The Caltech-256.

Hare, J. S. and P. H. Lewis (2004). Salient Regions for Query by Image Content. *The Thrid International Conference on Image and Video Retrieval (CIVR'04)*, Dublin, Ireland, July 21-23, 2004 pages 317-325.

Hare, J. S. and P. H. Lewis (2005). On Image Retrieval using Salient Regions with Vector-Spaces and Latent Semantics.

Hoiem, D., R. Sukthankar, H. Schneiderman and L. Huston (2004). Object-Based Image Retrieval Using the Statistical Structure of Images. *IEEE Conference on Computer Vision and Pattern Recognition*.

Hsu, C.-T. and M.-C. Shih (2002). Content-Based Image Retrieval by Interest Points Matching and Geometric Hashing. *Electronic Imaging and Multimedia Technology III*, September 2002, pages 80-90.

Hsu, C.-W., C.-C. Chang and C.-J. LinC (2003). A Practical Guide to Support Vector Classification.

Hu, M. K. (1962). Visual Pattern Recognition by Moment Invariant. *IRE Transaction on Information Theory* 8: 179-187.

Huang, J., S. R. Kumar, M. Mitra, W. J. Zhu and R. Zabih (1997). Image Indexing Using Color Correlograms. *Computer Vision and Pattern Recognition*, pages 762-768.

Idrissi, K., G. Lavoué, J. Ricard and A. Baskurt (2004). Object of Interest-Based Visual Navigation, Retrieval, and Semantic Content Identification System. *Computer Vision and Image Understanding; Special issue on color for image indexing and retrieval* 94(1-3): 271 - 294

Jain, A. and S. Pankanti (2000). Fingerprint Classification and Matching. *Handbook for Image and Video Processing*. A. Bovik, Academic Press.

Joliffe, I. T. (1986). *Principal Component Analysis*, Springer-Varlag.

Ke, Y. and R. Sukthankar (2004). PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *the Conference on Computer Vision and Pattern Recognition*, Washington, USA, pages 511-517.

Kim, D.-H., J.-W. Song, J.-H. Lee and B.-G. Choi (2007). Support Vector Machine Learning for Region-Based Image Retrieval with Relevance Feedback. *ETRI Journal* 29(5): 700-702.

Kim, S., S. Park and M. Kim (2003). Central Object Extraction for Object-Based Image Retrieval. *CIVR 2003 : International Conference on Image and Video Retrieval*, Urbana Champaign, IL, pages 39-49.

Kohavi, R. and F. Provost (1998). Glossary of Terms. *Machine Learning*(30): 271-274.

L. Fei-Fei, R. F., and P. Perona (2004). Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *CVPR 2004, Workshop on Generative-Model Based Vision.*

Lewis, P. H., K. Martinez, F. S. Abas, M. F. Ahmad Fauzi, M. Addis, C. Lahanier, J. Stevenson, S. C. Y. Chan, B. Mike J. and G. Paul (2004). An Integrated Content and Metadata Based Retrieval System for Art. *IEEE Transactions on Image Processing* 13(3): 302-313.

Ling Shao, M. B. (2006). Specific Object Retrieval Based on Salient Regions. *Pattern Recognition* 39: 1932-1948.

Loupias, E. and N. Sebe. (1999). Wavelet-based Salient Points for Image Retrieval. from http://rfv.insa-lyon.fr/~loupias/points/.

Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. *Proceeding of the International Conference on Computer Vision*, Corfu.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2): 91-110.

Ma, W.-Y. and B. S. Manjunath (1999). NeTra: A Toolbox for Navigating Large Image Databases. *Multimedia Systems* 7(3): 184-198.

Manjunath, B. S. and W. Y. Ma (1996). Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI - Special issue on Digital Libraries),* 18(8): 837-42.

Marques, O. and B. Furht (2002). *Content-Based Image and Video Retrieval*. Massachusetts, KLuwer Academic.

Martinez, A. M., P. Mittrapiyanuruk and A. C. Kak (2004). On Combining Graph-Partitioning with Non-Parametric Clustering for Image Segmentation. *Computer Vision and Image Understanding* 95: 72–85.

Mikolajczyk, K. and C. Schmid (2005). A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 27(10): 1615-1630.

Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. V. Gool (2005). A Comparison of Affine Region Detectors. *International Journal of Computer Vision* 65(1-2): 43-72.

Moreels, P. and P. Perona (2005). Evaluation of Features Detectors and Descriptors Based on 3D Objects. *10th IEEE International Conference on Computer Vision (ICCV 2005)*, 17-21 Oct. 2005 (publish), pages 800-807.

Müller, H., S. Marchand-Maillet and T. Pun (2002). The Truth about Corel - Evaluation in Image Retrieval. *The Challenge of Image and Video Retrieval (CIVR2002)*, London, UK.

Ortega, M., Y. Rui, K. Chakrabarti, S. Mehrotra and T. S. Huang (1997). Supporting Similarity Queries in MARS. *Proceedings of the 5th ACM International Multimedia Conference*, Seattle, Washington, 8-14 Nov. 1997, pages 403-413.

Pass, G., R. Zabih and J. Miller (1996). Comparing Images Using Color Coherence Vectors. *ACM Conference on Multimedia*, Boston, Massachusetts, pages 65-74.

Pinto, N., D. D. Cox and J. J. DiCarlo (2008). Why is Real-World Visual Object Recognition Hard? . *PLoS Computational Biology* 4(1).

Ponce, J., T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, C. K. I. Williams, J. Zhang and A. Zisserman (2006). Dataset Issues in Object Recognition. *Toward Category-Level Object Recognition (Sicily Workshop 2006)*, Sicily, LNCS.

Pontil, M. and A. Verri (1998). Support Vector Machines for 3D Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(6): 637-646.

Price, K. (2009, 24 MAR 2009). Computer Vision Bibliography. Retrieved 25 MAR, 2009, from http://datasets.visionbib.com/info-index.html.

R. Fergus, P. P., and A. Zisserman (2003). Object Class Recognition by Unsupervised Scale-invariant Learning. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*.

Ren, J., Y. Shen, S. Ma and L. Guo (2004). *Applying Multi-class SVMs into Scene Image Classification*, Springer Berlin / Heidelberg.

Rodhetbhai, W. and P. H. Lewis (2007). Salient Region Filtering for Background Subtraction. *Advances in Visual Information Systems*, Springer Berlin / Heidelberg. 4781/2007**:** 126-135.

Roth, V. and T. Lange (2004). Adaptive Feature Selection in Image Segmentation. *Pattern Recognition*, Springer Berlin / Heidelberg. 3175/2004**:** 9-17.

Rusinol, M. and J. Llados (2008). Word and Symbol Spotting Using Spatial Organization of Local Descriptors. *Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on* pages 489-496.

Schmid, C. and R. Mohr (1997). Local Grayvalue Invariants for Image Retrieval. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 19(5): 530-535.

Sclaroff, S., A. Pentland and R. W. Picard (1996). Photobook: Content-Based Manipulation of Image Databases. *International Journal of Computer Vision* 18: 233-254.

Sebe, N., Q. Tian, E. Loupias, M. S. Lew and T. S. Huang (2002). Evaluation of Salient Point Techniques. *International Conference on Image and Video Retrieval (CIVR'02)*, London, UK, July, 2002, pages 367-377

Shao, H., T. Svoboda, V. Ferrari, T. Tuytelaars and L. V. Gool (2003). Fast Indexing for Image Retrieval based on Local Appearance with Re-ranking. *Image Processing, 2003. ICIP 2003*, pages 737-740.

Shapiro, L. G. and G. C. Stockman (2001). *Computer vision*, Prentice Hall.

Shi, J. and J. Malik (1997). Normalized Cuts and Image Segmentation. *IEEE Conf. Computer Vision and Pattern Recognition*: 731-737.

Shi, J. and J. Malik (1998). Motion Segmentation and Tracking Using Normalized Cuts. *Int. Conf. Computer Vision*, Bombay, India, Jan 1998, pages 1154-1160.

Shi, J. and J. Malik (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8): 888-905.

Shioyama, T., H. Wu and S. Mitani (1999). Segmentation and Object Detection with Gabor Filters and Cumulative Histograms. *10th International Conference on Image Analysis and Processing (ICIAP'99)*, pages 412.

Shyu, B. K., C. Brodley, A. Kak, C. Shyu, J. Dy, L. Broderick and A. M. Aisen (1999). Content-Based Retrieval from Medical Image Databases: A Synergy of Human Interaction, Machine Learning and Computer Vision. *The Sixteenth National Conference on Artificial Intelligence*, Orlando, FL, July 18-22, 1999, pages 760-767.

Smeulders, A. W. M., M. Worring, S. Santini, A. Gupta and R. Jain (2000). Content Based Image Retrieval at The End of The Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(12): 1349-1380.

Smith, J. R. and S.-F. Chang (1995). Single Color Extraction and Image Query. *International Conference on Image Processing (ICIP'95)*, pages 528 - 531.

Smith, J. R. and S.-F. Chang (1996). VisualSEEk: A Fully Automated Content-Based Image Query System. *ACM Multimedia Conference*, Boston, MA, November 1996.

Stricker, M. A. and M. Orengo (1995). Similarity of Color Images. *Proceedings of Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381-392.

Tian, Q., N. Sebe, M. S. Lew, E. Loupias and T. S. Huang (2001). Image Retrieval using Wavelet-based Salient Points. *Journal of Electronic Imaging, Special Issue on Storage and Retrieval of Digital Media* 10(4): 835-849.

Tombre, K. and B. Lamiroy (2003). Graphics Recognition - From Re-Engineering to Retrieval. *Seventh International Conference on Document Analysis and Recognition, ICDAR*, pages 148-155.

Traina, A. J. M., J. Marques and C. T. Jr (2006). Fighting the Semantic Gap on CBIR Systems through New Relevance Feedback Techniques. *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pages 881 - 886

Tremeau, A. and N. Borel (1997). A Region Growing and Merging Algorithm to Colour Segmentation. *Pattern Recognition* 30(7): 1191-1203.

Tuytelaars, T. and L. J. V. Gool (1999). Content-Based Image Retrieval based on Local Affinely Invariant Regions. *Proc. Visual '99: Information and Information Systems*: 493-500.

Vincent, L. and P. Soille (1991). Watersheds in Digital Spaces: An Efficient Algorithm based on Immersion Simulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 583-598.

Vinyals, M., A. Ramisa and R. Toledo (2007). An Evaluation of an Object Recognition Schema Using Multiple Region Detectors. *Frontiers in Artificial Intelligence and Applications*, IOS Press, pages 213-222.

Voorhees, E. M. and D. K. Harman (2005). *TREC: Experiment and Evaluation in Information Retrieval*, The MIT press.

Wang, J. Z., J. Li and G. Wiederhold (2001). SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(9): 947-963.

Weber, M. (2000). Unsupervised Learning of Models for Recognition, Caltech.

Wu, Z. and R. Leahy (1993). An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 15(11): 1101-1113.

Zhang, D. and G. Lu (2001). A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures. *Intelligent Multimedia and Distance Education (ICIMADE01)*, Fargo, ND, USA, 1-3 June 2001.

Zhang, D. and G. Lu (2003). Evaluation of Similarity Measurement for Image Retrieval. *Proceedings of the 2003 International Conference on Neural Networks and Signal Processing*, 14-17 Dec. 2003, pages 928-931.

Zhang, H., R. Rahmani, S. R. Cholleti and S. A. Goldman (2006). Local Image Representations Using Pruned Salient Points with Applications to CBIR. *the 14th Annual ACM International Conference on Multimedia (ACM Multimedia)*, October 2006.

Zhang, L., F. Lin and B. Zhang (2001). Support Vector Machine Learning for Image Retrieval. *Proceedings of International Conference on Image Processing, 2001*, 7-10 Oct. 2001, pages 721-724.