Image Retrieval using SURF Features

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Ashish Kumar
(800932005)

Under the supervision of:
Mrs. Shalini Batra
Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2011

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*Image Retrieval using SURF Features*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mrs. Shalini Batra* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Kumar

(Ashish Kumar)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mrs. Shalini Batra)
Assistant Professor
Computer Science and Engineering Department

Countersigned by

(Dr. Maninder Singh)
Head 15/6/11
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

I would like to take this opportunity to express my thanks and gratitude to my thesis supervisor, **Mrs. Shalini Batra**, Computer Science and Engineering Department, whose constant guidance, insightful and unfaltering help was invaluable for the completion of the thesis. She has helped me to explore the vast variety of topics associated with the thesis in an organized manner and provided guidance to work towards a research-oriented venture.

I am also thankful to **Dr. Maninder Singh**, Head of Department, Computer Science and Engineering Department and **Mr. Karun Verma,** P.G. Coordinator, for their advice and guidance during the course of thesis work.

I would also like to express my thanks to my colleagues for their encouragement and support and to all those who have posted helpful tutorials, guides on topics related to the thesis on the internet.

Most importantly I would like to thank God for his blessings and my family for their constant support and encouragement which helped me through the various setbacks and hardships I faced during the completion of thesis.

(Ashish Kumar)

(800932005)

Abstract

Modern technology has made image capturing cameras ubiquitous and increased accessibility to the Internet and Internet based services and soaring popularity of social networking sites for sharing pictures has resulted in large amount of images being shared over the Internet. The availability of cheap storage has allowed individuals as well as Web companies to maintain and host large collections of images. The huge collections of images pose a challenge in terms of efficient retrieval of desired images to both users and Web companies. The traditional databases have been found unfit for processing and retrieving images as they are incapable of capturing the information within the images. Traditional databases were designed with fixed text based information structures in mind and can store image information like resolution, metadata and even images as binary files but they cannot do image processing, classification and matching of images on the basis of information stored in them. This limitation of traditional methods is remedied using methods that are able to classify and retrieve images based upon the content that is seen by the end user and not the miniscule data captured by image metadata.

The thesis discusses the implementation of image retrieval method used using image content as the differentiator between images. Also, discussed is a reproducible method to measure the performance of the proposed approach as well as the results of the performance of the image retrieval method on a standard image dataset.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid advances, innovation and refinement of technology in modern electronics has led to changes in the way visual data is captured, stored and accessed. There have been manifold changes in the world of image capture, firstly the miniaturization of capture devices like cameras, which has led to their incorporation into existing as well as emerging technologies. Secondly, storage devices have increased in capacity which together with their plummeting price as well as that of image capturing devices has led to creation of massive personal and commercial image collections being either privately stored or commercially made available on the Web thus people maintain and share collection of pictures on various social sites. Thirdly, the process of digitization of records has become prevalent almost everywhere due to the ease of conversion to digital format, storage, data retrieval. Fourthly, the advent of low cost Internet access and its widespread availability has provided capability to people to share images as and when they capture them. The amount of data generated due to all these factors is of unprecedented scale.

The retrieval of images from these large collections has created a challenge of efficient retrieval of correct images as specified by the user. The processing of images for the is not important just for purposes of information retrieval but also for application of content data mining techniques by companies hosting these images on the Web and seeking to monetize the content by not only providing image search results but also by mining information in images for advertising, user profiling and other purposes.

The pictures are generally lacking in context as unlike other multimedia like video or audio, consecutive images tend to be random pieces of captured data. Manual annotation of tagging relevant data is possible but this is exhausting and error prone method of indexing data for retrieval. It also tends to be problematic when people of other language than the one in which annotation was done try to search for images. Each picture contains a tremendous amount of information, for example, the picture in Fig 1.1 contains people, road, sky, trees, mountains, clouds, buildings, billboards, advertisement banners and a statue to name just a few of the prominent objects from an annotative point of view and may be of interest to a user.

Figure 1.1 A panoramic image of The Ridge, Shimla [1]

Most image formats support metadata storage within an image file. The metadata store various information about the image like Global Positioning System (GPS) coordinates of the place where the image was captured, time when the picture was taken, model of camera etc. This data is generally not enough to make reasonably detailed deduction to the contents of the image. It is preferable and much better to provide a visual search system that allows a person to search an image using an image similar to searching for text using text approach [2]. It then becomes more intuitive for the user to refine his results on the basis of the results that are returned to him rather than trying to use textual description of image content, which has its own problems of natural language ambiguities as well as the problem of conversion of text annotations from one language to other.

1.1 Computer Vision

Computer Vision is the field of computer science that aims to provide vision capabilities to computer systems that mimic human visual perception and is the enterprise of automating and integrating a wide range of processes and representations for vision perception. It includes as parts many techniques that are useful by themselves, such as image processing (transforming, encoding, and transmitting images) and statistical pattern classification (statistical decision theory applied to general patterns, visual or otherwise). It includes techniques for geometric modeling and cognitive processing [3].

1.2 Content Based Image Recognition

Content Based image Recognition (CBIR) as the name implies is an image retrieval system that is based on the contents of the image. It is the application of computer vision technology for purposes of object identification, object recognition and image retrieval from a database of images using image content like color, color gradients, color blobs, texture, edges etc.

Digital images are being produced in many areas of commerce, government, academia, and hospitals. Some of these collections are the product of digitizing existing collections of real physical photographs, diagrams, drawings, paintings, and prints and some are collection of original digital records. The usual way of searching these collections was by keyword indexing, or by browsing the contents manually. Digital image databases however, open the way to content-based searching. With the ever increasing number of sites on internet, easy accessibility and the amount of material that is being added every day, it has become more and more imperative to be able to search for the relevant information efficiently.

Image retrieval systems as part of search engines on the Web have become very popular. The image retrieval systems based on search engines are capable of providing image results on basis textual input describing the image. Some Web based popular image retrieval systems like Google Images by Google, Bing Images by Microsoft use metadata like annotations or information from the files in which the image is embedded. TinEye is one of the few online image search solutions that provide reverse image matching. It matches without textual metadata like the text surrounding the image or annotations for the image, but it can only detect matches to the input image whether it is cropped, resized or edited and not images similar to the original [4].

1.3 Digital Image

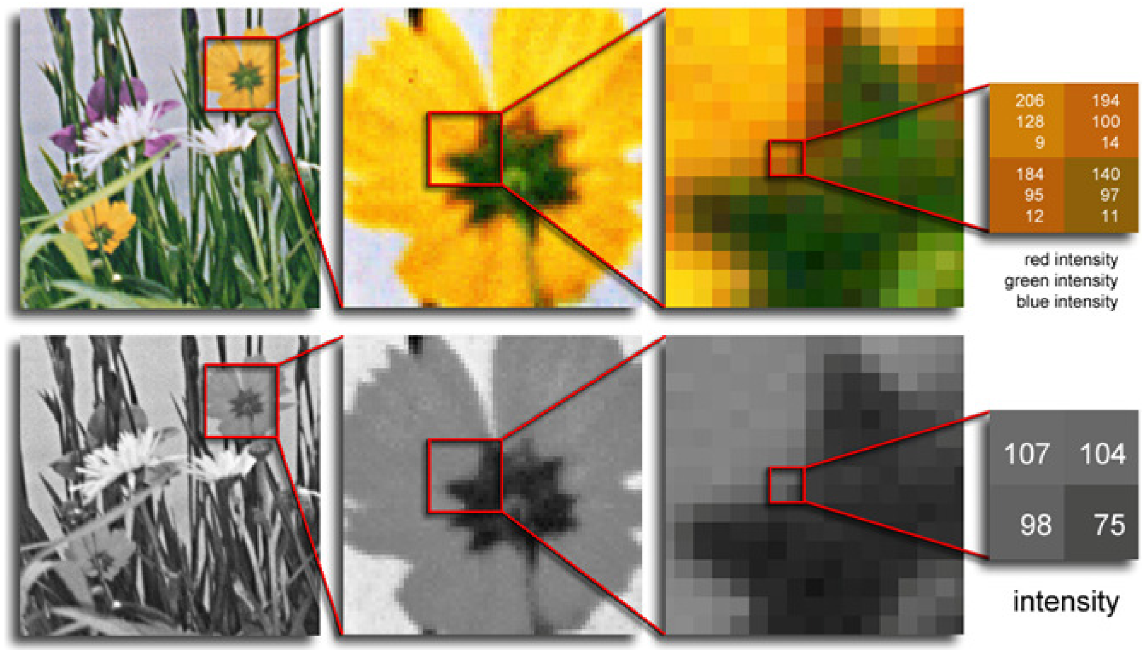

Figure 1.2 A color and grayscale image with highlighted pixel and their values [5]

A Digital Image is a snapshot of two dimensional visual such as a scene or scanned data from documents, such as photographs, manuscripts, printed texts, and artwork. The image is reproduced digitally by measuring the color at large number of points and creating an approximation of the image. The digital image is sampled and mapped as a grid of dots or picture elements (pixels). Each pixel is assigned a tone value (black, white, shades of gray or color) by using a fixed number of bits to represent the information stored, also known as the bit depth of the image. Generally each pixel is stored as 24 bits of information with 8 bits each for Red, Blue and Green (RGB), the primary colors, giving

2^8 (0-255) levels of color to each primary color. A 32 bit pixel adds 8 bits to RGB for alpha or transparency value of the pixel. The numerical values for each pixel are stored as per the image format standard with the application of compression techniques as necessary. The image file header contains information on how data is stored in images, the corresponding color space, the compression technique used and the method for its interpretation, so that it may be displayed on screen or printed.

Images are of two types:

- **i.** Raster Images
- **ii.** Vector Images

1.3.1 Raster Images

Figure 1.3 (a) Bitmap image [6]

Raster images store image data in a structured format dependent on the image format and when displayed on an electronic monitor is bit for bit copy of the stored image file. These are the images that people capture using cameras or when using a scanning device to generate an electronic copy of a document, art or photograph. These images are resolution dependent i.e. the resolution of an image directly affects the quality and retrievability of information in an image. The Fig 1.3 (a) is basically a raster image of a character from an old console game which shows obvious pixelization due to resizing of the image. Raster images are invariably larger in size than vector images. Usually, a raster image is referred to as a digital image.

Generally, raster images are stored in two types of formats based on the compression techniques applied on them:

i. Lossless format

   The image is stored using compression techniques that don't discard any data from the image. These images have the highest fidelity and accurate reproduction of the image captured within the limitation of the image capturing device. Portable Network Graphics (PNG) is an example of lossless image format [7].

ii. Lossy format

   The image is compressed using techniques that discard data during compression of image. The data is discarded on the basis of certain properties of human visual perception system, so the apparent loss of information from image is not discernible easily, at the same time reducing image size drastically. The popular lossy image format commonly known as JPEG is actually acronym to the name of the standards body (Joint Photographic Experts Group) with the formal name of the standard being ISO/IEC IS 10918-1 | ITU-T Recommendation T.81 [8].

Image formats also allow for metadata like annotations and GPS coordinates, camera model information with which the picture is taken, software with which the image is last modified and the time at which the image is taken is stored within the image.

1.3.2 Vector Images

Figure 1.3 (b) Vector image [6]

Vector images are made up of lines, curves and shapes of polygons as defined by mathematical equations to represent objects that constitute an image. These images are resolution independent and can be scaled, transformed and rotated in any manner without loss of information. Fig 1.3 (b) is a vector conversion of image in Figure 1.3 (a) and looks much better than it. Text can be represented using vector images in the form of fonts. Some of the disadvantages with vector images are that they cannot be used to depict pictures with continuous tones or photorealistic images. The most popular storage format for vector graphics is Scalable Vector Graphics (SVG). It is a widely-deployed royalty-free graphics format developed and maintained by a public group, World Wide Consortium (W3C) SVG Working Group [9]. Some other formats for storage of vector Adobe Illustrator (AI), CorelDraw (CDR), Corel Exchange (CMX) and fonts like PostScript Fonts and TrueType Fonts.

1.4 Image Processing

Image processing is any signal processing done on an input image to get various transformations of color, shade, tone, resolution etc. or information about the content of the input image in the form of a color histogram etc.

Image processing is mainly done for the purposes of [10]:

- **i.** improving the pictures representation for human perception
- **ii.** making it suitable for processing by a machine

The image processing is of two types:

- **i.** Analog Image processing

  It is the processing of two dimensional analog signals which make up an image. It deals with the hardware circuits, filters and optical and mechanical methods used to capture image data. Since, it involves actual hardware which the cost of image processing increases and it is not as popular as digital image processing for the processing of image data. Hardware solutions are important for embedded systems where due to power constraints general processors cannot be used for running image processing algorithms.

ii. Digital Image processing

It is the use of algorithms to perform image processing on digital images. It is a part of digital signal processing and has seen more development than analog image processing as the computer systems have become less expensive and computationally more powerful. Also, the algorithms that may be applied for digital processing are more numerous than those for analog processing allowing for processing not feasible using analog methods. Some commonly seen image processing methods are image resizing, color correction, color space conversion. Apart from this digital image processing is used for:

a. Image Restoration

Images that contain corrupted visual data like scans of old damaged photographs or images that have been digitally altered can be reverted to almost original condition using image processing techniques. In Fig 1.4 (a) we can see various white patches in the image which have been removed as shown in Fig 1.4 (b). Some other techniques used are removal of noise, removal of blur.

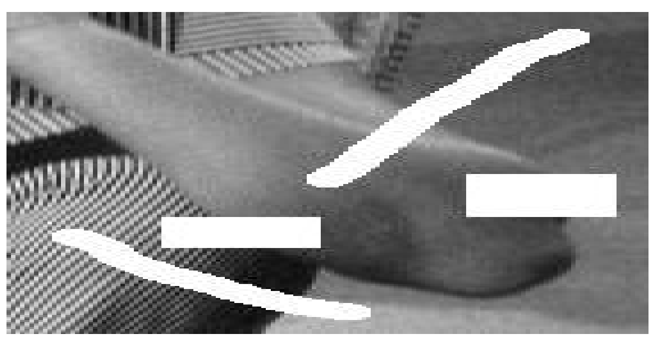

Figure 1.4 (a) Damaged Image [11]

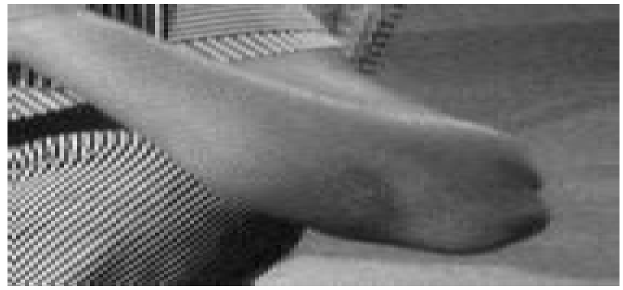

Figure 1.4 (b) Repaired Image [11]

b. Image Enhancement

Image processing is also used to enhance the properties of the image so that it becomes aesthetically more pleasing for humans. In Fig 1.5 (a) we can see a smooth image of an alley which has been sharpened as shown in Fig 1.5 (b), the sharpened image looks crisp and evidently offers more details as can be seen on the walls on both sides of the alley and the trees at the top of the image that are more prominent in Fig 1.5 (b).

Figure 1.5 (a) Original Image [10]

Figure 1.5 (b) Sharpened Image [10]

c. Image Feature Extraction

Figure 1.6 (a) Original Image [10]

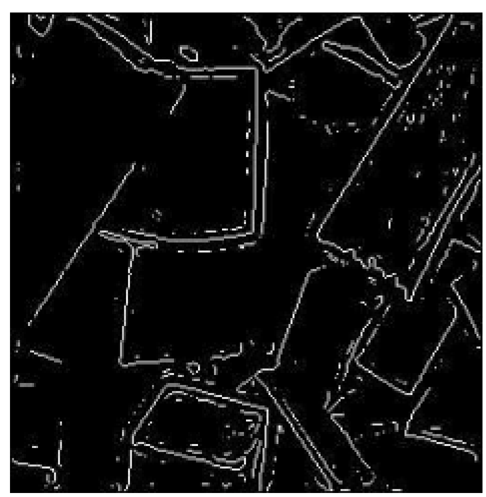

Figure 1.6 (b) Edges in image [10]

Digital image processing allows the use of much more complex algorithms for image processing, and hence, can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means. In particular, digital image processing is the only practical technology for computer vision, feature extraction, pattern recognition. In Fig. 1.6 (b) we can see the edge outlines of the objects depicted in Fig. 1.6 (a). These edges may be used to distinguish between different objects by a machine for object identification, tracking, matching and other purposes.

1.5 Machine Learning

Learning is defined in the Webster's dictionary as follows :

i. to gain knowledge or understanding of, or skill in by study, instruction or experience
ii. learning a set of new facts
iii. learning to do something
iv. improving ability of something already learned

Machine Learning is the principle of learning applied to computer systems and is an area of artificial intelligence that deals with the study of algorithms that improve automatically through experience [3]. The aim is to let the computer system to detect and adapt to patterns in the data as it processes it and optimize the process simultaneously.

Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more effectively the next time. Machine learning is needed when [17]:

i. Some tasks cannot be defined well except by example, i.e. we might be able to specify input/output pairs but not a concise relationship between inputs and desired outputs. We want machines to be able to adjust their internal structure to produce correct outputs for a large number of sample inputs and thus suitably constrain their input/output function to approximate the relationship implicit in the examples.

ii. It is possible that hidden among large piles of data are important relationships and correlations. Machine learning methods can often be used to extract these relationships (data mining).

iii. Human designers often produce machines that do not work as well as desired in the environments in which they are used. In fact, certain char-acteristics of the working environment might not be completely known at design time. Machine learning methods can be used for on-the-job improvement of existing machine designs.

iv. The amount of knowledge available about certain tasks might be too large for explicit encoding by humans. Machines that learn this knowledge gradually might be able to capture more of it than humans would want to write down.

v. The system keeps changing its state with respect to the inputs as well as the time over which such inputs are given. New knowledge about tasks is constantly being discovered by humans.Vocabulary changes. There is a constant stream of new events in the world. Continuing redesign of AI systems to conform to new knowledge is impractical, but machine learning methods might be able to track much of it.

1.5.1 Type of machine learning

The type of machine learning is dependent on the way feedback is presented to the learning algorithm to help it learn from the training data.

i. **Supervised Learning**

It involves creation of a classifier given a classified set of classified training examples. The classifier's job is to assign a class to an object. The algorithm is first presented with training data which consists of examples which include both the inputs and the desired outputs, thus enabling it to learn a function. The algorithm should then be able to generalize from the presented data to unseen examples. In situations where there is a cost to labeling data, a method known as active learning may be used, where the learner chooses which data to label [18][19].

ii. **Unsupervised Learning**

It is used with data that does not have class labels attached to it. The algorithm is presented with examples from the input space. Clustering algorithms are usually used for unsupervised learning. Clustering algorithms follow mainly two approaches of either creating hierarchical clusters where similar information is at different levels or non-hierarchical clusters where the information is stored in disjoint clusters [18][19].

iii. **Reinforcement Learning**

An agent explores an environment and at the end receives a reward, which may be either positive or negative. In effect, the agent is told whether he was right or wrong, but is not told how. Examples include playing a game of chess (you don't know whether you've won or lost until the very end) [19].

Desirable Features

There are many machine learning algorithms and the choice of using one over another is generally dictated by the task in which the algorithm will be used. But some generally desirable features, favored across all machine learning algorithms are [19]:

i. Simple solutions are appropriately favored over complicated ones.
ii. Algorithm is powerful enough to learn the solution, of a given problem.
iii. Stable to parameter variations of the problem.
iv. Converges to a solution in finite time.
v. Scales reasonably with the number of training examples, the number of input features and the number of test samples.

1.5.2 k-Nearest Neighbor

In pattern recognition, the k-nearest neighbor algorithm (k-NN) [20] is a method for classifying objects based on closest training examples in the feature space, in other words it is the method of finding in a D dimensional space the nearest point to a set of N points such that the distance between them is minimum. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm classifies an object by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). The k-NN algorithm has advantages and disadvantages like any other method, which are as follows [20]:

Advantages

- **i.** It is analytically easier to manage.
- **ii.** The implementation is simple relative to other algorithms.
- **iii.** For large number of samples it is nearly optimal.
- **iv.** Uses local information which can yield highly adaptive behavior
- **v.** The parallel implementation of the algorithm is relatively easier to other algorithms.

Disadvantages

- **i.** Large memory storage requirements, all the results need to be stored till algorithm completion.
- **ii.** The recall is computationally intensive to calculate.
- **iii.** It has problems dealing with high dimension data.

The simple approach of finding the nearest point becomes computationally impractical as the number of points N and the number of dimensions D increase. There are two approaches to counter this and speed up the nearest neighbor search. They are as follows [20]:

- **i.** Bucketing
- **ii.** kd-Trees

The algorithm finally used, builds upon kd-trees, so they are explained further below:

1.5.3 kd-Trees

Kd-tree data structure was proposed in 1975 as a multidimensional binary search tree (or kd-tree, where k is the dimensionality of the search space) as a data structure for storage of information to be retrieved by associative searches. In addition to its storage efficiency, a significant advantage of this structure is that a single data structure can handle many types of queries very efficiently [21]. It is a binary tree that recursively splits the whole input space into partitions, in a manner similar to a decision tree acting on real-valued inputs. Each node in the kd-tree represents a certain hyper-rectangular partition of the input space; the children of this node denote subsets of the partition. Hence, the root of the kd-tree is the whole input space, while the leaves are the smallest possible partitions this kd-tree offers and each leaf explicitly records the data points that reside in the leaf. The partitioning continues until the number of data points in the hyper-rectangle fall below a certain threshold. The tree is built in a manner that adapts to the local density of input points and so the sizes of partitions at the same level are not necessarily equal to each other. For a given query point, the algorithm works by first descending the tree to find the data points lying in the cell that contains the query point. Then it examines surrounding cells if they overlap the ball centered at the query point and the closest data point that has been reached so far during traversal [20] [21].

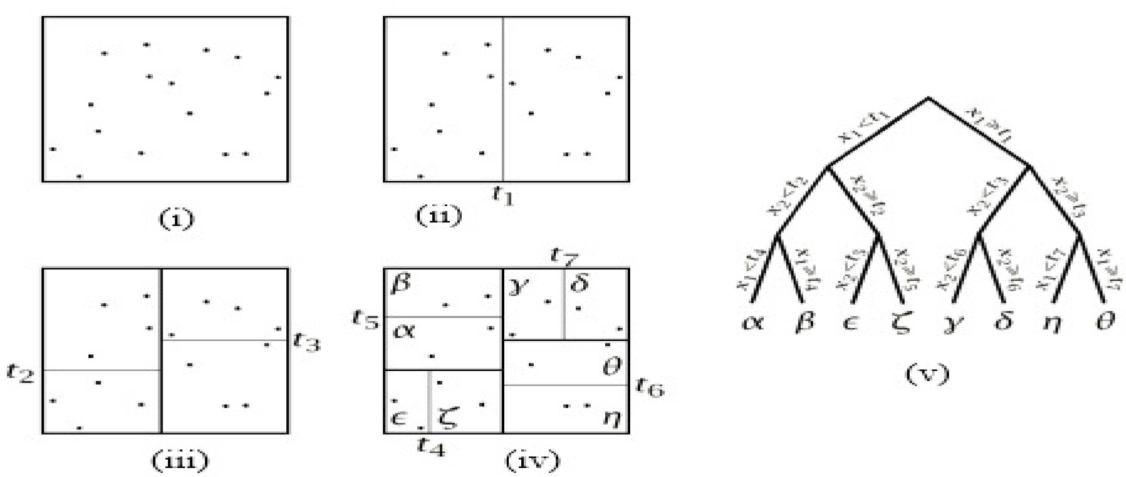

Figure 1.7 2-D kd-tree representations [20]

The Fig 1.7 gives two views of a 2 dimensions kd-tree. t_i represent the partition thresholds. x_i is the point density for a given partition. Sub-figures (i)-(iv) show the partitioning of the tree. The leaves represented by the Greek alphabetic letters are the smallest partitions for the given threshold that this tree can provide. The (iv) and (v) represent the same final tree in different ways.

1.5.4 Fast Calculation of Approximate Nearest Neighbor

Existing algorithms for the exact nearest neighbor problem have high complexities so, in practice, high dimensional nearest neighbor problems are usually solved with brute force search, that is, computing the distance of the query point with each data point in the database. To compensate for the computationally intensive recall of brute force search and achieve better performance a fast approximate version of k nearest neighbor based on a variation of kd-trees known as randomized kd-trees is used rather than a brute force search [22].

1.5.4.1 Randomized kd-Trees

A randomized kd-tree is built by choosing the split dimension randomly from the first D dimensions on which data has the greatest variance. Speed of search increases by building multiple randomized kd-trees and searching across them in parallel using a single priority queue of the values that needs to be searched. The creation of randomized kd-trees is such that they are mostly unique, thus avoiding any wasteful results caused due to similarity of tree structures. As linear search is too costly, an approximate nearest neighbor search is done, in which non-optimal neighbors are sometimes returned. Such approximate algorithms can be orders of magnitude faster than exact search, while still providing near optimal accuracy [22].

1.6 Applications of Content Based Image Retrieval

At a conceptual level the applications of CBIR may be categorized on the basis of user aims while searching or simply browsing an image collection and though not complete some of the general methods of image search are as follows [23]:

i. **Associative Search**

Users of search by association at the start have no specific aim other than find interesting things. Search by association often implies iterative refinement of the search, the similarity or the examples with which the search was started. Systems in this category typically are highly interactive, where the specification may by sketch or by example images.

ii. **Specific Search**

Another class of users aims the search at a specific image. The search may be for a precise copy of the image in mind, as in searching art catalogues. Target search may also be for another image of the same object of which the user has an image. Target search may also be applied when the user has a specific image in mind and the target is interactively specified as similar to a group of given examples.

iii. **Category Search**

The third class of applications, category search, aims at retrieving an arbitrary image representative of a specific class. It may be the case that the user has an example and the search is for other elements of the same class. Categories may be derived from labels or emerge from the database. In category search, the user may have available a group of images and the search is for additional images of the same class. A typical application of category search is catalogues of varieties.

1.6.1 Practical Applications of CBIR

The possible applications of CBIR are numerous, with some of the possible applications as follows [24]:

i. Crime prevention

Police can use CBIR to search images from surveillance cameras and other sources against images in criminal databases to identify and apprehend criminals.

ii. Military

It can be used to distinguish between friendly and enemy vehicles, personnel, automate identification of targets, guidance of missiles etc.

iii. Intellectual property

Reverse image search engines like TinEye are used to identify use of copyrighted images without permission, as well as misuse of trademarks and logos.

iv. Architectural and engineering design

Architects need to create designs and proof of concept images, drawings and visualizations to show clients, it is much easier if prior information can be searched for useful information.

v. Medical diagnosis

Automated image analysis of various visual data generating diagnostics like X-Ray scans to speed up the diagnosis and treatment of patients.

vi. Geographical information and remote sensing systems

CBIR systems can be used to automate the process of monitoring huge swathes of unpopulated area or area which doesn't have sudden changes. For example monitoring forest areas for fires or mountainous areas for flash floods.

vii. Cultural Heritage

Art Libraries and Museums contain visual material which can be stored in electronic form and made available to people who are physically far from the place. CBIR systems can give the people ability to search through the art collection for images of interest.

viii. Advertising

Image hosting websites and online advertising companies can automatically provide relevant advertisements on the basis of the content that is being hosted on the site or the content that has been uploaded by the user for more effective marketing.

ix. Web Searching

CBIR can be used to enable people to search for friends on social networking sites (Facebook), find images for a particular location, landmark (Flickr). It can also be used to filter inappropriate content using firewalls.

Chapter 2

Literature Review

The Multimedia Information Retrieval is a field that deals with the search of knowledge in all forms of information media like video, audio and image. Content based approach has been a fundamental approach to retrieve the required information correctly from the media databases with focus being on improving the search. The work to extract the digital information began as soon as the idea of digitizing content that was present in physical mediums such as books, vinyl records gained foothold. From a theoretical perspective, areas such as artificial intelligence, optimization theory, computational vision, and pattern recognition have contributed significantly to the underlying mathematical foundation of MIR. Psychology and related areas such as aesthetics and ergonomics provided basic foundations for the interaction with the user. Furthermore, applications of pictorial search into a database of images already existed in specialized forms such as face recognition for use in biometric applications, robotic guidance to traverse a terrain without striking any obstacles and taking the simplest route, and character recognition in textual data [2].

At the forefront however was the field of computer vision which provided some of the first algorithms for searching features in video, audio, and images. With the growth of the internet Web engines caught on, and started to provide image searches. Efforts were also made for integration of such systems directly into commercial database systems. It was realized by scientists during the course of developing media information systems that there was a widening semantic gap between the low level features like colors and textures that were used in computations by scientists and high level features like objects in an image that users generally searched for using words from their daily language when searching for images of interest. One of the earliest image based search engine to address the semantic gap was the ImageScape. In this system, the user could make direct queries for multiple visual objects such as sky, trees, water, and so on, using spatially positioned icons in a WWW index containing 10+ million images and videos using key frames. The system used information theory to determine the best features for minimizing uncertainty in the classification [2].

An image collection presents unique challenge because unlike a video, which though a successive presentation of images in a fast sequence to give the illusion of motion, usually has its subject matter change gradually from frame to frame and the object is relatively easier to find due to its dominant presence in consecutive frames of the video, images in a collection may differ widely in the scope of matter they cover [2]. For example a picture collection for a vacation may cover topics like parks, public landmarks, tourist attractions each pertaining to wildly different visual information.

Early attempts were mainly focused towards human face recognition but steadily the search diversified towards detecting objects in general. The difficulty of detecting objects from a general image led to research and advancements in proposal of novel and better similarity features like color features based on NF, RGB and m color space. Color can provide significant information about the content of an image [2]. Among the methods that use color as a retrieval feature, the most popular one is probably that of color histograms. The color histogram is a global statistical feature which describes the color distribution for a given image. Other low-level features widely used by researchers for indexing and retrieval of images, except color are texture and shape.

In order to exploit the strong aspects of each of these features while constructing an optimum and robust CBIR system, a plethora of methods, introduced over time, have been based on combinations of these features [2].Some recent work include using Multi-Channel based CBIR systems that work using multiple color channel representation of an image to find the relevant images [25] , using texture for image similarity and retrieval [26], using Scale Invariant Feature Transform (SIFT) with user feedback to determine the closest match [27]. Some new methods use not only the image content information but also the associated metadata like GPS data to segment images based on location for better image data segmentation [28].

Learning algorithms became an important part of image retrieval systems as it was realized that pattern recognition between underlying relationships of features would yield better results than was possible by simple matching of features. They allow the computer system to build a semantic understanding of the image collection and reduce the effects of noise introduced due to real world clutter in the image contents as well as ordering of images in vast collections. The earliest learning systems were based on neural networks, also used are components based searches, statistics based methods have also shown great promise. Some image retrieval systems also integrate human feedback in training the systems so there is a more human centered relevance to search results [2].

It is not only essential to create good image similarity measures but it is also important that the result of searches be available to the user in a reasonable period of time. One of the first approaches was to create image retrieval systems based on SQL databases. But they were found to have poor performances as some of the basic assumptions on data integral to their design don't hold in case of image data. Researchers then turned to similarity based databases using tree like structures to perform similarity matches [2]. Hybrid approaches that involve traditional Relational Database Management Systems (RDBMS) have also been proposed [29]. Even as the typical SQL database systems began to implement higher performance table searches, the search keys had to be exact such as in text search. Audio, images, and video were stored as blobs which could not be indexed effectively. Other data representations have also been suggested besides k-d trees. Vector quantization has shown as an effective method for searching large databases [2]. A 2-tier signature-based method for indexing large image databases has been proposed. Type 1 signatures represent the properties of the objects found in the images. Type 2 signatures capture the inter-object spatial positioning. Together these signatures allow the system to achieve a 98% performance improvement [2].The popular image search services provided by Google and Microsoft, through Google Search and Bing, respectively are prominent examples of large scale Web based proprietary CBIR implementations that use not only image content but also text and other metadata like hyperlinks, GPS, coordinates, time, user clicks etc. to provide accurate searches.

The search for better features continues to capture more relevant information as well as in terms of computation scalability in regards to application use in a server application or running on the mobile devices like Smart Phones.

The proposed system uses SURF algorithm which is immune to various changes in image content like scaling, rotation and other transformations for feature detection and descriptor creation because relative to another state of the art algorithm SIFT [30] it takes less computation time while providing the approximately same results[31][32]. Also, color and texture techniques used in many CBIR systems have to deal with the problem that the features produced describes the whole image and it has been shown that relative to global features local features are much better [33].

The randomized kd-tree algorithm was chosen for feature matching and retrieval because it scales better for high dimension vectors in comparison to linear search, fast calculation of best (approximately) matched features and efficient storage and retrieval of features, calculated for the images in image database which may number in hundreds of thousands. Also, it is has been found to be better than other approximate search algorithms like Approximate Nearest Neighbor (ANN) [22][34] and Local Sensitivity Hashing (LSH) [22][35].

Chapter 3
Problem Statement

3.1 Problem Definition

The advent of inexpensive electronic technology to capture, store and share image data whether it be photographs, documents, paintings or medical images has brought with challenges similar to those faced by textual information, namely those of storage and retrieval. It has One of these problems can be described as efficient retrieval of the user desired images from an image collection using the parameters chosen by the users. Traditional methods have involved using image annotations the images but a thorough annotation is impossible

The problem with image retrieval lies in finding a mix of features that can be calculated from one picture to another reliably as well as it being unique for the information it captures, i.e. two different objects cannot share the same kind of description.

A method of image retrieval based on existing technology is proposed and its formulation as well its relative efficacy and its strengths and weaknesses are discussed. The proposed image retrieval system is able to calculate and index a large number of image features and retrieve results based on a query image. The relative efficiency of the system is measured against annotated images, one can use a measure based on these, which in theory should produce the most meaningful possible automatic arrangement of the image set, clustering together the photographs depicting the same people, places, or objects. The dataset has been manually annotated and thus the measure to an extent will reflect the human perspective of the search.

3.2 Methodology

The steps taken to create an image retrieval system are given as below:

- Study of the basics of image retrieval systems
- Review of libraries and software enabling image retrieval technique implementation
- Determine a feature for image information differentiation
- Calculate the feature information for each image
- Store the feature information in optimal manner, for fast access and matching
- Retrieve the information

The image data set Multimedia Information Retrieval Flickr (MIRFlickr) [36] image dataset is used for the purposes of testing the image retrieval system and testing its relative efficacy via the annotation data present for the images present in the image dataset. The Open Source Computer Vision library (OpenCV) [37] was used for image processing, feature extraction and feature matching purposes.

Chapter 4

Image Retrieval using SURF Features

4.1 Image Features

A feature is a metric or some quantifiable value into which an image can be broken down, such that the feature represents the contents of the image at a high level. Images contain a lot of data like color information, texture, geometrical shapes, color blobs, corners etc. The first step is to detect such interest points in the image. The most valuable property of an interest point is detect or is its repeatability. The repeatability expresses the reliability to detect or find the same physical interest points under different viewing conditions. The second step is calculation of description of the interest points. The other property is that the feature should be unique i.e. if similar point is being described in two or more images then that point should have similar description so there is no ambiguity and incorrect matches are reduced.

The descriptors need to be of proper dimensions, as a large descriptor makes the computation take longer. But if the descriptor is small then it may discard some useful information.

4.2 Speeded Up Robust Features (SURF)

SURF [12] is a scale and rotation invariant detector and descriptor feature. Scale and rotation invariance mean that an object can be identified even though if the representation gets scaled in size or it is rotated about an axis in its representation in an image. Variance occurs due to the way information exists in reality and the incompleteness with which it can be captured from a recording. Invariance is an important property of image features, as measurement of similarity is possible only with respect to those features between two images which cannot be duplicated i.e. a feature describes is unique to the data point it describes [13].

The algorithm works in two stages

i. Interest point detection

SURF uses an interest point detector based on Hessian matrix. The descriptor vector is generated on the basis of points identified in this step. The steps to compute a detector are as follows [12][14]:

a. Input image is used to compute an integral image as it helps speed up the calculation of any rectangular area, especially the Haar wavelets. For a given point (x, y) in image Integral image I can be described as follows [15]:

$$I_{\sum}(x, y) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i, j) \tag{1}$$

b. The Hessian computation for a point X = (x, y) in the image at a scale of σ follows, where the L_{xx}, L_{xy} and L_{yy} are second-order Gaussian derivatives [15]:

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \tag{2}$$

c. In SURF, the second-order Gaussian derivates are approximated based on box filters and are denoted as D_{xx}, D_{xy} and D_{yy} instead. The Hessian determinant is computed from these terms as follows [12]:

$$\det(H_{approx}) = D_{xx}D_{yy} - 0.9D^{2}_{xy} \tag{3}$$

d. The Hessian matrix is computed for different filter sizes, where the filter size represents the region around which the matrix determinant is computed, with various scale factors. The process is repeated for various octaves. After calculating the Hessian matrix at different scale factors (different octaves and filter sizes), the interest points are chosen by computing the local maxima (in a 3x3x3 neighborhood) in scale and image space [16].This step is O (mn $\log_2$ (max (m, n))) for a m × n size image (m is the height of the image and n is the width in pixels).

e. Each extracted interest point is further improved by quadratic localization.

ii. Descriptor Generation

Descriptor vector computes the Haar wavelets, which is the coarse grain pixel contrast of a rectangular region, around a given interest point. It describes how pixel intensities are distributed within a scale dependent neighborhood of each interest point [15]. Typically descriptor vectors of length 64 or 128 are used.

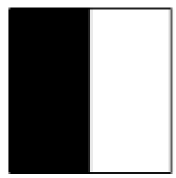

Figure 4.1 (a) Haar wavelet for gradient calculation in x-direction [12]

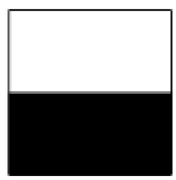

Figure 4.1 (b) Haar wavelet for gradient calculation in y-direction [12]

The next step calculates the descriptors for all detected key points which may or may not necessarily result in descriptors. The steps to compute a descriptor vector are [14]:

a. The computing descriptor vector determines a reproducible orientation that is identified based on a circular region surrounding the interest point. To assign the orientation, Haar wavelet responses (in x and y direction) are computed in a circular neighborhood of 6s (where s is the scale factor at which the interest point was identified).

b. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window (covering an angle of π/3). The longest vector (a sum of horizontal and vertical responses) is chosen as the dominant orientation.

c. After computing the dominant orientation, the descriptor vector is computed by choosing a square region of size 20s (where s is the scale factor at which the interest point was identified). The region chosen around the interest point and oriented along the dominant orientation is split up into smaller 4x4 square sub-regions and Haar wavelet responses (dx in the x and dy in the y direction) are calculated within each of the sub-regions at 5x5 regularly spaced sample points.

d. The wavelet responses are summed up in each region and the following four dimensional vectors per 4x4 region forms the descriptor vector [12]:

$$V = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \qquad (4)$$

It is a unique invariant description of the point detected which uniquely identifies the point and can be used to match the points even if target image suffers from distortion due to image noise, illumination, scale change, rotation change and change of the viewpoint from which the image was taken.

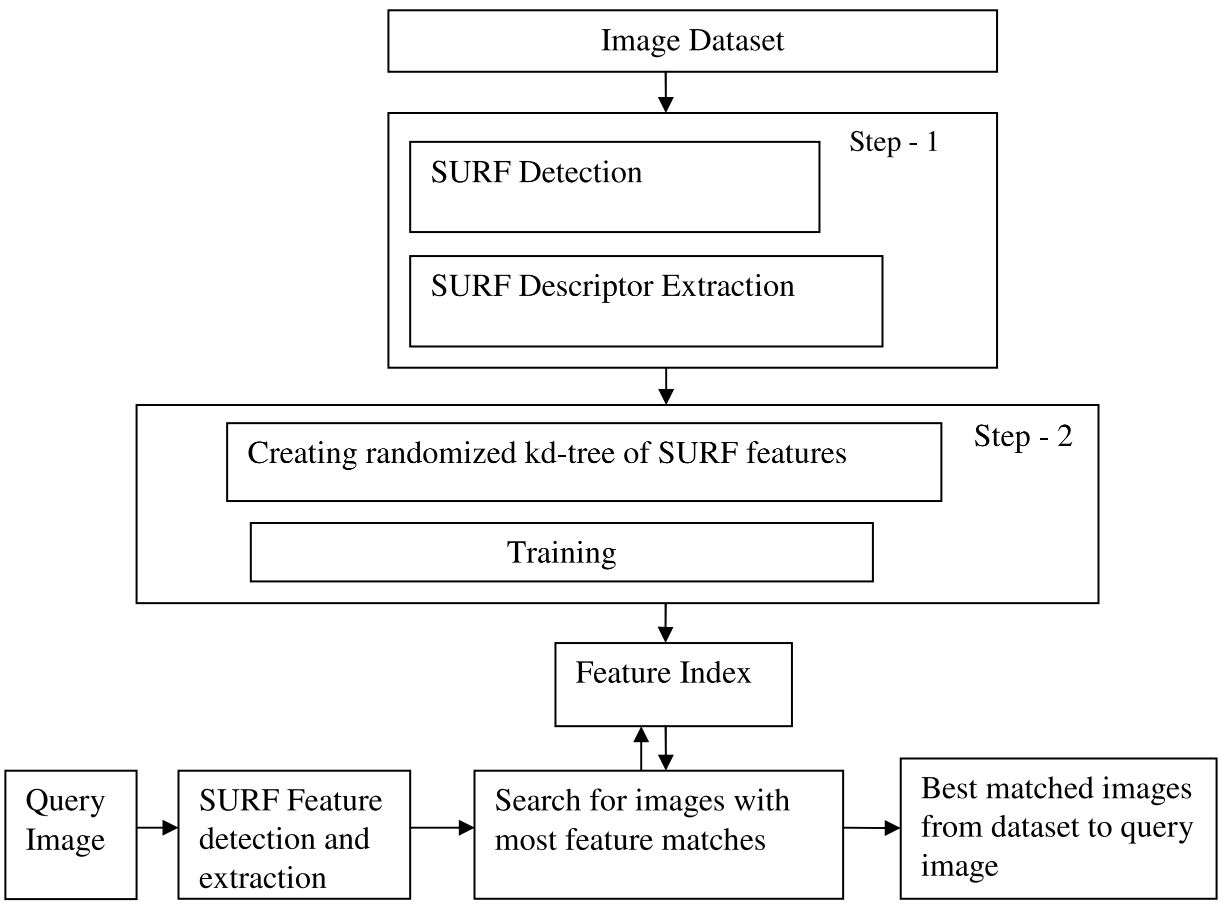

Figure 4.2 A block diagram of image retrieval system

All Image Retrieval Systems have some common features like image feature extraction, storage and retrieval. The image features determine the interesting feature points in an image and the subsequent matching mechanisms retrieves the best matched features as efficiently as possible.

The Fig 4.1 shows a logical breakdown of the design of the proposed image retrieval system. There are roughly three stages which are as follows:

i. The first stage is processing the images and extracting SURF descriptors. Steps followed are:

  a. Each image is read serially, as a grayscale picture a requirement of SURF algorithm.

  b. SURF interest point detector is applied on each image to detect points of interest in the image.

  c. SURF descriptor extraction is applied on the detected points which may or may not return a descriptor for the above calculated point.

ii. The second stage consists of storing the SURF descriptors, in such a manner that similar descriptors are stored together and are accessible in a fast manner.

  a. The kd-tree acts as part of storage and learning mechanism, as it stores the most similar vectors together, effectively creating clusters of similar vectors.

  b. The training stage is used to select the number of leaf nodes of the kd-tree that will be considered to achieve a specific precision.

iii. The third stage is taking the query image from the user extracting SURF descriptors from the image and finding its closest match.

  a. A fast approximate search algorithm based on randomized kd-trees is used. Step 2 creates an index of matched features for fast retrieval, here created using kd-tree data structure.

  b. The algorithm follows the best bin first strategy while calculating the best nearest matches for a preset number of neighbors. In this method most queries return nearest neighbor or very near neighbors.

c. The best matched image is calculated by adding the number of features that are matched to the query image and finding the image with maximal matches.

The functionality has been implemented in the program that processes an image set of thousand images each at a time. The following screenshot shows the arguments to the program:

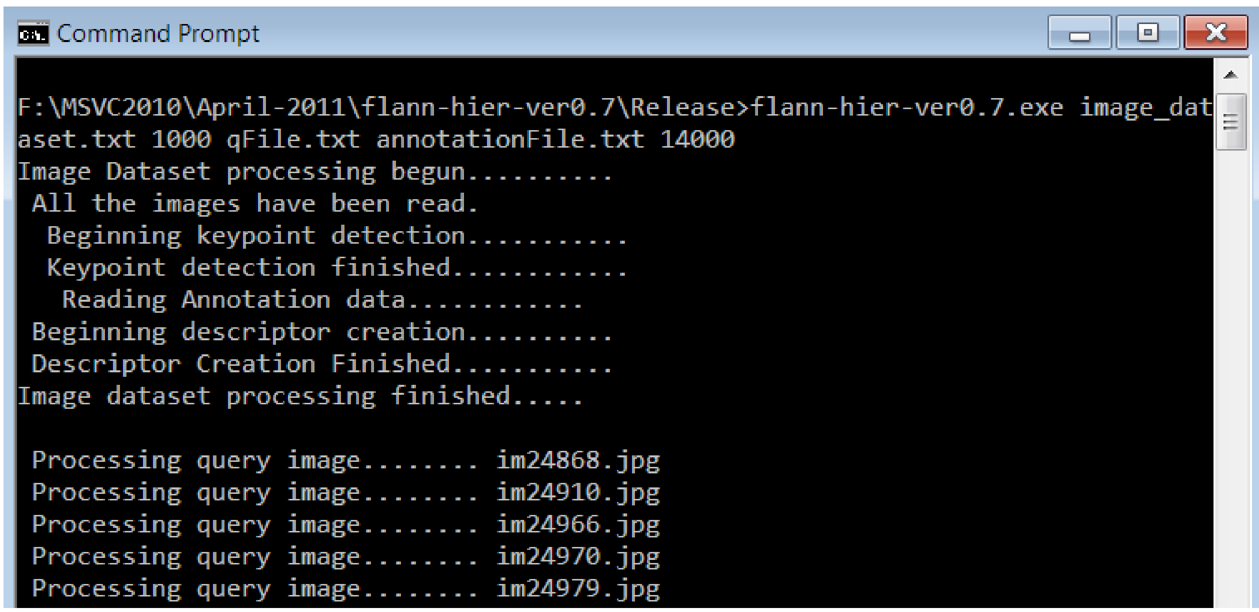

Figure 4.3 Program Screenshot

The arguments taken by the program are as follows:

i. The first argument is the name of a file that contains the path to the images of the image dataset, the image dataset is the set of images from which matches are to be discovered.

ii. The second argument is for the number of images that are to be read from the dataset.

iii. The third argument is the name of the file that holds the path to query images, images to be retrieved need to be similar to these images.

iv. The fourth argument is the file that contains the paths to the annotations containing files.

v. The fifth argument is the number before the numeral part of the image of the first image of the set. It is used when processing the annotations of a image set.

The various steps followed are:

i. The program reads all the dataset images to memory.

ii. It then detects key points within the images which are subsequently used for creating descriptors of length 64 and 128. The descriptors are used to compute the match for each query image based on fast approximate algorithm where k has been preset to 2- 5.

iii. The information about the image from which a feature is derived is stored in the feature itself, the number of matches between query and dataset image is calculated using this information.

iv. The correct prediction rate is calculated by considering top hundred image matches for a query image on the basis of feature matching. The annotations corresponding to these images are calculated from annotations of query image and then they are divided by the total annotations of a category to which the query image belongs. The resultant fractions are summed and averaged over the number of query images to calculate the percentage of correct matches.

v. All the data generated is saved in a file.

The program makes certain assumptions about data files it requires:

i. The images need to be in a folder named images and image names are of the format “im*.#” .The ‘*’ represents a unique decimal numeral to number the image and ‘#’ represents any for three character image extension supported by the OpenCV library. Example im6147.png, im23.bmp, im343.jpg.

ii. The annotation file begins with a number that denotes the number of lines of subsequent data to be read. Each subsequent line is a path to an annotation file.

Each annotation file contains only numeric part of image name, numbers are in ascending order, one per line.

A visual representation of the features detected by the SURF algorithm as useful markers of image content description is shown in the example below. An image of a car is taken as a query image as shown in Fig. 4.3, and three images from the data set are taken. The image in dataset Fig. 4.4 has cars almost in the background, but the Fig 4.5 has cars in the foreground. The dataset image Fig. 4.6 entirely is dominated by the presence of cars.

Figure 4.3 Image 1

Figure 4.4 Image 2

Figure 4.5 Image 3

Figure 4.6 Image 4

The subsequent three images in grayscale (SURF only works on grayscale images) show the correspondence, the algorithm has established between the interest points of the images. The left half consists of a query image and the right half consists of a training image.

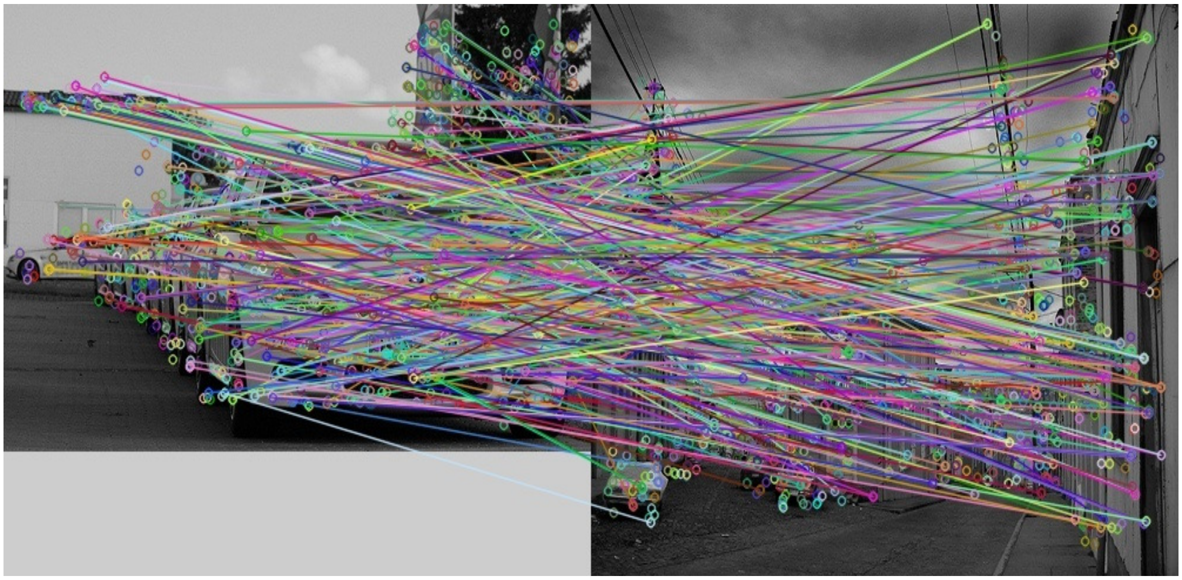

Figure 4.7 Feature mapping from Image 1 to Image 2

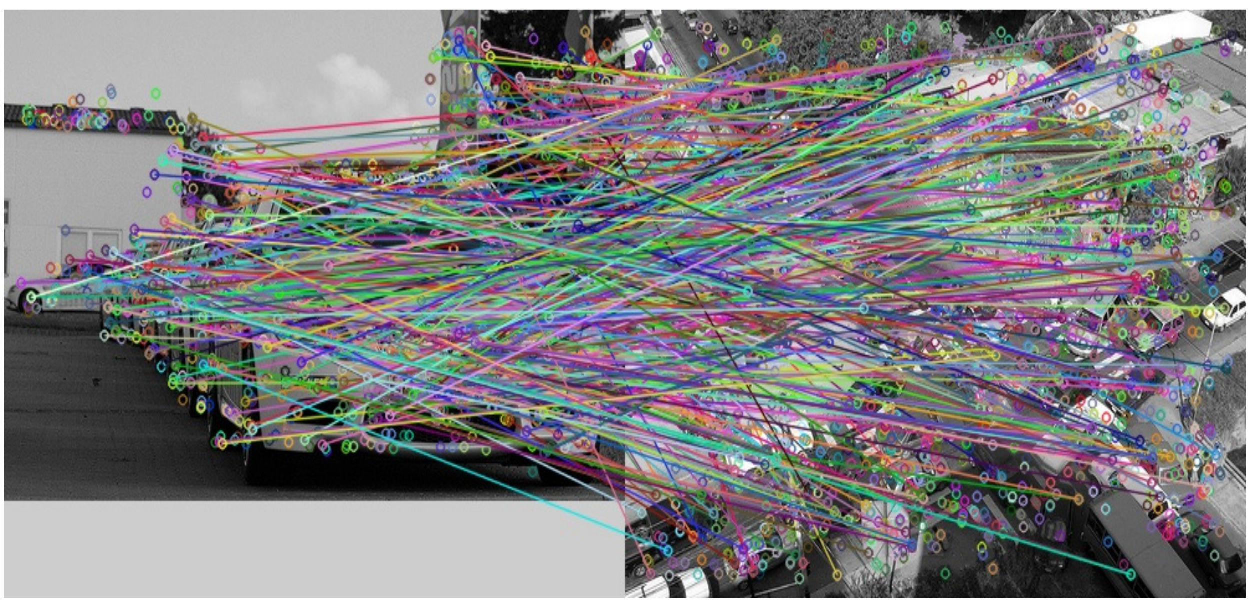

Figure 4.8 Feature mapping from Image 1 to Image 3

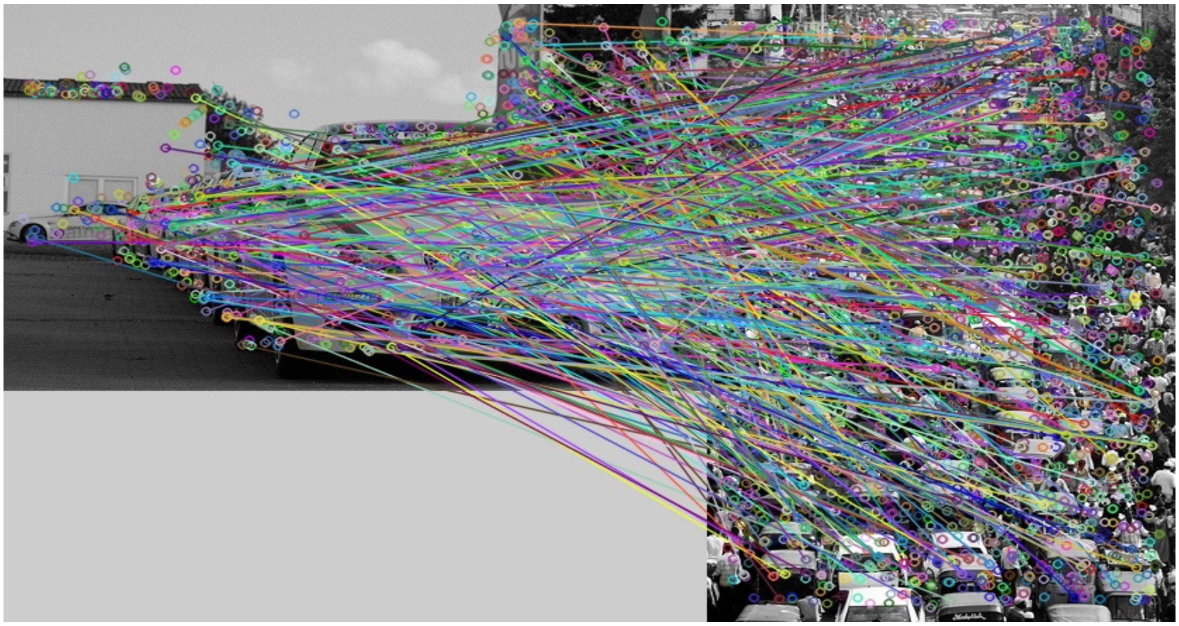

Figure 4.9 Feature mapping from Image 1 to Image 3

Chapter 5

Testing and Results

The program was tested on images from the MIRFlickr image dataset. In absence of any standardized testing for the measuring the performance of an image retrieval system and the fact that CBIR systems often need to be tuned [2] for the dataset that they are being designed, a testing method was devised based on the annotations available with the MIRFlickr Image dataset. The method based on calculating relative efficiency of search against manual annotations gives an approximate performance of the image retrieval system.

Steps followed in testing are:

i. Image sets of thousand images per set were prepared from MIRFlickr Image set. The number of images chosen so as to strike a balance between comprehensive results and the computation time taken on the test system.

ii. Files containing paths of the images in the dataset was prepared for each set of images.

iii. A file containing the paths to files containing annotation data was prepared. All the available thirty eight annotation files were used.

iv. A test set of images was constructed from the MIRFlickr Image set. The images of the test were not part of the image sets constructed in step (1). The test set images were taken randomly from the image dataset.

v. A file with path of query images was prepared.

vi. Two sets of SURF features are calculated of lengths 64 and 128 respectively to test for any variances in accuracy that may occur due to the fact that a 64 element vector stores less information than stored by a 128 element vector.

vii. The distance between neighbors is calculated using the Euclidean algorithm for distance calculation

viii. The program uses the preset defaults values for the various algorithms used.

ix. Each test set was processed separately and the results generated were noted.

5.1 Testing Platform

The test computer system had a 1.5 GHz processor and 2GB RAM, the program was developed in Microsoft Visual Studio 2010 and the open source library used was OpenCV ver. 2.2.

5.2 Result Tables

No.	k = 2	k = 3	k = 4	k = 5
1	10.72%	11.14%	11.05%	11.03%
2	10.77%	10.93%	10.97%	11.08%
3	11.39%	11.65%	11.56%	11.64%
4	11.34%	11.60%	11.74%	11.98%
5	11.13%	11.11%	11.12%	11.13%
6	10.73%	10.86%	10.84%	10.95%
7	11.18%	11.16%	11.24%	11.2%
8	10.59%	10.51%	10.79%	10.80%
9	11.02%	10.74%	10.84%	10.86%
10	10.96%	10.95%	10.75%	10.80%
11	10.88%	10.84%	10.85%	10.95%
12	11.53%	11.64%	11.42%	11.46%
13	10.75%	10.89%	11.10%	11.20%
14	10.96%	11.22%	11.12%	11.15%
15	11.35%	11.29%	11.11%	11.05%

Table 5.1 SURF - 64 results

No.	k = 2	k = 3	k = 4	k = 5
1	11.08%	10.87%	10.86%	10.68%
2	11.20%	11.24%	11.32%	11.29%
3	11.22%	11.43%	11.39%	11.70%
4	11.76%	11.86%	11.82%	11.53%
5	11.43%	11.40%	11.48%	11.10%
6	11.05%	11.19%	11.36%	11.29%
7	11.52%	11.56%	11.44%	11.61%
8	10.79%	10.83%	10.67%	10.64%
9	10.94%	10.89%	10.81%	10.80%
10	10.98%	11.09%	10.82%	10.84%
11	10.91%	11.17%	10.88%	10.70%
12	11.30%	11.50%	11.35%	11.61%
13	11.30%	11.48%	11.79%	11.77%
14	11.17%	11.50%	10.98%	11.02%
15	11.36%	11.39%	11.27%	11.30%

Table 5.2 SURF - 128 results

5.3 Interpretation of Results

The following observations can be made from looking at results in Table 5.1 and Table 5.2:

i. The results show that according to the annotation data for the image dataset the accuracy of the image search lies between 10% to 12%.

ii. The results show that increasing value of k Nearest Neighbors, increases albeit unevenly and almost negligibly the accuracy of the search.

iii. The increase in SURF descriptor length from 64 to 128 elements, in general increases the accuracy, but overall the increase in accuracy is negligible.

The low success rate can be due to the following reasons:

i. The images in the dataset are almost randomly distributed, therefore, the learning algorithm has problems in accurate classification of images from disparate data.

ii. The image retrieval system is accurate, but is matching features that may not be readily apparent to the naked eye.

iii. The limited amount of annotations does not cover all the matches made by the image retrieval system.

iv. Due to the interface provided to the algorithm implementation in OpenCV the various presets were used as they were, tweaking the presets for the dataset may have provided better results.

v. It was observed that pictures with large amount of geometry and texturing in image content were biasing the results in their favor. For example in Fig. 5.4 and Fig.5.5 two separate image of the Big Ben (Fig. 5.1 and Fig. 5.2) tower have less corresponding matches than the knitting design in Fig.5.3.

Figure 5.1 Image 5

Figure 5.2 Image 6

Figure 5.3 Image 7

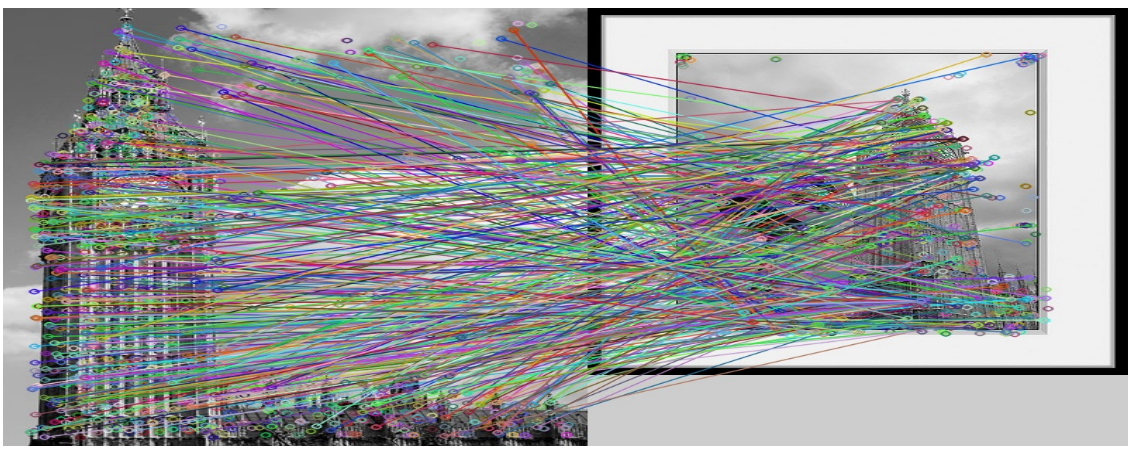

Figure 5.4 Feature matching from Image 5 to Image 6

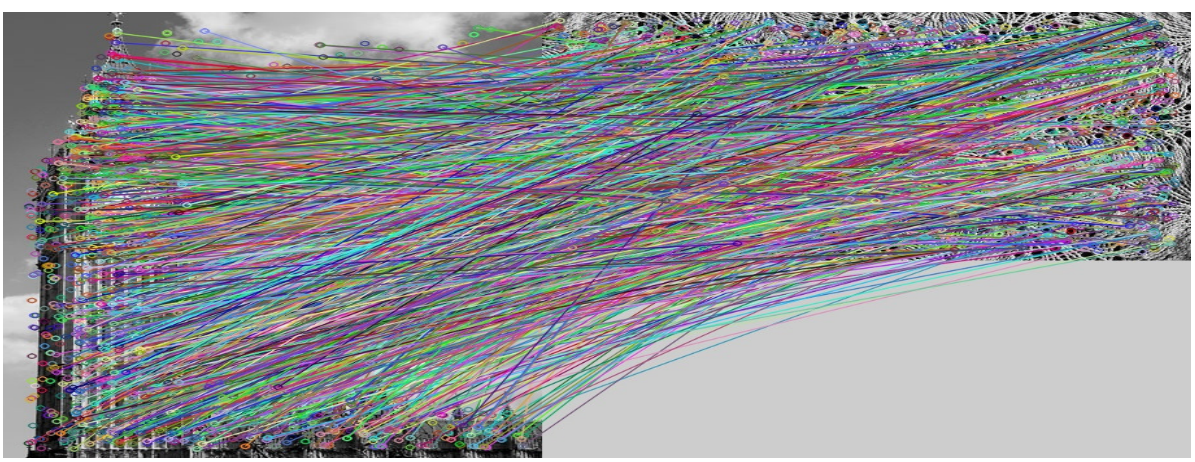

Figure 5.5 Feature matching from Image 5 to Image 7

5.4 Image Query and Results

The following are the query and search results produced by the program. Fig. 5.6 (a), Fig. 5.7 (a) and Fig. 5.8 (a) are query images and Fig. 5.6 (b), Fig. 5.7 (b), Fig. 5.8 (b) are their respective top ten results.

Figure 5.6 (a) Query image

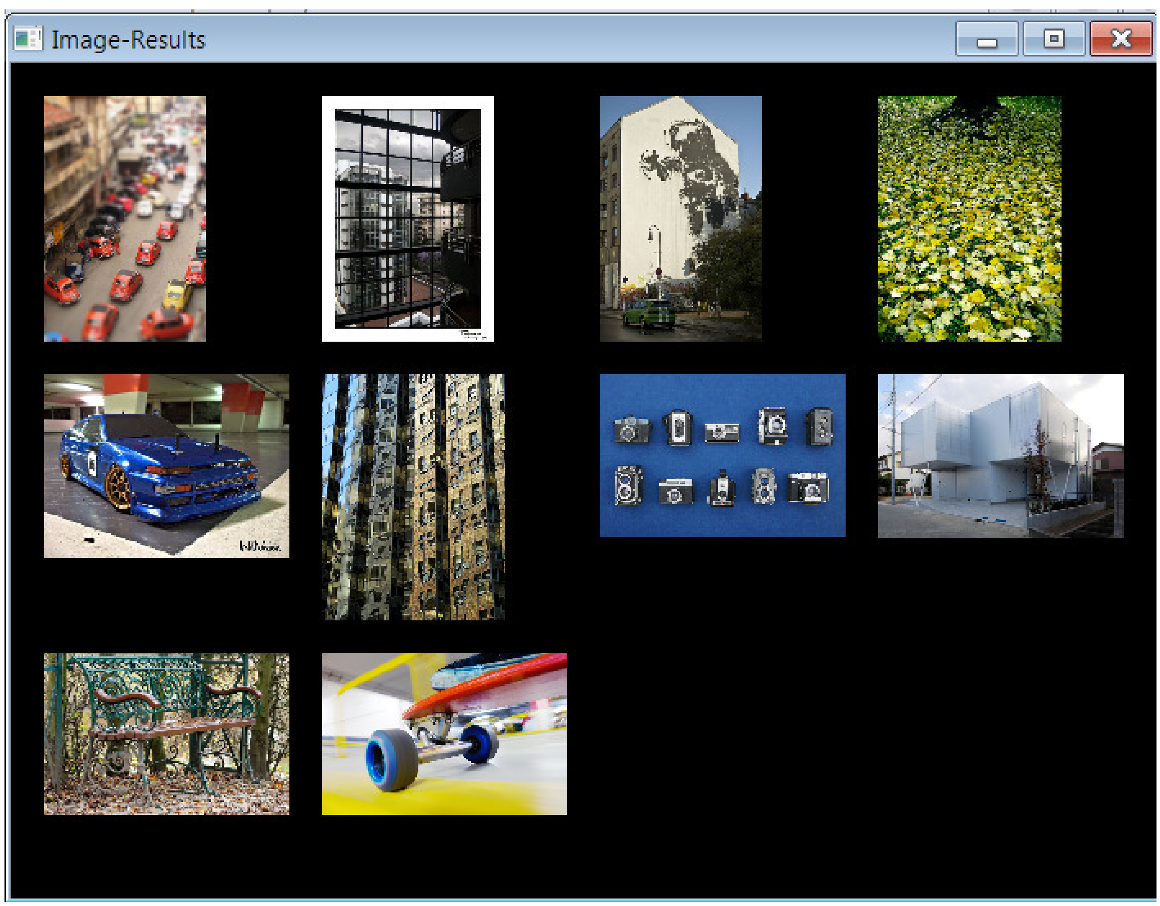

Figure 5.6 (b) Result

Figure 5.7 (a) Query image

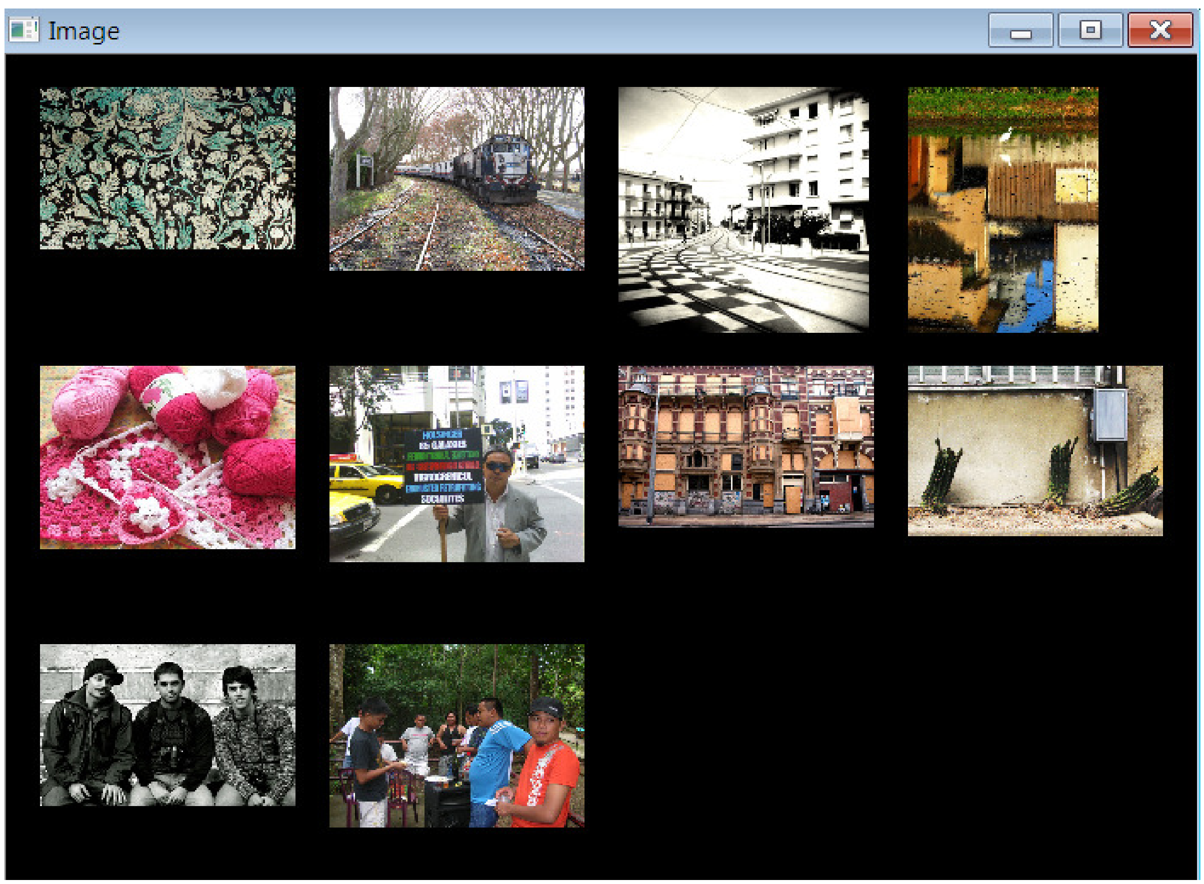

Figure 5.7 (b) Result

Figure 5.8 (a) Query image

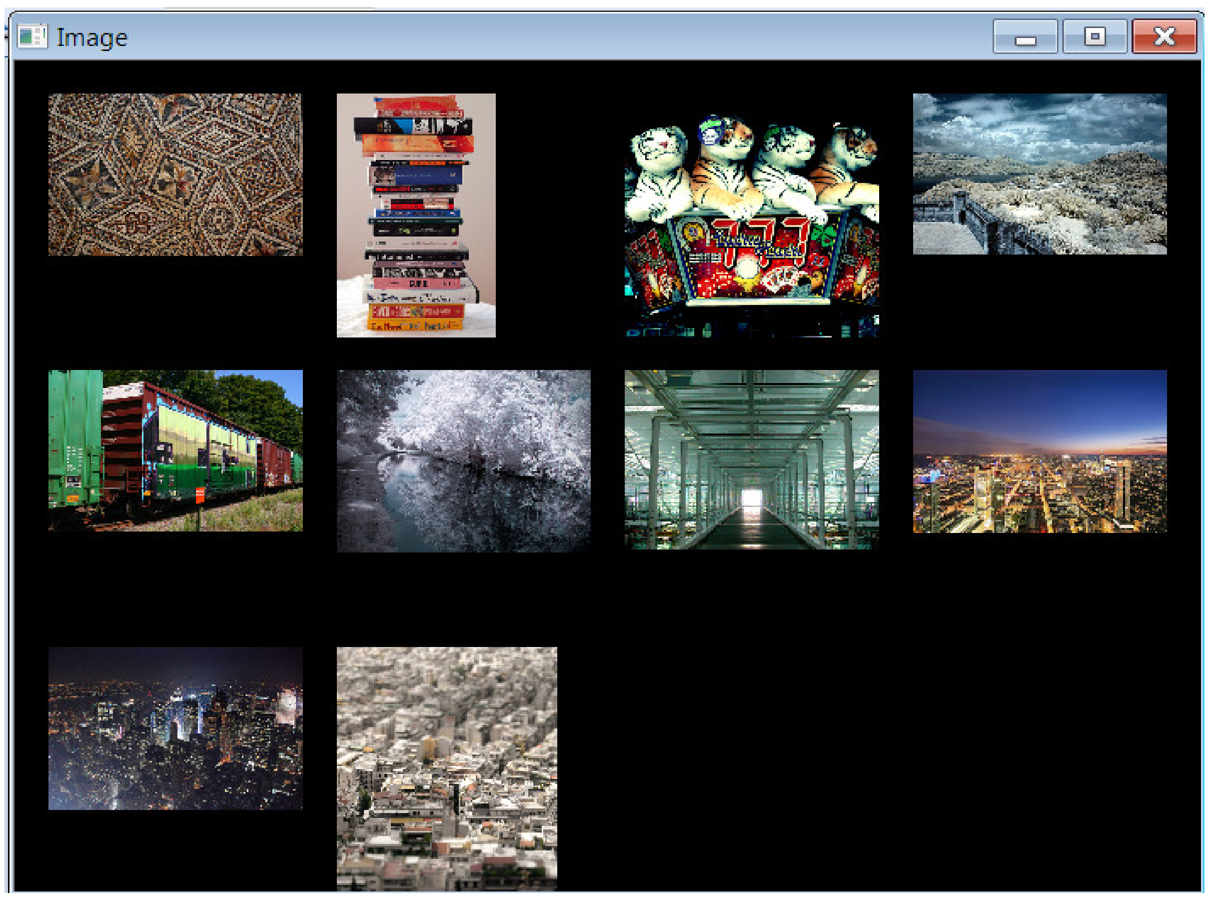

Figure 5.8 (b) Result

Chapter 6

Conclusion

6.1 Conclusion

The primary aim of the thesis to implement an image retrieval system was successfully accomplished and its performance was also, measured and analyzed. After analyzing the results it can be concluded that image retrieval works albeit with low apparent success rate and that this rate is reflected for the top hundred images retrieved by the system for a query image. Also, the SURF descriptor of length 64 is enough for feature description and matching purposes as the SURF descriptor of length 128 does not provide any appreciable gain relative to the computation time and the increased memory requirements of a larger descriptor. It can also be concluded that increasing the number of nearest neighbors does not appreciably increase the positive matches of the image features and that the Randomized kd-Tree algorithm is biased for images with larger geometry and texture content.

6.2 Future Scope

The thesis work may be improved upon in the following manner:

- **i.** Use other distance measuring algorithms like Minkowski distance, Manhattan distance or Mahalanobis distance for measuring the nearest neighbor. Other distance measures may help in better discrimination among the distances of image features of the same category and those from another category.
- **ii.** Usage of other features like color, texture and metadata if available to complement the existing approach and get better results.

References

[1] Carlos A. Zambrano, Shimla Panorama, Available: http://www.flickr.com/photos/arturnuta/5015141852/

[2] Michael S. Lew, Nicu Sebe, Chabane Djeraba, Ramesh Jain, "Content-Based Multimedia Information Retrieval: State of the Art and Challenges, ACM Transactions on Multimedia Computing, Communications, and Applications," vol. 2, issue 1, 2006, pp. 1-19.

[3] D.H. Ballard and C. M. Brown, 1982, "Computer Vision", Prentice Hall, New Jersey, USA.

[4] Paul Bloore, TinEye, 2008. Available: http://www.tineye.com

[5] Peters, Richard Alan, II, "Introduction and Overview", Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008, Available: http://www.archive.org/details/Lectures_on_Image_Processing.

[6] Johannes Kopf, Dani Lischinski, Depixelizing Pixel Art, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011), 30(4), Article no. 99.

[7] PNG, Online Information Resource, Available: http://www.w3.org/Graphics/PNG/

[8] JPEG, Online Information Resource, Available: http://www.jpeg.org/

[9] SVG, Online Information Resource, Available: ,http://www.w3.org/Graphics/SVG/

[10] Alasdair McAndrew, An Introduction to Digital Image Processing with Matlab Notes, Victoria University of Technology, Available: http://appnote.avrportal.com/download/appnotes/Image-processing/An-Introduction-to-Digital-Image-Processing-with-Matlab-Notes-for-SCM2511-Image-Processing.pdf

[11] Bertalmio, M.; Vese, L.; Sapiro, G.; Osher, S, "Simultaneous structure and texture image inpainting," Image Processing, IEEE Transactions on, Volume: 12 Issue: 8, Aug. 2003. Page(s): 882 -889.

[12] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded up robust features,” in Proc. of the 9th European Conference on Computer Vision (ECCV’06), ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3951. Graz, Austria: Springer-Verlag, May 2006, pp. 404–417.

[13] Kan Deng, “OMEGA: On-Line Memory-Based General Purpose System Classifier,” PhD thesis, Robotics Institute, Carnegie Mellon University, 1998. Technical Report CMU-RI-TR-98-33.

[14] S. Srinivasan et al., “Performance characterization and optimization of mobile augmented reality on handheld platforms,” in IEEE Inter-national Symposium on Workload Characterization, Oct 2009.

[15] Chris Evans, Notes on the OpenSURF Library, Available: http://sites.google.com/site/chrisevansdev/files/opensurf.pdf

[16] Chen, W.-C., Xiong, Y., Gao, J., Gelfand, N., Grzeszczuk, R., Efficient extraction of robust image features on mobile devices, In Proc. ISMAR 2007, 2007.

[17] Nils. J. Nilsson, Introduction to Machine Learning, Draft Notes, Stanford University, Available: http://ai.stanford.edu/~nilsson/mlbook.html

[18] T. G. Dietterich, Machine Learning, In Nature Encyclopedia of Cognitive Science, London: Macmillan, 2003.

[19] Martin Sewell, Machine Learning, Department of Computer Science University College London April 2007.

[20] Ricardo Gutierrez-Osuna, Nearest Neighbor, Lecture notes, Texas A&M University, Available: http://research.cs.tamu.edu/prism/lectures/pr/pr_l8.pdf

[21] J. L. Bentley, “Multidimensional binary search trees used for associative searching.” Communications of the ACM (CACM), Vol. 18(9):509–517, 1975.

[22] Marius Muja, David G. Lowe, “Fast Approximate Nearest Neighbors With Automatic Algorithm Configuration”. In VISAPP International Conference on Computer Vision Theory and Applications 2009, pp 331-340.

[23] Arnold W.M. Smeulders, Senior Member, IEEE, Marcel Worring, Simone Santini, Member, IEEE, Amarnath Gupta, Member, IEEE, and Ramesh Jain, Fellow, IEEE , “Content-Based Image Retrieval at the End of the Early Years”

[24] J. Eakins, M. Graham, Content-based image retrieval, Technical Report, University of Northumbria at Newcastle, 1999.

[25] James C. French, James V. S. Watson and Xiangyu Jin and W. N. Martin, "Integrating Multiple Multi-Channel CBIR Systems (Extended Abstract)," Proc. Inter. Workshop on Multimedia Information Systems,pp.85—95, 2003.

[26] Selim Aksoy, Professor Robert, M. Haralick, "Textural features for content-based image database retrieval," In Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries, in conjunction with CVPR'98, 1998.

[27] X. Wangming, W. Jin, L. Xinhai, Z. Lei, and S. Gang. "Application of image SIFT features to the context of CBIR." In CSSE '08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, pages 552-555, Washington, DC, USA, 2008. IEEE Computer Society.

[28] Yan-Tao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua and Hartmut Neven, "Tour the World: Building A Web-Scale Landmark Recognition Engine." NUS Grad. Sch. for Integrative Sci. & Eng., Nat. Univ. of Singapore, Singapore, Singapore.

[29] Seung-Hoon Lee, Gi-Hwa Jang, Su-Hyun Lee, Sung-Hwan Jung; Yong-Tae Woo , "A content-based image retrieval system using extended SQL in RDBMS ," Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on , vol., no., pp.1069-1072 vol.2, 9-12 Sep 1997.

[30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[31] Bauer, J., S¨underhauf, N., & Protzel, P. (2007). "Comparing Several Implementations of Two Recently Published Feature Detectors." In Proc. of the International Conference on Intelligent and Autonomous Systems, IAV, Toulouse, France. Kan Deng.

[32] Christoffer Valgren and Achim Lilienthal, "SIFT, SURF and Seasons: Long-term Outdoor Localization Using Local Features." Proc. of 3rd European Conference on Mobile Robots,2007.

[33] Mohamed Aly, Peter Welinder, Mario Munich, and Pietro Perona, “Automatic Discovery of Image Families: Global Vs. Local Features.” IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, November 2009.

[34] David M. Mount and Sunil Arya, A Library for Approximate Nearest Neighbor Searching (ANN), Available: http://www.cs.umd.edu/~mount/ANN/

[35] Alex Andoni, Locality-Sensitive Hashing (LSH), Available: http://www.mit.edu/~andoni/LSH/

[36] M. J. Huiskes, M. S. Lew (2008). The MIR Flickr Retrieval Evaluation. ACM International Conference on Multimedia Information Retrieval (MIR'08), Vancouver, Canada.

[37] Bradski. G, The OpenCV Library, Dr. Dobb's Journal of Software Tools, Available: http://opencv.willowgarage.com/wiki/Welcome

List of Publications

Accepted

[1] Ashish Kumar, Shalini Batra, "Image Retrieval using SURF features and Annotated Data", accepted in International Journal of Advanced Research in Computer Science.