Efficient Image Retrieval Methods For Large Scale Dynamic Image Databases

Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science (by Research)

in

Computer Science

by

Suman Karthik

200407013

sumankarthik@research.iiit.ac.in

International Institute of Information Technology

Hyderabad, India

May 2009

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Efficient Image Retrieval Methods For Large Scale Dynamic Image Databases" by Mr. Suman Karthik, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Dr. C. V. Jawahar

To my Family

And

Friends

Acknowledgements

I would like to thank Dr. C. V. Jawahar for his role as a guide and mentor to me in the past five years. I was fortunate enough to have him as my advisor. He has been a great influence on me in academic and more importantly in non academic pursuits. His clarity of thought and work ethic have been instrumental in inspiring me in my academic endeavors.

I would also like to thank Dr. P. J. Narayanan, for his support in both my academic and non academic endeavors and Dr. A. Namboodiri and Dr. J Sivaswamy for various references and guidance in different subjects related to the stream.

I would like to thank my dear friend Balu without whom I wouldn't be at IIIT Hyderabad. Its been my pleasure to actively collaborate and work with both pradhee tandon and chandrika pulla. The best thing about my time at IIIT hyderabad has been CVIT and CVITians, i would like to acknowledge some of them and apologize to all those i have missed. People from the old CVIT Sesh, MNSSK, Vakiba and Somu have been great influences on me. Other contemporaries have been Visesh, Pramod, Ranjith, Booty, Vardhaman, Paresh, Tarun, Jagmohan, Jyotirmoy, Avinash and juniors like Skp, Shiben, pramodp, Suman, praveen, sanjeev, Ranta, pooja, rakesh, sreekanth, chetan, Narsimharaju, naveen and rasagna. Being surrounded by so many intelligent and creative lab mates, always ready to talk, argue, and shoot down ideas, was certainly the best thing about my entire graduate degree. I would also like to express my gratitude to other IIITians like Rishi, Srirarm and Prashant.

Finally, I would like to appreciate the patience and support from my Family and friends. I owe deep gratitude to all my friends and family members. Without their blessings and support, throughout, this thesis would not be a reality.

Abstract

The commoditization of imaging hardware has led to an exponential growth in image and video data, making it difficult to access relevant data when it is required. This has led to a great amount of research into multimedia retrieval and Content Based Image Retrieval (CBIR) in particular. Yet, CBIR has not found widespread acceptance in real world systems. One of the primary reasons for this is the inability of traditional CBIR systems to scale effectively to *Large Scale* image databases. The introduction of the Bag of Words model for image retrieval has changed some of these issues for the better, yet bottlenecks remain and their utility is limited when it comes to *Highly Dynamic* image databases (image databases where the set of images is constantly changing). In this thesis, we focus on developing methods that address the scalability issues of traditional CBIR systems and adaptability issues of *Bag of Words* based image retrieval systems.

Traditional CBIR systems find relevant images by finding nearest neighbors in a high dimensional feature space. This is computationally expensive, and does not scale as the number of images in the database grow. We address this problem by posing the image retrieval problem as a text retrieval task. We do this by transforming the images into text documents called the Virtual Textual Description (VTD). Once this transformation is done, we further enhance the performance of the system by incorporating a novel relevance feedback algorithm called discriminative relevance feedback. Then we use the virtual textual description of images to index and retrieve images efficiently using a data structure called the Elastic Bucket Trie(EBT).

Contemporary bag of visual words approaches to image retrieval perform one-time offline

vector quantization to create the visual vocabulary. However, these methods do not adapt well to dynamic image databases whose nature constantly changes as new data is added. In this thesis, we design, present and examine with experiments a novel method for incremental vector quantization(IVQ) to be used in image and video retrieval systems with dynamic databases.

Semantic indexing has been invaluable in improving the performance of bag of words based image retrieval systems. However, contemporary approaches to semantic indexing for bag of words image retrieval do not adapt well to dynamic image databases. We introduce and examine with experiments a bipartite graph model (BGM), which is a scalable datastructure that aids in on-line semantic indexing and a cash flow algorithm that works on the BGM to retrieve semantically relevant images from the database. We also demonstrate how traditional text search engines can be used to build scalable image retrieval systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of digital cameras and cheap commodity hardware, has paved the way for an exponential growth in the amount of image and video content generated. This growth, coupled with the rapid distribution and dissemination capabilities of world wide web means large scale dynamic image/video databases. Some prominent examples of this growth are sites like Flickr and Youtube. These sites are essentially very large repositories of community generated image and video content that are growing at an ever increasing pace. As was discovered for textual data before it, information retrieval is a viable way of dealing with such information overload. In case of images the information retrieval paradigm used is content based image retrieval. Such retrieval based on content is significantly challenging when it comes to visual data. Even text retrieval systems still rely on keywords and their context to retrieve and do not actually understand the content. This problem is aggravated when it comes to retrieval of visual content which by its very nature is subjective and hence is not limited to interpretation as text content is. For example in Figure 1.1 one can see how the content of two very simple images can have many different interpretations. Some of the interpretations of Figure 1.1 like dogs, wild dogs, predators, teens, boys and girls, are classes which can be discerned from the visual content of the image. Other interpretations like Africa, wild, friends, happy are a result of apriori human knowledge applied to the visual content. We can only hope to model and achieve the former, which by itself is a very

(a) 

(b) 

Figure 1.1: Two images taken from flickr with their tags (a)dogs, wild dogs, Africa, predators, wild. (b)teens, boys and girls, friends, students, happy, huddle

challenging task.

It is important to appreciate the distinction between current methods of wide spread image retrieval *(circa. April 2009)* in the real world which use textual cues to retrieve images and content based image retrieval which uses the image content to retrieve images. For example, querying google's image search with the term *sr71* retrieves images as shown in Figure 1.2, one can clearly notice the bold text below the retrieved image that highlights the text *SR-71* and the context of its occurrence in the relevant webpage. Even the recently launched google similar images(GSI) which take visual similarity into consideration still relies predominantly on textual cues. Figure 1.3 shows retrieval results for GSI using the top image given as an exemplar, the relevant images contain both *black birds* and the *blackbird* which is an informal name of the *SR71*.

1.1 Problem

Figure 1.4 shows a rudimentary content based image retrieval system. Here images are first gathered from different data sources and stored in a database which forms the content storage of the CBIR system. Then feature extraction is done from all the images using features like color, shape, texture etc. These features are designed to mimic low level human

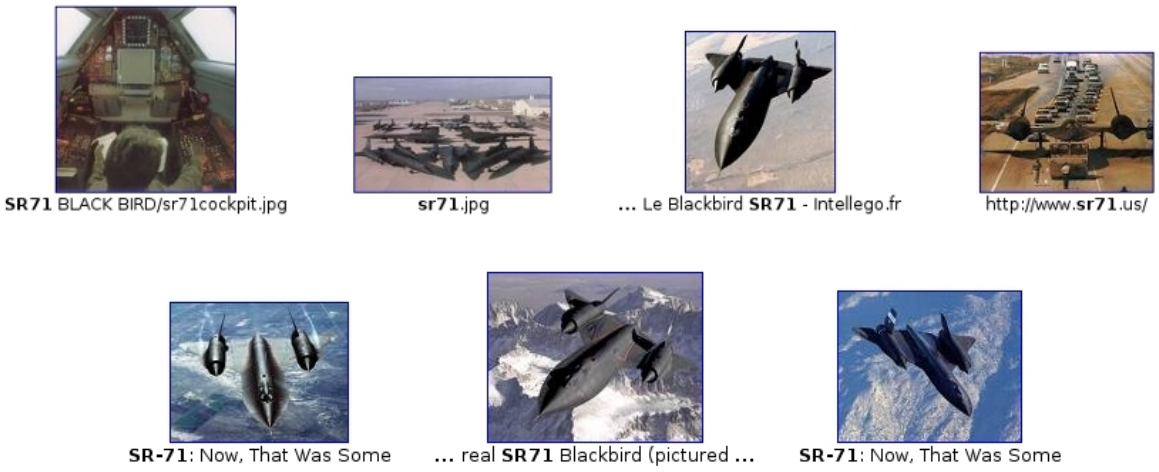

Figure 1.2: Is the retrieval result for the query SR71 using google image search. The text SR71 used to retrieve the relevant images is highlighted below each of the retrieved images, showing how text cues are used to retrieve relevant images.

vision, this helps the computer to *associate* images at the lowest level as humans do. Once features are extracted they are represented by a feature vector, which is a point within the high dimensional feature space. Finally when the user gives the CBIR system a query image, the given query is first interpreted using feature extraction and indexing, then the relevant images are retrieved using a nearest neighbors query in the index. The K nearest neighbors methods choose the K nearest feature vectors to the given feature vector within the feature space. Once the nearest neighbors are retrieved the corresponding images are presented to the user.

The two critical bottlenecks for performance and quality in a CBIR system are the semantic gap and scalability. The semantic gap is the difference between machine and human perception of visual data. The semantic gap can be tackled by coming up with better feature vectors that mimic high level human vision, building machine learning modules that model human visual perception better, etc. The scalability and adaptability of a CBIR system is effected by the indexing and searching modules of the system. The naive CBIR system that

Figure 1.3: Is the retrieval result when the top image is given as exemplar to google similar images(GSI). The retrieval results clearly show the predominance of textual cues over visual cues, as *black birds* and the airplane *blackbird* are retrieved for the given query

retrieves images based on nearest neighbor search in a high dimensional feature space is inherently not scalable. The K nearest neighbors approach is very computationally expensive due to its high complexity. This algorithm, works for small databases but quickly becomes intractable as either the size or the dimensionality of the feature space becomes large. If S is the feature space, nearest neighbor search has a complexity of O(Nd) where N is the number of images in the database and d is the number of dimensions in S. When millions of images are involved in a high dimensional feature space of hundreds or thousands of dimensions traditional CBIR systems become impractical.

Content based image and video retrieval is a viable methodology for many applications. However, traditional CBIR systems are neither scalable nor adaptable enough to be a viable

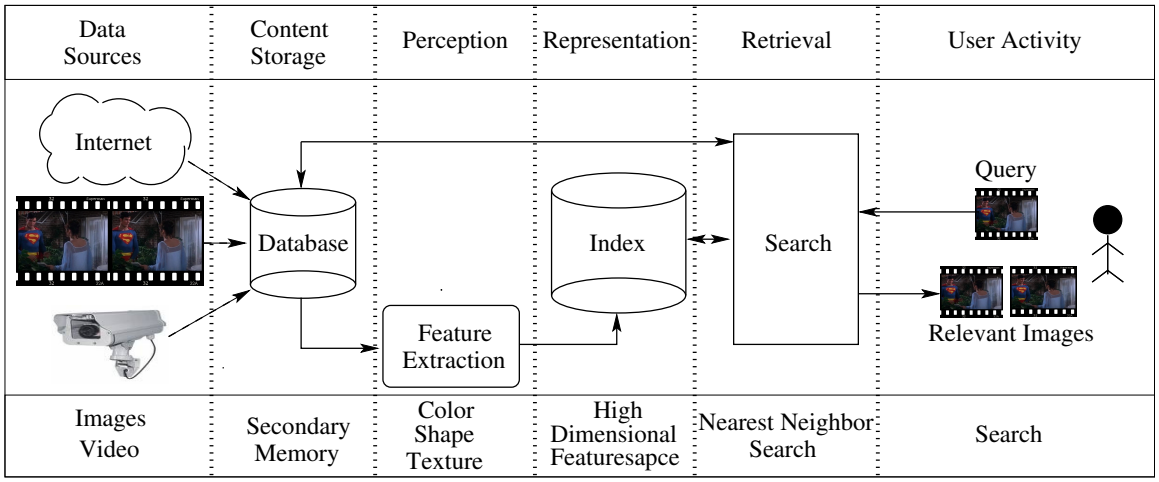

Figure 1.4: The figure shows the block diagram of a naive CBIR system. The features being lowlevel color, texture and shape features and the indexing being feature vector representation within the feature space. For retrieval K nearest neighbors is used in the feature space.

solution for largescale dynamic image databases. This lack of scalability and adaptability is one of the primary reasons why visual content based retrieval have not been adopted more widely. These are pertinent problems to solve. In this thesis we address the scalability problem of CBIR systems.

1.2 Objective

Our goal is to develop methods to retrieve images and videos based on their content in large scale and highly dynamic image and video collections. These large scale and dynamically changing (i.e. image and video collections where the visual nature of data is constantly changing with the addition of new content) collections resemble image/video databases that are created by active data sources like user generated content, crawling the internet, image and video feeds. Although, reasonably successful attempts have been made for certain aspects and applications, a holistic approach to the afore mentioned objective is missing.

1.3 Motivation

Achieving the afore mentioned objectives would ensure the viability of Content based retrieval as a solution for a whole slew of applications.

- **Image Search:** Image search of-course is the most obvious application of the methods and frame work. As of now *(circa April 2009)* regular image search engines only use text and hyperlink cues to retrieve images. Although the quality of their results have significantly improved over the years they are still agnostic to the content within the images. CBIR has not been a viable alternative or augmentation due to its inability to scale to millions of images. If these search engines are augmented with the ability to use content for retrieval the results will be massively improved.

- **Multimedia Search:** Video search is also akin to the image search. Right now videos have to be annotated by hand to show up in retrieval results which is neither fool proof nor a realistic way of scaling. For example, processing a 2 hour movie consisting of 10,000 keyframes, each comprising an average of 100 descriptors of 128 dimensions for a 20,000 word vocabulary using K-means would require around 72 hours. The methods developed in this thesis can be applied to video search to achieved far better results.

- **Multimedia Archives:** Searching multimedia archives is another prime example of where these methods can be used. Usually multimedia archives are annotated, filed and categorized manually. It would save a lot of money and effort while at the same time increasing the retrieval performance if this process can be effectively automated. A scalable and efficient retrieval scheme would also allow the capture the flood of multimedia at a much higher rate.

- **Copyright Infringement:** With the dawn of Web2.0 and UGC(User Generated Content) violation of copyright laws of multimedia have become rampant in their availability to the public yet are difficult to screen for and identify. The constant struggle to identify and remove such content has been evident on popular video sites like youtube.

Copyright violation of stock images over the net is yet another domain to look at. A scalable and adaptable system will be able to deal effectively with the millions of images that are crawled and indexed.

- **Satellite Imagery:** Classification and recognition of satellite imagery and surveillance imagery from reconnaissance aircraft is an application that has uses from defense to remote sensing. Automating processing and retrieval of such huge amounts of data could be a real cost saver.

- **Autonomous Navigation:** Autonomous navigation has become important for everything from vehicles to robots to cruise missiles. The ability to quickly query and retrieve information from large image databases is very important to such tasks.

While these are only some of the applications for real world image retrieval systems, their widespread use has not been possible due to various challenges one has to face in implementing such systems. In this thesis we develop methods that address the problem of content based retrieval in large scale dynamic databases. The emphasis of these methods is to make content based retrieval as scalable and adaptable as possible without much loss in the retrieval performance.

1.4 Contributions

The main contributions of this thesis are:

1. We developed and discussed methods for efficient, scalable and adaptable image retrieval from large scale and dynamic databases. These methods include transformation of color images into documents using *Virtual Textual Description(VTD)* with the help of *grid based vector quantization* for CBIR. The usage of '*Discriminative Relevance Feedback* based on VTD improved the retrieval performance of the system by incorporating a learning element to better model the query when compared to another

relevance feedback scheme called region based importance. We also proposed and new indexing scheme for this CBIR system called an *Elastic Bucket Trie(EBT)* that had better performance characteristics than spatial indexing for CBIR.

2. We designed and presented a novel method called incremental vector quantization(IVQ) for use in image and video retrieval systems with dynamic databases. We demonstrated the quality of the codebooks as well as their adaptability and speed of creation by using various standard and generic datasets. We look at this work as a promising development towards building effective codebooks for large scale *user generated* databases where huge volumes of new visual data is continuously added.

   We then proposed a method and a data structure that tackle representation of the term document matrix and on-line semantic indexing where the database changes. We introduced a bipartite graph model (BGM) which is a scalable data structure that aids in on-line semantic indexing, which can be incrementally updated. We also introduced a cash flow algorithm that works on the BGM to retrieve semantically relevant images from the database. We examined the properties of both BGM and cash flow algorithm through a series of experiments. Finally, we demonstrated how they can be effectively implemented to build large scale image retrieval systems in an incremental manner.

1.5 Outline of the thesis

The structure of the thesis is as follows: In chapter 2 we broadly review existing work in the field of CBIR, Spatial Indexing, Relevance feedback, Bag of words model, Vector quantization for image retrieval and Semantic indexing for image retrieval, chapter 3 introduces VTD based CBIR for color images, discriminative relevance feedback for improving retrieval performance and elastic bucket trie for efficient image retrieval. In chapter 4 we tackle the problem of scalable retrieval with Bag of Words model in a highly dynamic database. We develop and discuss the incremental vector quantization (IVQ) algorithm and show how it can be implemented in tandem with text retrieval engines. In chapter 5 we discuss BGM and

cash flow algorithm and show how they can be used for scalable and incremental semantic indexing over a large scale and highly dynamic database.

Chapter 2

Background

2.1 Introduction

This thesis has a wide context as far as domains are concerned. They range from Content Based Image Retrieval(CBIR) to Bag of Words (BoW) model. This chapter is meant to familiarize the reader with most of the relevant concepts and their state of the art when it comes to the issues this thesis addresses. The reader should note that going through [2, 3, 4, 5, 6] would be beneficial to gain a better understanding of some of the domains we deal with.

2.2 CBIR

Early research into image retrieval had begun, as far back as four decades ago. Both the database management systems and computer vision communities started working on image retrieval in the early 70's [2]. In the early days, the popular frame work for image retrieval was one that was built around manual annotation of images. Though significant advances were made ranging from query evaluation to multimodal indexing[7, 8], the systems were still primarily dependent on manual annotation for their retrieval. This kind of framework uncovered two significant problems.

- Manual annotation was not scalable for large image collections.
- Annotation of subjective content was inexact and dependent on the annotator and hence is inadequate in its scope.

Though further developments in the field have rendered the first problem irrelevant, the solution to the second problem however still eludes. This second problem was an early incarnation of what is referred to in the modern image retrieval literature as "The Semantic Gap". The 1990's saw the emergence of CBIR as a separate field of work in its own right. This was a direct result of the scalability issues with manual annotation. For the first time, images were being retrieved based on the very content they contain rather than the annotations provided by a subjective observer. They made use of visual **features** like color, texture and shape, which they extracted from the image in a process termed **feature extraction**. With the basic design of a CBIR system set, researchers started working in three primary areas[2]. They were visual feature extraction, multi-dimensional indexing and retrieval system design.

2.2.1 Features

In the early days of content based image retrieval, global feature based image retrieval was prolific. These schemes used primitive features of color, shape and texture over the entire image to retrieve relevant images. Global features view the image as a whole and calculate its features. Some of the predominantly used features are color histograms, color moments, color sets, gabor filters, co-occurrence matrix, shape context, etc. The shortcomings of such global schemes in effectively being used to retrieve images, is mentioned in detail in [9].

Later, spatial layout based schemes, sampled images in finer detail by dividing them into many small, usually equal sized parts. They then continued to extract the local features from each part. This evolved into the paradigm of region based image retrieval [3, 4, 5]. In this general framework, the image is segmented into different homogeneous regions based on either colour, texture, shape or all three of them. These schemes range from segmenting the image into objects to segmenting them into homogeneous color patches. These schemes

model the way in which humans perceive visual content better and there by obtaining better performance. However, accurate object segmentation, in general, is very costly in terms of computational resources. On the other hand, inaccurate segmentation leads to drop in precision of retrieval.

Research along this direction came into its own with pioneering work done by Carson *et al.* in their blobworld system [3]. Since then many improvements have been suggested to the general approach of region based image retrieval, the most notable of which was the work done by Wang *et al.* [4, 5]. Visual features and feature extraction are still active fields of research within the CBIR community. Visual features ultimately are used to retrieve similar images. The word "similar", however, must be disambiguated by mentioning that it only means similar in a feature space. The similarity of two images in the physical world however, is not constrained by their visual similarity. This gap between human perception of similarity and the machine perception of similarity is called the **semantic gap**.

2.2.2 Semantic Gap

The term *semantic gap* refers to the differences in perception and representation of the same information between two different entities. In the context of humans and computers this refers to the gap in the way each of them perceive, understand and describe the same data. For example in the Figure 2.1 the human reader sees a military jet flying while the computer is only able to understand a bit representation of that data. The machine is not aware that it is an aircraft, it has no context of what an aircraft is and how to understand what an aircraft looks like from binary data. This mismatch in human and computer ability to transcend the data and understand the true content underlying it is the essence of the semantic gap and the problems it creates. This semantic gap between man and machine can be attributed to three primary discrepancies between man and machine. They are

- Visual perception: The computer sees RGB values while the human sees objects and scenes.

Figure 2.1: A Sukhoi30 aircraft

- Training data: The computer is not aware of other images with the same higher level concepts, while the human has apriori data of almost all visual information he comes across.

- Learning model: The computer has no way of associating, grouping or categorizing images over time. The human brain achieves that seamlessly.

The primary purpose of any CBIR system is to bridge the semantic gap by tackling the discrepancies in these three areas and ultimately try to model how humans see and understand images.

Visual Perception

Human vision is a product of millions of years of evolution. It is an incredibly complex system from the eye to the visual cortex. Human vision has been an active area of research for researchers from varied fields from medicine, neuroscience, cognitive sciences and computer scientists, yet a consistent model of human vision still remains elusive. [10]. The way in which humans see the world has been consistently redefined as more and more studies are done

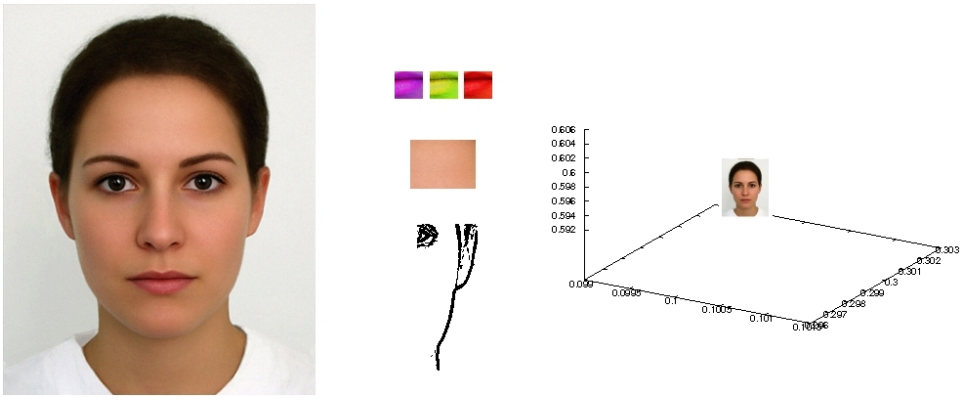

Figure 2.2: How an image is interpreted by a computer using color texture and shape features

on the topic. However when concerning CBIR one needs to take some primary attributes of human vision into consideration.

- Our ability to sense color
- Our ability to sense shape
- Our ability to sense brightness
- Our ability to sense depth

In order to replicate a model of human vision on a computer one must take these factors into consideration. Early CBIR systems and most current CBIR systems still use low level *Color, Texture and Shape* features [2] to model human vision. With these feature vectors the computer tries to replicate rudimentary visual perception that is somewhat similar to human visual perception as seen in Figure2.2

Training Data

From the moment we are born the human eye senses an enormous amount of visual input. Along with this we also experience a wealth of other sensory input. When all these are combined together in both simple and complex ways over time it forms our knowledge of

Figure 2.3: A mosaic created from more than 53,000 categories of more than 7 million images. Courtesy 80 million tiny images project *Torollba et al*[1]

the physical world. With this enormous amount of knowledge it is a trivial task for humans to recognize, interpret, associate and categorize things by their visual appearance alone. A computer however does not have either the inbuilt mechanism to understand the data nor that amount of data available to it. In CBIR the usual approach of providing this *training data* to the computer is by using *relevance feedback* mechanisms over a large number of images(refer to Figure2.3) to allow the users to group images into classes. Hence a rudimentary mechanism of collecting data and aggregating knowledge by grouping or association is setup along with their relevance to visual features.

Learning Model

The learning model in a CBIR system is system that tries to replicate human knowledge of visual classes by trying to learn what visual features correspond to what concepts. For example, in Figure 2.4, each of the images represents an apple yet they vary vastly in color, shape and texture. Even with the vast amount of variation the human brain is able to recognize all of them as apples. This is due to some abstract model of apples that the human brain has constructed from all its past experiences. The objective of a learning CBIR system

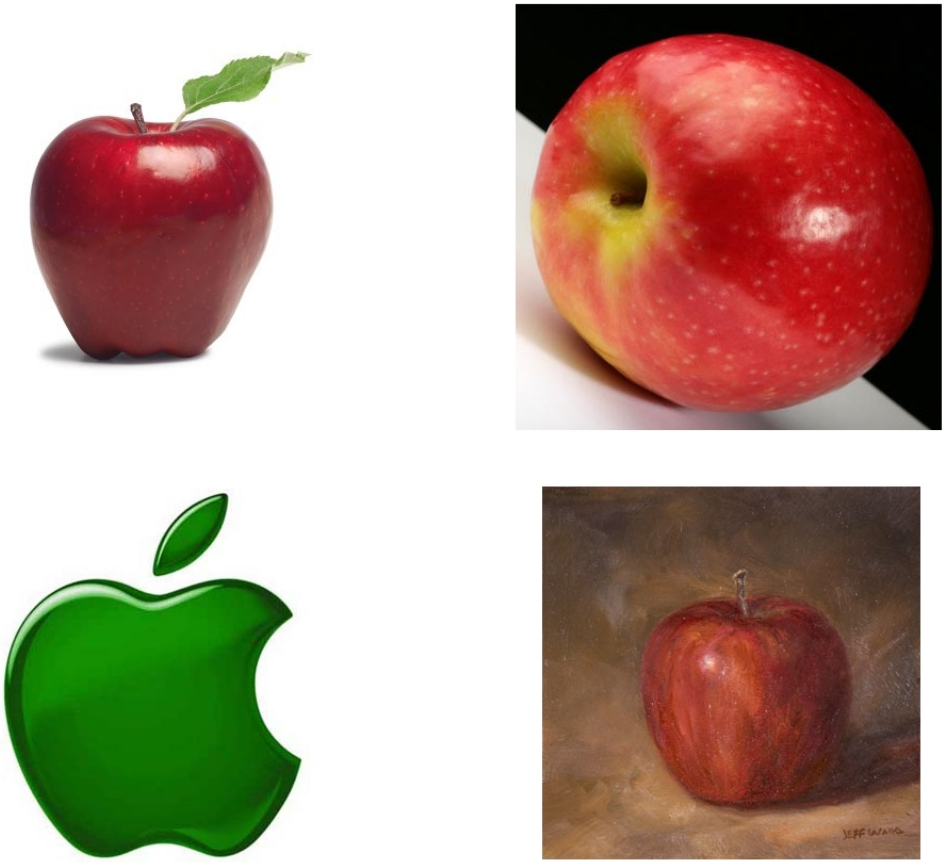

Figure 2.4: A set of images containing an apple each

is to construct a model as similar to the user's abstract model as possible in an arbitrary feature space with the given training data [11]. These learning algorithms can be grouped into generative, hierarchical [12, 13], discriminative or hybrid [14, 15] algorithms. These models along with the models for visual perception and training data are an oversimplification of the way humans understand the world visually. As a consequence though significant steps have been taken in bridging the semantic gap, the divide however still remains too wide for large scale application of CBIR.

2.2.3 Relevance Feedback

Early CBIR systems were only able to retrieve images based on the nearest neighbors in a feature space. Though the human operating the system was able to provide a wealth of knowledge to the system these systems were not able to leverage it. What was needed was a viable way for CBIR systems to learn from the interactions by getting the human into the loop. This lead to the widespread adoption of relevance feedback in CBIR systems. Relevance feedback [16] is a technique adopted by Image Retrieval researchers from text retrieval to improve performance of CBIR systems. The typical process of relevance feedback is as follows. For a given initial query the CBIR system fetches the N-nearest neighbors of the query in an arbitrary feature space using arbitrary distance metric. Once presented with the retrieved images the user critiques on them by choosing the relevant images. All the relevant images and in some cases non-relevant images are used by the relevance feedback algorithm to either refine the query or other variables that effect image retrieval. Having changed it's internal model to be more inline with the user's model of the concept the next set of nearest neighbors are retrieved and presented to the user. This process is reiterated over time.There are a large number of varied algorithms for relevance feedback yet they fall into three main classes namely, statistical relevance feedback algorithms, kernel based relevance feedback algorithms and entropy based relevance feedback algorithms.

Statistical Algorithms These were the earliest methods of heuristic weight adjustments. They used the nature of the distribution of relevant data in the features space to effectively cluster relevant examples. Most of these methods try to take advantage of the fact that under certain transformations the image database can be clustered into relevant and irrelevant images Or where the relevant images become clustered and the irrelevant ones become sparsely dispersed. The relevance feedback data is used to achieve this transformation.

Kernel Based Algorithms These methods use some kind of kernels to achieve relevance feedback. SVM (support vector machine) based algorithms primarily dominate in this class and have become even more prominent in recent years [17].

Entropy Based Algorithms Entropy is an estimation of the deviation of a random variable from pure randomness. They have also grown to incorporate algorithms using information gain, mutual information, active learning and other information theoretic methods.

Other Algorithms Relevance feedback has been a very active field of research in the CBIR community for more than a decade. New algorithms are constantly being suggested, some of which do not fall into any of the above classes. These include SOM (self organizing maps) algorithms, neural networks, decision trees and other approaches.

Shortcomings

For a CBIR system to work well many subsystems must come together successfully and many variables must be set right. A detailed study of this can be seen in [11]. However there are two important factors whose mismatch limits the effectiveness of a relevance feedback systems.

- **Feature Space:** The feature space is a constrained by the low level feature vectors extracted from the images. If the feature space is not appropriate then learning by relevance feedback will not be able to improve the retrieval performance.

- **Learning Algorithm:** The learning algorithm must be capable of appropriately modeling the user behavior over the feature space. If it cannot do that, even a good feature space is of little consequence when it comes to performance.

Often one finds the relevance feedback is far less effective in CBIR than in text retrieval. The primary reason being that low level image features fail to capture the semantics of an image as effectively as words do in text documents.

2.2.4 Indexing

Multimedia data is hard to index using regular database management systems. Traditionally the CBIR community widely preferred spatial data structures for deploying multimedia databases. These were predominantly R-Trees [18], X-Trees [19], S-Trees, Variants of S-trees and R-Trees like R*-Trees [20], S*-trees and later there were also TV-trees [21]. All of these tried to efficiently index spatial data or multi dimensional data like image, video and audio for building efficient retrieval systems. These data structures are better suited for global feature based image retrieval schemes than region based methods. Yet some have adapted them to work for region based image retrieval as done by Carson *et al.* [3]. However, with the advent of relevance feedback techniques in image retrieval, spatial data structures have become inefficient. Once relevance feedback is used the traditional "spherical" or "window" queries in the feature space are transformed into highly elliptical and other shapes making the usual spatial data structures very inefficient. Since relevance feedback changes the query continuously in shape and dimension the spatial data structures have been found to be inefficient.

2.3 CBIR and Bag of Words

In recent years CBIR has been incorporating methods from the object recognition, object classification and text retrieval communities. These include better local detectors and descriptors from the object classification community, better image modeling, retrieval methods

and document indexing adapted from text retrieval community and better semantic analysis methodologies borrowed again from the text retrieval.

2.3.1 Local Descriptors

The new wave of local photometric descriptors are undoubtedly the base on which the modern object classification research is built upon. They have been heavily used in all kinds of scenarios like object recognition [22, 23, 24, 25], object classification [26, 27], image retrieval [28, 29], robot localization [30]. Most of these modern descriptors are scale and rotation invariant, they are also robust to affine lighting change and are very distinctive. These attributes have caused a paradigm shift in whatever fields they have been effectively used.

Region Detectors Usually the first step towards computing local descriptors is detecting regions within the image for whom the descriptors will be calculated. For this task, region detectors are used. There are a multitude of region detectors belonging to many classes but all of them have to meet some common criteria to be considered good region detectors.

- They must be scale invariant to cope with the same content being projected at different scales/distances.
- They must be rotation invariant to cope with in the plane rotation of the camera/scene.
- They must be robust to affine photometric changes to cope with out of plane rotations of the camera/scene.
- They must have high recall for robust matching of corresponding local descriptors.

These detectors include scale and affine invariant detectors, blob detectors, affine covariant detectors. These include DoG(Difference of Gaussian), LoG(Laplacian of Gaussian), MSER, Harris Affine, Hessian Affine and many such detectors. A detailed study of the performance of many such detectors was carried out by *Mikolajczyk et al.* [31].

Figure 2.5: Original image of an artist's graffiti and the corresponding affine covariant regions detected using Harris affine detector

Descriptors Local descriptors are used to encode image point or patch data from the interest point or region detectors. The aim of local descriptors is to encode the image patch into a representation that has the following qualities.

- They are highly distinctive: chances of false positives and false negatives are low.
- Invariant to affine photometric changes: robust to changes in lighting and brightness.
- Invariant to rotation and scaling: able to cope with rotation and scale changes.

In recent years, distribution based descriptors have found great success both in research and commercial applications. These include SIFT[32], PCA-SIFT[33], GLOH[34], SURF[35], LESH[36]. SIFT (Scale Invariant Feature Transform) is however the one that stands out as the first and is the most widely used local descriptor both for research and commercial purposes.

Bag of Words The *Bag of words* model is a text retrieval technique that simplifies model of a text document to be just an unordered set or collection of words or terms. Most text search engines use this model to deal with documents. More recently the **BoW** or Bag of Words model has been adapted to computer vision applications(See Figure 2.7) like object categorization and object recognition [37]. Here the image is represented as

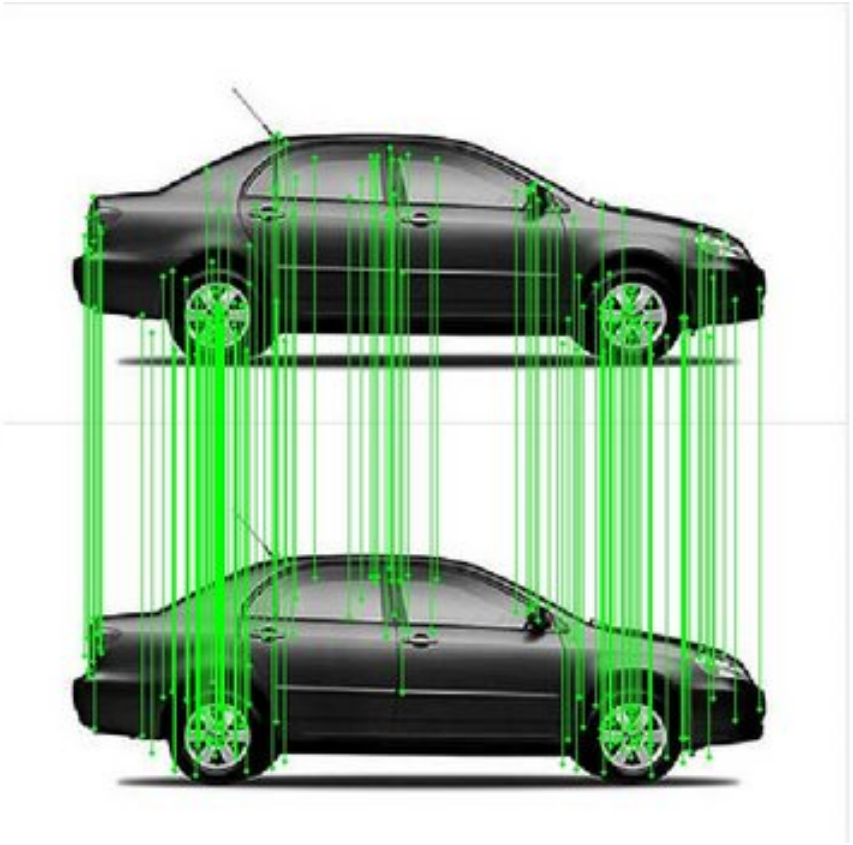

Figure 2.6: SIFT matches for Toyota Corolla

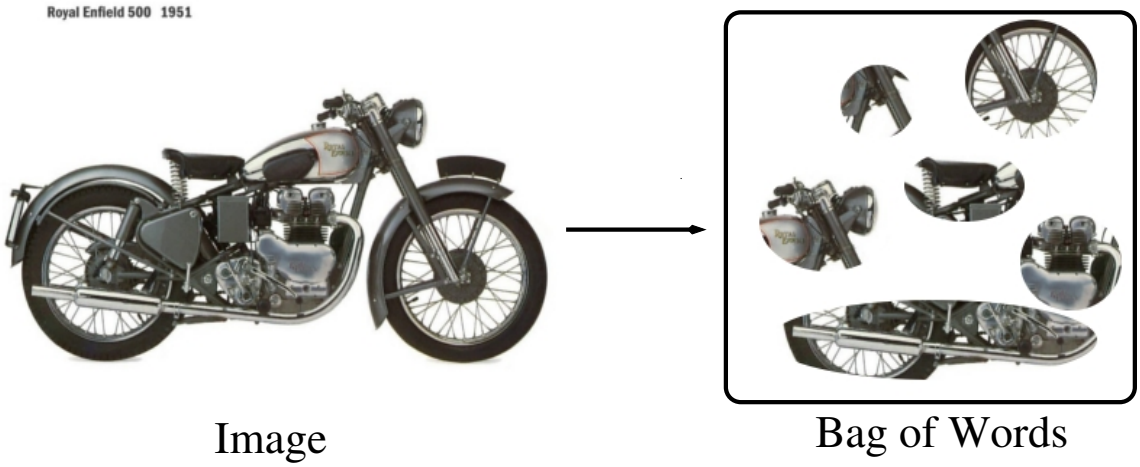

Figure 2.7: The above diagram shows an example of the bag of words model in computer vision. Here the image is sampled and image patches are extracted using local detectors. These patches are further encoded into a feature vector using local descriptors. Each of these image patches is a visual word. Finally the image is represented as a collection of image patches using the bag of words model. The important thing to note is that the spatial consistency among the words/patches is not maintained in the BoW model

a distinct set of *visual words* or a visual word histogram. The visual words are usually arrived at by vector quantization in an arbitrary feature space. This model has been used to tackle many vision problems[38, 39, 6, 40, 41, 42, 43, 44, 45, 46, 47] with very good results. These approaches are shown to be well suited for tasks such as object categorization, object recognition, object retrieval and scene classification. The success of these approaches, in large part, is due to the model's ability to accommodate natural scene variance in the form of pose changes and occlusion. The quantization of a very high dimensional feature space (using an algorithm like Kmeans)[48, 49] to build a compact codebook that encodes the similarity between descriptors, paves the way for efficient retrieval systems. The power of bag of words model to create efficient image and video retrieval systems has been explored by Sivic and Zisserman[6] as well as Nister and Stewenius[50]. The problem of building large scale image retrieval systems has also been looked into by Torralba *et al.*[1], though

not utilizing the bag of words model. State of the art retrieval systems describe the images by sparse or dense descriptors and index them in an offline phase to build highly scalable retrieval systems.

In text retrieval the bag of words model is intuitive and simple. However, when it comes to images the application of this model is not immediately apparent. The model architecture as seen in Figure 2.8 shows how images are transformed and used as documents in a bag of words model. First the data acquired from multiple sources is stored in the content database, then feature extraction is done on each image. Typically, interest point detectors are first used to identify interesting regions within the image. Figure 2.7 shows the detected regions in the corresponding image.The detected regions are then encoded with a high dimensional visual descriptor like SIFT[32]. Once the feature vectors are extracted they are quantized using vector quantization. Vector quantization converts feature vectors into symbols or words by quantizing the feature space into discrete cells. The resulting words are all considered independent of each other. These words are then used to index the image and retrieve relevant images when appropriate.

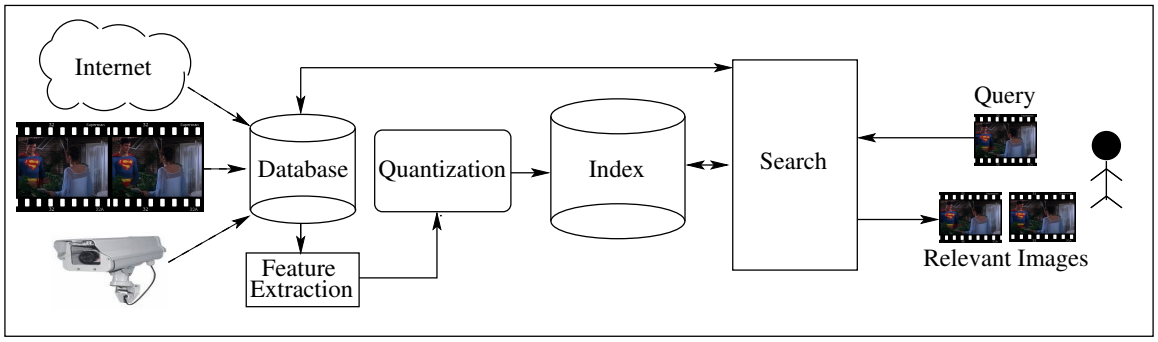

Figure 2.8: The above block diagram shows the architecture of a BoW based image retrieval system. After data acquisition from various sources the features are extracted and vector quantization is done. Vector quantization converts feature vectors into symbols or words. These words are then indexed in the index and are used for retrieval as and when required by the search module.

2.3.2 Vector Quantization

Vector quantization or feature space quantization is used to discretize a feature space into visual words. The discretization of image features allows the problem of image retrieval through nearest neighbor search in high dimensional spaces to be posed as a search and retrieval problem in a document collection[6]. This transformation makes large scale image retrieval systems viable. A good quantization algorithm for large scale highly dynamic image retrieval systems should be highly adaptable to new data, be able to accurately represent underlying data and build compact codebooks(vocabularies created by quantization of the feature space) that are robust to new data. Vector quantization is an integral part of image retrieval using the bag of words model.

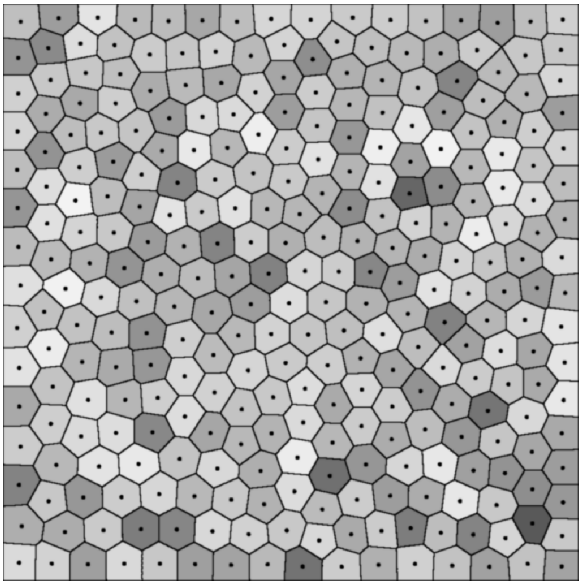

Figure 2.9: The above vornoi diagram is a graphical representation of vector quantization using K-means on two dimensional data. Here the vornoi cells represent the visual words in the feature space and the dots represent the means. Each vornoi cell is associated with an identifier or a symbol and all the feature vectors in the feature space are labeled with the symbol of their relevant cell. The cells form the vocabulary of the bag of words model

The most dominant feature space quantization or perceptual coding algorithm in use is k-

means clustering algorithm [51, 52, 48, 49]. k-means aims to partition n observations (feature vectors) into k clusters in which each observation belongs to the cluster with the nearest mean. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data iteratively[53, 54, 55]. At every iteration the observations are reassigned to the relevant means until the means converge with respect to a given criteria (maximum iterations, no change in means, etc).

Agglomerative and online quantization techniques have also been explored [56]. Agglomerative methods usually begin with each element as a separate cluster and merge clusters using some criteria. Agglomerative clustering algorithms are a good candidates for hierarchical clustering too. Hierarchical clustering algorithms [50, 57] coupled with supervised learning of randomized decision trees and forests [58, 59] have gained prominence for object categorization, image classification and image segmentation. More recently adaptive vocabulary forests[60] and discriminative visual vocabularies have also been used [61, 62].

2.3.3 Semantic Analysis

Latent semantic analysis is a text retrieval concept which is used to analyze a set of documents so that the latent *concepts* and their relationships to the documents and words in the documents can be uncovered. The quality of the retrieval using bag of words model is further enhanced with the help of semantic indexing techniques like probabilistic Latent Semantic Analysis(pLSA)[63] and Latent Dirichlet Allocation(LDA)[64]. Semantic analysis of a document corpus can be viewed as unsupervised clustering of constituent words and documents around hidden or latent concepts in the corpus. Adaptation of PLSA and LDA to visual bag of words has provided promising results for static image databases[65, 66, 67, 68]. More recently semantic analysis is also being used in conjunction with spatial constraints for object segmentation [69, 67, 70], scene classification [47] and model learning [71, 72, 73]. Initially LSA or LSI was done by decomposing the co-occurrence matrix generated from the document-term vectors of a document set using SVD. This however did not have a solid statistical accurate which was achieved to some degree by **pLSA** [74], or probabilistic latent

semantic analysis. Both pLSA and LDA (latent Dirichlet allocation) found success in the text retrieval community. With the introduction of Bag Of Words models for images, in recent years both pLSA and LDA have been successfully used in Object Recognition [75] and Scene Analysis [64]. Though pLSA and LDA are very good at achieving what they are meant for scalability is a very serious concern.

2.3.4 Indexing

The Bag Of Words model lends itself well to implementing text based indexing methods and thereby improving efficiency of the system. Yet this has been a very sparsely explored area in object recognition and image retrieval. Few notable examples are the Video Google system [6] which used an inverted filesystem to retrieve relevant images and a voting based indexing system by Mikolajczyk *et al.* [76]. An inverted index is an index that stores a mapping from words to the documents in which those words occur. The inverted index is usually used to carry out text search[77]. TF or term frequency is the number of times a term t_i appears in a given document d_x and IDF or inverse document frequency is inversely proportional to the number of documents a term t_i occurs in the document corpus D_x. TF, IDF values are used for ranking retrieved results. Here the intuition is that the greater the TF and IDF values the more important the term is in the document.

2.3.5 Largescale Databases

CBIR has traditionally not been able to scale effectively to image collections beyond thousands of images. As the size of the database grows so does the need to represent it more distinctively. However as the dimensions of the feature space grows we run into *the curse of dimensionality*. The curse of dimensionality hints at the problems caused by the exponential increase in the volume of the feature space with a linear increase in the number of dimensions. The curse of dimensionality is also the primary cause for concern when dealing with the scalability of CBIR. Since CBIR either uses nearest neighbor search[78] or spatial indexing to retrieve relevant images one sees the decline in performance as the number of

dimensions grow. Spatial indexing datastructures like R-Trees [18], X-Trees [19], S-Trees, Variants of S-trees and R-Trees like R*-Trees [20] ,S*-trees, TV-trees [21] were used to index the feature space. All of these tried to efficiently index spatial data or multi dimensional data like image, video and audio for building efficient retrieval systems. However the situation has changed with the incorporation of relevance feedback into mainstream image retrieval. Once relevance feedback is used the traditional "spherical" or "window" queries in the feature space are transformed into highly elliptical and other shapes making the usual spatial data structures very inefficient. Since relevance feedback changes the query continuously in shape and dimension the spatial data structures have been found to be inefficient. There have also been attempts to accelerate nearest neighbor search in high dimensions using LSH(locality sensitive hashing) [1, 79, 80, 81] however these schemes do not take into account the skewing effect relevance feedback has on the feature space. The inability to accommodate this skew degrades the CBIR system's retrieval performance.

2.3.6 Dynamic Databases

With the introduction of bag of words based image retrieval a new problem has come up when dealing with dynamic image databases. Dynamic image databases are databases whose nature(defined by kind of visual concepts the images contain) is constantly changing with the constant addition of new images. However, two important parts of contemporary bag of words based image retrieval are not good at dealing with a dynamic database of images. State of the art BoW retrieval systems describe the images by sparse or dense descriptors and index them in an offline phase to build highly scalable retrieval systems. As the database dynamically evolves, the codebook is unable to accurately represent the underlying data. This necessitates the re-computation of the codebook at regular intervals. As the number of images and associated visual concepts increase, the computation becomes prohibitively expensive even for thousands of images on commodity hardware, often taking days or months to compute. To scale offline quantization to large scale databases the data is usually sampled and a small percentage of the images are used to compute the codebook. Even, in

these cases traditional vector quantization methods cannot scale effectively[57]. This has resulted in the development and application of new methods like hierarchical and approximate Kmeans algorithms for building codebooks [50, 57]. Agglomerative and on-line quantization techniques have also been explored [56]. Quality of the model has also been optimized using discriminative visual codebooks [61]. However, any such database specific offline approaches are not extendable to the situations where the database is constantly evolving. Semantic indexing in a dynamic image collection also poses a considerable challenge. As new images are constantly added to an image collection the semantic index is unable to accurately represent the changing database. This necessitates updation of the semantic model and indexing it at regular intervals which is time consuming and not scalable for large databases with millions of latent concepts. As the number of images and associated concepts increases, these computations become expensive. Traditional semantic indexing methods range from statistical methods like Latent Semantic Indexing [63] to probabilistic generative models like Probabilistic Latent Semantic Analysis(pLSA)[63] and Latent Dirichlet allocation(LDA)[64] and their incremental variants like incremental pLSA proposed by *Wu, et al.*[82]. One of the factors that make these methods challenging to adopt in a dynamic setting is their computational complexity. The other factor that makes the adopion of these methods challenging is the selection of number of global semantic topics. Even for incremental pLSA selecting the number of latent topics in a changing database of millions of images is difficult.

2.4 Challenges for the work

One faces a significant number of challenges when trying to build such a system.

- **Vector quantization** is an important step in efficiently handling high dimensional data if one wants to avoid costly nearest neighbor calculations in high dimensions. When the amount of data and feature space becomes huge vector quantization and its relevant parameters are not so straight forward and are dependent on many issues. Getting their balance right is very hard and at the same time highly relevant to the

performance of an image retrieval system.

- **Semantic Indexing** Algorithms like LSA and PLSA that have been known to improve retrieval performance are not scalable when it comes to huge datasets. Suitable alternatives must be found that are both scalable while providing a meaningful improvement in retrieval performance.
- **Relevance Feedback** Relevance feedback is a technique by which the systems understanding of a query is continuously updated hopefully for the better. Traditional relevance feedback methods are not that computationally efficient in high dimensional spaces.
- **Indexing** Indexing is one of the primary factors that affect's the speed and scalability of a retrieval system. Traditional CBIR indexing and retrieval from spatial indexes is inefficient for data in high dimensional spaces.

2.5 Vocabulary

In this thesis much of the vocabulary pertaining to CBIR and image retrieval is used interchangeably, following are some terms that you might encounter in the rest of the thesis and their meaning and context.

- The phrase *Image retrieval* often means content based image retrieval and is not indicative of image retrieval using textual cues
- The phrase *Bag of Words* is most widely used to indicate bag of visual words rather than the text retrieval model, unless it is explicitly specified.
- The phrases *semantic indexing* and *semantic analysis* are used interchangeably.
- The phrase *dynamic database* or *highly dynamic database* is used to refer to multimedia databases where the visual nature of the multimedia in the database constantly changes due to the addition of new data to the system.

Chapter 3

CBIR for largescale databases

Traditional CBIR systems find relevant images by finding nearest neighbors in a high dimensional feature space. This is computationally expensive, and does not scale as the number of images in the database grow. We address this problem by posing the image retrieval problem as a text retrieval task. We do this by transforming the images into text documents using grid based quantization of the feature space. This text description of image is called a Virtual Textual Description (VTD). Once this transformation is done, we further enhance the performance of the system by incorporating a novel relevance feedback algorithm called discriminative relevance feedback. Lastly we use the virtual textual description of images to index and retrieve images efficiently using a novel datastructure called the Elastic Bucket Trie (EBT). We show how EBT compares to traditional spatial indexing methods and discuss its adaptability to adapt effectively to relevance feedback algorithms.

3.0.1 Structure of Chapter

We propose a novel general representation where images are treated as documents, and segments are treated as keywords. The virtual textual representation transforms the CBIR problem into a modified text retrieval problem, thereby allowing us to use the wealth of knowledge to tackle the general problems in CBIR (Section 3.1). We demonstrate the use,

practicality and performance of our virtual textual representation scheme with an example implementation and a pictorial example. Using this representation, we develop a discriminative relevance feedback scheme creating a unique blend to improve both performance and flexibility. The proposed relevance feedback scheme, tries to find the discriminative regions instead of the salient regions to improve the retrieval (Section 3.2). These regions are discovered in a way that can aid long term learning and at the same time refine the results at each iteration. We validate our scheme under different conditions through a series of experiments (Section 3.3). We also show that our scheme can be extended to achieve better performance without trading it for flexibility. We then introduce a modified elastic bucket trie for indexing and retrieval scheme for image databases (Section 3.4). It is much more efficient than the traditional spatial data structures used to access multimedia data and is at least one order better than these schemes. Our scheme is also able to work without any modification with relevance feedback schemes as is required by spatial indexing and retrieval schemes.

3.1 Virtual Textual Description

Images by their nature are subjective. Their content cannot be effectively described in a quantitative manner. When humans describe an image they do so by extracting objective features or concepts like sky, clouds, flowers, cars, bikes, people etc. This however cannot be done by a contemporary CBIR system as it is not capable of comprehending these concepts. Instead the image can be seen or interpreted by these systems in the form of primitive features. These low level features are computed from pixels or patches. There is a gap between these low-level representations and the high-level concepts, popularly known as the semantic gap. In order to bridge this gap of subjective visual features and objective high level concepts, Carson *et al.* [3] and Wang *et al* [5] developed an objective low level feature representation and retrieval framework called region based image retrieval. In these methods generally the image is divided into objective segments such that each segment is

homogeneous in nature in some visual characteristics, which means that the image is a collection of segments that are visually coherent concepts in themselves. The aim of region based image retrieval is to find some mapping of the concept that the user is looking for on to a set of segments [83]. If this can be successfully done the concept can be deduced as a set of segments by the system, thereby being able to bridge the semantic gap to some extent. We take region based retrieval one step further by proposing that a set of visual segments representing a visual concept is much like a set of words representing a subjective intention, or like a set of words making a coherent essay with a central theme. Drawing such parallels to text documents we further try to quantize the visual concepts by converting the segments into words and the image into a text document comprising of these words.

In our virtual textual representation an image is referred to as a document and its segments are referred to as keywords. Such a transformation is advantageous as one can now solve the CBIR problem as a modified or a special case of text document retrieval problem. Once the image has been divided or partitioned into visually coherent and compact units or segments, each segment is transformed into a string called a keyword. These keywords are obtained by binning visual features and applying a linear or nonlinear transformation. The segments are transformed into words such that segments that are visually similar to each other have the least hamming distance in their strings. Such a transformation may at first seem lossy however such a transformation actually improves the generalization capabilities of the system. Once the segments have been transformed to keywords and the images converted to documents we cannot directly use cosine distance to find the distance between two images as done in text retrieval. This is because in text documents each word is an atomic unit where changing even a character would mean the meaning of the word is lost. However in our virtual textual representation each character is an atomic unit and these atomic units put together to form a keyword. Hence we need to solve the problem differently.

A sunset described visually in terms of color by a human would be something as follows. sunset $\rightarrow$ (Orangish *or* Reddish) Hue on Top AND (Yellow *or* Bright Yellow) Hue in the middle.

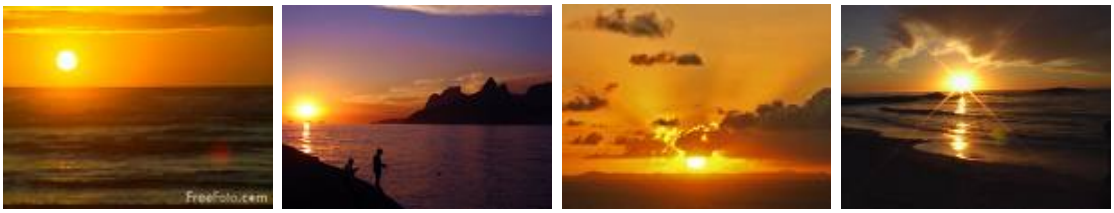

Figure 3.1: Images of sunsets with a lot of variation can accommodate the afore mentioned visual description of (Orangish *or* Reddish) Hue on Top AND (Yellow *or* Bright Yellow) Hue in the middle.

Human beings tend to describe visual content as a group of visually coherent regions. Hence we can see that the sky is expressed as orangish or reddish hued region on top. Such a general description of a sunset allows for a lot of variation as does the human recognition of a generic sunset. The concept of 'Sunset' is, by definition, visually and conceptually broad and inexact in nature. This broad description allows us and the scheme to accommodate other visually different concepts like clouds and buildings in the sunset image.

An image can be described and distinguished as a collection of regions or segments in order to better handle the content. Here the image becomes a collection of discrete visual concepts that are put together to form one visually coherent concept. This is like a bunch of words put together to form a coherent essay or document or description. We hence draw the parallels between the logical compactness of words and segments in images and documents. For example we see that for a concept sunset orangish, reddish, yellow, bright yellow are keywords in textual form. This is carried on into the image domain where images are modeled as text documents and segments are keywords of these documents. Such a modeling tries to mimic human visual interaction or description rather than human visual perception. Hence visual concepts can be communicated effectively between the user and the system.

In our scheme, an image is treated as a visual document akin to a text document and the major or the important segments of the image are treated as keywords in the text document as seen in Figure 3.2. Once the image is segmented each segment is visually described in the form of a word where the word is a 6 character string(specific to our implementation) in-

stead of linguistic representation like “Orange” or “Blue”. This word is the result of binning visual features of the image and applying a linear transformation to obtain a 6 character string in the text domain. This six character string is called a “keyword” and each image is called a “Document”. The nature of these Keywords is such that they are inherently broad or inexact representations of their respective segments unlike numerical representations. In our scheme, the distance between two documents cannot be calculated by cosine distance as in document retrieval. This is because the keywords themselves have a distance between them which incorporate more fuzziness into the scheme and as a consequence robustness. We use hamming distance to calculate distance between two keywords and hence two segments. Consequently least cumulative hamming distance between two images produced by any configuration is used as the “Inter Document” or “Inter Image” distance(Section 3.1.1).

A representation of an image as a document and segments as keywords, allows us to pose the CBIR problem as a special “Text Document Retrieval” problem. Such a transformation has the promise to improve the ability to index and retrieve images based on content using accumulated knowledge and practices in the text document retrieval domain. Existing proprietary or open source database systems can be used to store and index the images and also to efficiently retrieve these images. This would not be possible using the conventional feature based representation and spatial databases would have to evolve. Our representation can become translation and transformation independent as and when required automatically by dropping the importance associated with positions of the segments. Our scheme can also handle occlusion as the segments are independently modelled, and occlusion of one or more of the segments will be handled gracefully.

3.1.1 Grid Based Quantization and Image Retrieval

The image is initially mapped into an appropriate color space which represents human visual perception much more accurately. This image is then quantized into a discrete number of uniform bins in the feature space. The image is then segmented based on the color and spatial constaints. The segmentation algorithm is a heuristic algorithm designed to be much

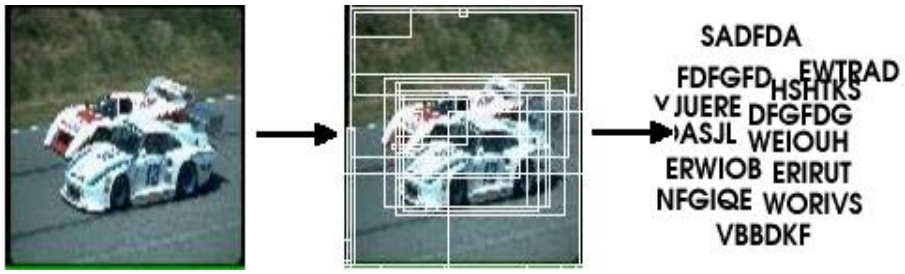

Figure 3.2: An example of an image being converted into virtual textual representation. First the image is segmented into different parts or visual words, then these parts are transformed into words by quantizing the individual colour, texture and shape features within each visual word. Finally we have a virtual textual representation of the image

more robust and handle occlusion or collection of similar objects. The segmentation is very efficient when compared to other contemporary implementations[4, 5] of region based retrieval. It can afford this efficiency because of the concept refinement features built in to the scheme through relevance feedback that make up for the loss of segmentation accuracy.

Once segmented, each segment is treated as a visual word. This visual word is converted into text by a linear transformation as shown in the Figure 3.3.

When an exemplar image is given as a query, its representation (collection of all the keywords) Q is extracted by the feature extraction module, where Q_i is the ith keyword in the document. Every other image document K_j is compared with Q to obtain a similarity score S_j for image documents Q and Kj.

$$S_j = \prod_{i=1}^{n} max(H_{k=1}^{m}(Q_i, K_{jk})) \tag{3.1}$$

$$H = (6 - hammingdistance + 1) \tag{3.2}$$

Where n is the number of keywords in Q and m are the number of keywords in K_j. Once we get all the S_j we have.

$$\left(\; S_1 S_2 S_3 \ldots S_{m-1} S_m \; \right) \tag{3.3}$$

We then sort the S_j and take the top N images or documents as the most relevant. Here the *hammingdistance* is subtracted from 6 to convert the distance metric into a dissimi-

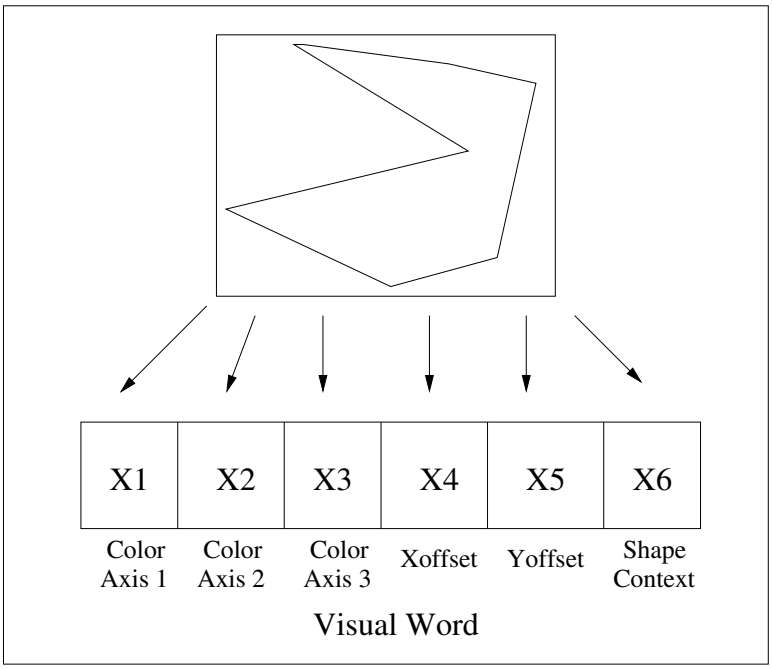

Figure 3.3: The above figure demonstrates how a visual word is converted into a text or symbol representation in the example implementation. Here X1, X2, X3 are the symbols assigned to quantized bins in the colorspace. X4 and X5 are the quantized x and y offset of the segment from a reference and X6 is the shape context of that particular image.

larity measure. This scheme is also very efficient as the problem has been modeled into a partial string matching problem, where earlier floating point calculations were heavily used. Now the calculations can be made with simple bit operations instead of costly floating point operations. The above described linear transformation is but an example of a way in which an image can be transformed into a symbolic or textual representation. This however might not be suitable for all situations, for example situations where there are really dense clusters separated by sparse spaces in the feature space. In such cases the sparse areas are over sampled and the dense areas are undersampled. Hence different situations would require different quantization schemes but the general framework of the scheme will remain consistent. Usually in a normal region based image retrieval, if 50 to 70 segments are produced and each segment is described by 6 to 7 floating point numbers as features. In our case we use

6 to 7 symbols to represent each feature vector, or a 6 character string. Already space efficiency is achieved by our representation. Further, each floating point distance computation (Minkowski) involves several complex arithmetic operations like square root, cube root, addition and subtraction. This makes floating point based region based image retrieval 50^2 to 70^2 times more inefficient when compared to global feature based methods. Our method on the other hand uses bit operations and text indexing to achieve almost quasi linear execution performance, making it atleast 10 times more efficient than the traditional schemes.

3.2 Discriminative Relevance Feedback

Recent years have seen the development of many relevance feedback strategies for region based image retrieval as in the work done by Jing *et al.* [84]. But most of the existing systems still use relevance feedback techniques built for global feature based image retrieval. Other region based relevance feedback algorithms make use of region weighting to achieve retrieval. Such techniques do not effectively distinguish a class of images in the presence of other classes in the database. Rather they tend to cluster images based on the nature of the relevant class which may lead to accidental biases toward unimportant features or regions, like the concept of 'road' when one is looking for the concept of 'car' because the visual concept 'road' commonly occurs with that of the concept 'car'. At the same time not much work or attention has been given to the efficiency and indexing of region based image retrieval schemes. Our relevance feedback scheme differs from contemporary relevance feedback schemes. Most of the schemes try to either obtain a region weighting or try to extract the regions of these images based on which regions are most dominant in the relevant images. Such schemes have a tendency to become biased toward features that do not actually represent the concept. Other schemes finding the most salient regions in an image which can also lead to similar bias. For example a couple of "Red Buses" will lead the system to deduce that the regions with red are the important regions for the concept "Bus" which is clearly not the case(the correlation between the color red and the concept bus is

incidental and not true in all cases).

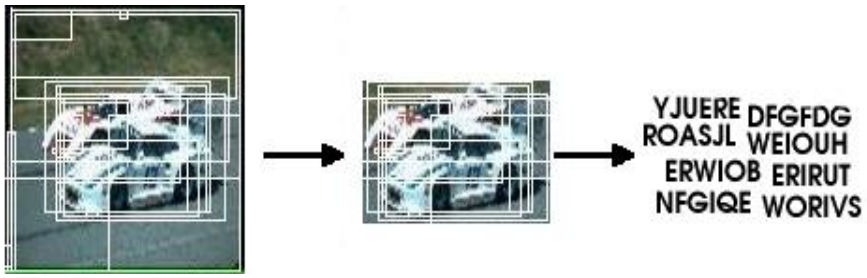

Figure 3.4: The different words or image patches that make up the car are further refined during discriminative relevance feedback and a only the most discriminating words are retained. This improves both the classification performance and the efficiency of the scheme.

In our relevance feedback scheme we obtain the most discriminative regions or keywords instead of the important keywords of a particular class of images. Given a set of retrieved images R and once the user marks all the relevant images P and the rest are the set of irrelevant images N we calculate the most discriminative keywords. This is done by defining a "Segment To Image" or "Keyword To Document" distance D_{si} which represents how close a segment or keywords is to an image. If SEG is the set of all the segments of P, then a pseudo-image of top num keywords whose cumulative distance to images in P is the least and the cumulative distance to images in N is the highest. This is quantitatively represented by a discriminability measure for each keyword in P calculated as discussed(Section 3.2.1). Hence we make a new pseudo-image with the most discriminative keywords of image class represented by R, allowing us to pick the representative segments dependent on the other classes in the database. This is done over many iterations.

As the relevance feedback scheme used tries to pick what makes each class unique, this uniqueness can be easily captured to aid in learning the concepts in the long term across multiple relevance feedback sessions. As the scheme is flexible, with slight modifications anything from spatial constraints to optimal segment grouping can be incorporated to achieve better results. Such a scheme will aid in distinguishing visually similar looking concepts. Once these keywords are obtained we make a pseudo-image or document out of the most

discriminative keywords. This pseudo-document is refined over further relevance feedback iterations. Hence in the end we have keywords or segments that are able to represent very specifically the concept they represent.

3.2.1 Algorithm

1. Obtain query image Q.
2. Obtain the image document (Collection of Keywords).
3. Image set R is retrieved from the database by the nearest neighbor retrieval algorithm.
4. Obtain feedback from user on R as P set of relevant image documents and N set of irrelevant image documents.
5. Calculate the most discriminative keywords from P and N
   - Calculate the Relevance score r_p among P for each keyword in P.
   - Calculate the Relevance score r_n among N for each keyword in P.
   - Obtain discriminative score d_r for all the keywords in P as $\frac{r_p}{r_n}$.
   - Sort the keywords in descending order of discriminative score d_s.
6. Pick top *num* keywords from the set of keywords such that all of them are mutually dissimilar by a minimum Hamming distance of x.
7. Collect these *num* keywords and construct a new pseudo image document and loop to step 2 until the user quits.

In the above algorithm we can see that only the keywords from P are used to estimate the new image or the pseudo image document of the concept at hand. Here we try to find the regions or keywords that are exclusive to a particular concept rather than keywords that are important to a particular concept. We also provide a threshold for discriminative capability

of two regions or keywords using x as the minimum hamming distance because of the need to eliminate redundant regions and at the same time allowing the pseudo image document to be as expressive as possible. Our algorithm can be termed as a hybrid bag of words approach as we are starting out with a generative model of what a particular concept is, then this model is modified by a discriminative learning model that refines the generative model to achieve discriminability from other concepts in the dataset.

3.3 Results and Analysis

We tested two methods or algorithms: discriminative relevance feedback (DRF) and relevance feedback based on region importance (Bayesian). First we converted all the images into pseudo images with the help of VTD. In the Bayesian or generative method we ignored the negative feedback images and boost the importance of words from the positive images. The methods were tested on two image sets $D1$ with 225 images and 7 categories and $D2$ with 1162 images and 15 categories. All the images in the two databases were taken from the corel image database [85]. $D1$ was used to confirm the methods ability to perform under well defined and visually disparate concepts and $D2$ was used to test the robustness of the schemes under conceptually different categories that are visually very similar.The retrieval set was of size 20 and this was used to calculate precision over a number of iterations.

$$Precision = \frac{NumberofRelevantImagesRetrieved}{SizeofRetrievedSet} \tag{3.4}$$

Here we find that our method DRF clearly outperforms the Bayesian probability based salient region retrieval method. We also observed that our scheme was able to distinguish very well between even hard to distinguish categories like “Surfers” and “Waves” or “Flowers” and “Roses”, and this is more prominent when one considers that the only features of significance here are 3 color features (Figure 3.3). Another important observation is that the DRF’s precision fluctuates, Bayesian however shows a stable increase in precision in the majority of the cases. Also as the number of distinct concepts grows DRF tends to browse through a wide variety of these classes based on the discriminability. So DRF requires some

Concept	Images	DRF	Bayesian
Bus	30	82	58
Car	34	98	62
Flower	30	63	42
Rocks	29	60	29
Sunset	35	92	56
Surfers	28	56	31
Train	30	74	54

Table 3.1: The above table contains 4 columns for dataset D1 as follows. Column 1 contains the class of images. Column 2 contains the number of images from each class Column 3 contains the precision(percentage) of Discriminative relevance feedback (DRF) Column 4 contains the precision(percentage) of a simple Bayesian relevance feedback approach (Bayesian)

iterations to get its bearing in the concept space. The performance of DRF on visually coherent concepts is outstanding. This can be clearly seen in the tables of $D1$ and $D2$ above. In both cases the user critiques on wether the given images are relevant or irrelevant. It was assumed the user critiques are consistent and deterministic regarding the relevance of an image to a concept.

3.4 Elastic Bucket Tries

Tries are ordered tree data structures that are used as associative retrieval entities that retrieve a record for the given string. Bucket tries and elastic bucket tries(EBT) [86] are variants that have the ability to pool various records with common key prefixes of a certain length into one bucket or block until the bucket overflows when more than N records are inserted into the bucket or block. Here, N is the maximum number of records allowed in a block. It is advisable to have each block of size 4096 bytes or one page for the x86 architecture

based systems. This ensures that any block is loaded into the main memory with the least amount of disk access which is the evident bottle neck. Here we have a special situation where all the possible strings or all the keywords of the document image are of the same length. So the maximum depth of the trie is $(m+1)$ where m is the length of all the strings. The root node is a null character that acts as an entry point to all the other strings. Each level also has an extra *Null* character node to accommodate for partial string matching in other than a prefix sense(for example using suffix trees for string matching).

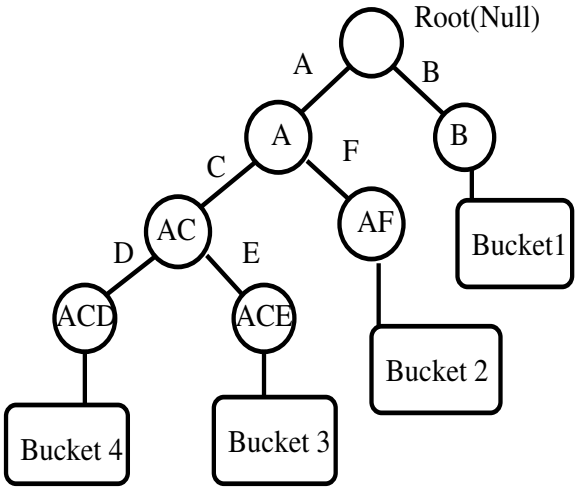

Figure 3.5: A Simple Bucket Trie

Buckets This data structure is designed to a cater to image databases of varying size from only a few hundred images to millions of images. Since this is for a dynamically scalable data structure and is designed to be deployed on anything from a workstation to a server it needs to allocate buckets or blocks on a demand basis. Though the entire trie can be populated with the leaves pointing to blocks right at the time of initialization, as the alphabet at each level is already known we do not do that because of efficiency and storage considerations. It is also due to the fact that a fully realized trie in the form of keys could be very sparsely populated as far as records go. This is the reason why new buckets are created or allocated only when existing buckets overflow.

Records Each record is a representative of a segment from an image in the database. It has the image name, handle or id. It has one string representative of the segment called the keyword of the segment. This keyword is used to decide which bucket this record falls into.

Insertion When a record r is to be inserted into a modified EBT (Elastic Bucket Trie) T the keyword of the record or the string representative of the segment within the record is obtained. From the root node r_n which is a null string the record descends through the trie until it reaches a bucket B at some level L such that $L \leq (m+1)$ where m is the size of all keywords or strings of the trie. Once the bucket is reached the record is inserted. If an overflow occurs the bucket is split into num_b new buckets where num_b is the size of the alphabet of the next character in the string. All the new buckets are placed one level lower than the original bucket after adding one character to the prefix of each bucket(this is where our modified EBT is different from an EBT, in a traditional EBT the buckets do not descend to reveal new leaf nodes). This splitting though costly is used to dynamically allocate space to the records on demand rather than allocating all the space at once, and this splitting only continues till the level $(m+1)$ where an overflow will result in another bucket or block being appended to the original bucket to contain the overflow. Hence buckets at the bottom level are not split. Hence limiting the total number of splits to a constant number.

The modified EBT does not have any deletion mechanism for the records. This is in harmony with the cheap secondary storage and dynamically increasing multimedia databases of today where deletion is treated as an unnecessary overhead. We hence avoid all the costs of merging buckets or blocks.

Retrieval Retrieval in our modified EBT is very efficient and is designed for and incorporated into a region based image retrieval framework in such a way that the trie need not change to accommodate for the change in the query due to relevance feedback. Hence retrieval is made independent of the dynamic nature of the interactions between the user and the system. When an image I is given as a query and I is a set of all the segments representing the image then for each segment S_i. $T = T \cup Retrieve(S_i, EBT)$ where $i = 1 \rightarrow n$.

Where n is the total number of segments in I and T is the set of all the records retrieved by querying for all the segments. The images whose handle occur the most are retrieved from storage in descending order ensuring that the image with the highest number of similar segments is first retrieved. Here partial segment matching is also taken care of due to the multiple levels at which buckets can occur. And every time there is relevance feedback from the user and the system is adapted a new pseudo image is given as a query and the same process continues over again.

3.4.1 Analysis

It can be shown that the modified EBT is far superior to standard spatial data structures for indexing and retrieval in a region based framework with a simple comparative scenario. We analyze the costs associated with insertion and retrieval in an R-tree and our modified EBT by comparing the worst case scenario complexities in both R-tree and the EBT. A record r is inserted into both the R-tree R_t and the EBT T_r. Then this record must be retrieved from the data structure. We calculate the standard costs of these operations while ignoring their variable costs. Lets assume the number of dimensions of the feature space is the same as the string length of all the keywords in the trie which is m, this is true because here each character represents one dimension. We also assume that an equally variable number of node n_i exist at every level i of the structures as one needs equal ground to compare both the data structures. Splitting is not accounted for while counting the cost.

R-tree Following is the cost associated with insertion and retrieval in an R-tree for a given image I.

1. Obtain record r_i from image. –constant time C
2. Start at root node of the R-tree.
    - Compare lower bound for m dimensions using floating point comparison n_i times. – cost of operation $m * n_i * C$

- Compare upper bound for m dimensions using floating point comparison n_i times. – cost of operation $m * n_i * C$

3. If the target block is at level l repeat above l times. - cost of operation $2(m * n_i * C) * l$
4. If target block reached insert record or retrieve block. – constant cost C
5. Repeat from 2 t times where t is the number of segments in I. – Total cost $2(m * n_i * C) * l * t$

EBT Following is the cost associated with insertion and retrieval in a modified EBT.

1. Obtain record r_i from image. –constant time C
2. Start at root node of the R-tree.

   - Compare single character using EXOR n_i times – cost of operation $n_i * C$

3. If the target block is at level l repeat above l times. - cost of operation $(n_i * C) * l$
4. If target block reached insert record or retrieve block. – constant cost C
5. Repeat from 2 t times where t is the number of segments in I. – Total cost $(n_i * C) * l * t$

From the Table 3.3 we see that the modified EBT clearly outperforms the R-tree by an order. That is the EBT performs one order better than the R-Tree. Such performance improvement was made possible due to the transformation of images into documents and segments into keywords. Hence by converting the spatial indexing and retrieval with relevance feedback into a problem that can be solved by EBT we have overcome inefficiencies. This data structure is both scalable and adaptable with minimum change to other modules in the system. Its inherent capability to merge well with relevance feedback of any type makes it an ideal data structure in dynamic CBIR systems.

3.5 Summary

We developed and discussed methods for efficient, scalable and adaptable image retrieval from large scale and dynamic databases. These methods include transformation of color images into documents using *Virtual Textual Description (VTD)* with the help of *grid based vector quantization* for CBIR. The usage of '*Discriminative Relevance Feedback* based on VTD improved the retrieval performance of the system by incorporating a learning element to better model the query. We also proposed and new indexing scheme for this CBIR system called an *Elastic Bucket Trie (EBT)* that had better performance characteristics than spatial indexing for CBIR. In recent years the advent of high performance photometric detectors and descriptors has opened up a new front for image retrieval based on the bag of words model. Even the Bag of Words model is more scalable than traditional CBIR it has some drawbacks when dealing with Dynamic Image Databases. The next chapter deals with vector quantization for dynamic image databases using bag of words model.

Concept	Images	DRF	Bayesian
Bus	91	88	63
Car	39	85	54
Flower	74	60	48
Cat	58	22	15
Sunset	135	85	40
Surfers	89	54	28
Train	82	66	52
Skiers	65	13	9
Sailboat	64	34	32
Tools	79	81	66
Waterfall	86	30	27
Wave	74	23	2
Bicycle art	78	54	52
Birds	82	34	26
Roses	101	87	56

Table 3.2: The above table contains 4 columns for dataset D2 as follows. Column 1 contains the class of images. Column 2 contains the number of images from each class Column 3 contains the precision of Discriminative relevance feedback (DRF) Column 4 contains the precision of a simple Bayesian relevance feedback approach (Bayesian)

Data Structure	R-Tree	EBT
Complexity	$2(m * n_i * C) * l * t$	$(n_i * C) * l * t$
Operations	Arithmetic	Logical
No. Of Splits	Indefinite	Fixed
RF Support	NO	YES
Efficiency	Low	High

Table 3.3: Comparison of R-tree and EBT

Figure 3.6: A Small Selection Of Retrieved Results after 5 iterations of Discriminative Relevance Feedback. The label below each row indicates the class of the image the user was looking for. One can see qualitatively the high precision of the system

Chapter 4

Incremental Vector Quantization For Dynamic Databases

4.1 Introduction

Dynamic databases are becoming ubiquitous with the emergence of large public and private visual databases that are growing at an unprecedented rate. We are interested in addressing the issue of efficient creation and maintenance of quality codebooks in large scale and highly dynamic image and video retrieval systems. That is, given a dataset of images to which new images are being constantly added, we want to update the codebook efficiently without effecting the retrieval performance. This implies that the required method must be incremental in nature and does not require the re-computation of the codebook when new images are added to the system.

In recent years, the bag of visual words model has been adapted to vision problems with great success [38, 39, 6, 40, 41, 42, 43, 44, 45, 46, 47, 87, 88, 89, 90, 91]. These approaches are shown to be well suited for tasks such as object categorization, object recognition, object retrieval and scene classification. The success of these approaches, in large part, is due to the model's ability to accommodate natural scene variance in the form of pose changes and occlusion. The quantization of a very high dimensional feature space (using an algorithm like

Kmeans)[48, 49] to build a compact codebook that encodes the similarity between descriptors, paves the way for efficient retrieval systems. The power of bag of words model to create efficient image and video retrieval systems has been explored by Sivic and Zisserman[6] as well as Nister and Stewenius[50]. The problem of building large scale image retrieval systems has also been looked into by Torralba *et al.*[1], though not utilizing the bag of words model. State of the art retrieval systems describe the images by sparse or dense descriptors and index them in an offline phase to build highly scalable retrieval systems. As the database dynamically evolves, the codebook is unable to accurately represent the underlying data. This necessitates the re-computation of the codebook at regular intervals. As the number of images and associated visual concepts increase, these computations become prohibitively expensive even for thousands of images on commodity hardware, often taking days or months to compute. To scale offline quantization to large scale databases the data is usually sampled and a small percentage of the images are used to compute the codebook. Even, in these cases traditional vector quantization methods cannot scale effectively[57]. This has resulted in the development and application of new methods like hierarchical and approximate Kmeans algorithms for building codebooks[50, 57]. Agglomerative and on-line quantization techniques have also been explored[56]. Quality of the model has also been optimized using discriminative visual codebooks[61]. However, any such database specific offline approaches are not extendable to the situations where the database is constantly evolving.

On the other hand, there are data independent quantizations like dividing the feature space into a regular grid. Even for a modest number of dimensions the vocabulary size of such schemes would be too large for effective use, making them impractical. Most feature spaces used in tandem with bag of words model are highly sparse and a grid based quantization algorithm will needlessly represent empty grid elements that are a majority [56, 78, 92]. Further, not all feature distributions correspond to images that are likely to appear in a given database. Usually only a minuscule fraction of possible images are ever seen. These considerations make grid quantization in higher dimensional spaces very expensive. A quantization algorithm that is fast, incremental and one that creates quality codebooks is needed.

In this thesis, we design and propose the Incremental Vector Quantization (IVQ) algorithm. It is designed to incrementally quantize the featurespace while meeting quality constraints. Further we compare the quality of codebooks created by our algorithm to that of K-means using the holiday [93] dataset and a generic dataset. We show that quantization is speeded up by a factor of 100 to a 1000. We also compare the incremental efficiency of IVQ vs K-means in creating and maintaining codebooks using a large dataset of more than a 100,000 images. Finally we show real world examples of the speed of IVQ's incremental quantization as well as it's retrieval performance as the database evolves.

K-means is the popular algorithm of choice for bag of words applications, owing to its simplicity and effectiveness. It is however, unsuitable for large scale, highly dynamic image retrieval. One of the major concerns is high computation time, which is aggravated by the inability to accelerate nearest neighbor search in high dimensional feature spaces. For example, quantizing a 2 hour movie consisting of 10,000 keyframes, each comprising an average of 100 descriptors of 128 dimensions for a 20,000 word vocabulary using kmeans would require around 72 hours. The time complexity of K-means is of the order $O(NKI)$, where N is number of feature vectors in the feature space and K is the number of means and I is the number of iterations. This time consuming quantization is viable for one time training on small datasets but cannot scale to very large datasets that are continuously growing. Running K-means every time new images are added and propagating the changes to the index and other systems downstream would be prohibitively costly. For instance, quantizing a second 2 hour movie one has already been quantized would require around 144 hours, or twice the time taken to quantize the first movie. K-means is also not aware of the perceptual nature of the underlying data. The means are drawn towards dense regions in the feature space. This, often leads to bias in the system towards high density regions present in feature space during offline quantization, resulting in an inconsistent codebook as new data is added to the system.

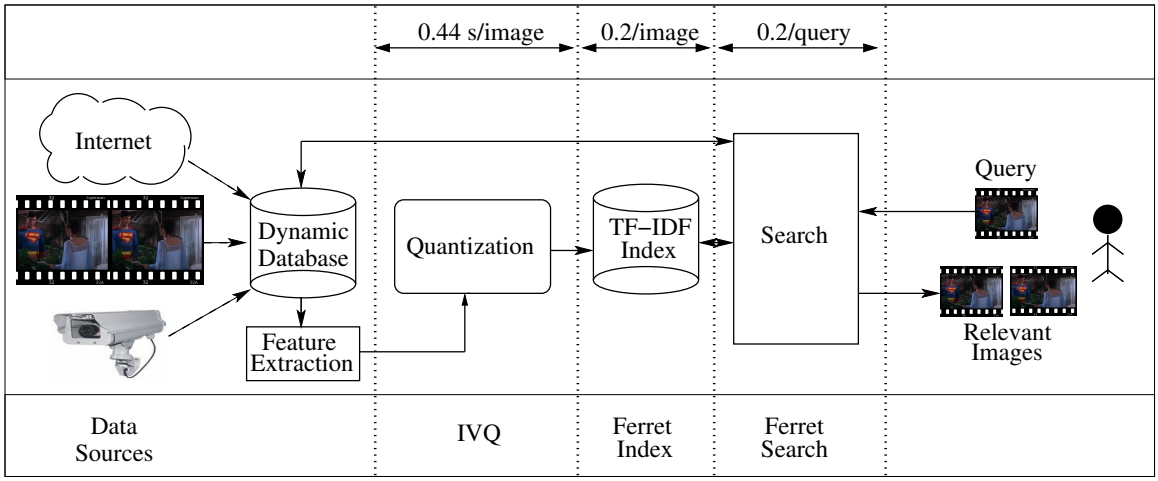

Figure 4.1: Image retrieval system for a dynamic database using IVQ for quantization and Ferret text search for indexing. The dynamic database is updated with images from data sources like the internet, movies and videos, sensors or camera feeds. The quantization time per new image is on average 0.44 seconds using IVQ. The indexing and retrieval speeds using the Ferret index is around 0.2 seconds per image. At such pace without considering feature extraction a one hour movie can be quantized and indexed in less than 50 minutes.

4.2 Vector Quantization

Kmeans is the most used algorithm for vector quantization in bag of words model. But as alluded to above kmeans is not ideal for use in evolving databases. Hierarchical kmeans is an obvious alternative to Kmeans that is much faster when compared to K-means. However, the partitioning imperfections at each level of the hierarchy add up, sometimes leading to a reduction in the quality of quantization[57]. Approximate K-means also has similar quality problems due to imperfections in distance calculation. As one tries to accelerate K-means, quality usually suffers. An alternative would be to use density based clustering algorithms, like DBSCAN [94]. DBSCAN is better suited for perceptual coding because unlike Kmeans it uses a global dissimilarity constant to cluster the data. Other alternatives like spectral clustering and Mean shift clustering cannot even be considered due to their prohibitive

computational cost.

Quantization of the feature space results in a definite loss of information, primarily in the form of perceptual loss and binning loss. After quantization, each distinct word in the code book is assumed independent of every other word in the codebook. Such an assumption allows for compact modeling of documents (word histograms) and for building learning systems that are independent of the underlying feature space unlike K-nearest neighbor methods[78]. This loss of descriptive and discriminative power through quantization is compensated for, to some degree, in most bag of words applications by the use of learning algorithms. Quality of quantization determines how well image retrieval system performs.

4.3 Incremental Vector Quantization

A good quantization algorithm, designed for retrieval in evolving image collections, should have the ability to adapt to new images as and when they appear. Working in an incremental manner, minimally effecting the current quantization and codebook would be desirable trait. The ability to build codebooks representing the underlying data without bias would be required. Other important factors to consider when trying improve the quality of the quantization for retrieval are, perceptual loss and binning loss. Mismanagement of these losses may eventually lead to under discretization or over discretization as explained below.

Perceptual Loss: Quantization in the context of a visual feature space is *perceptual coding*. Quantization here is an approximation or encoding of the underlying feature space. The similarity within feature space is encoded by dividing the feature space into independent perceptual bins. In these bins the more dissimilar a point is from the representative point (for example the mean in Kmeans) the greater the perceptual loss of the encoding with regards to that particular point. If $\mu_1, \mu_2, \ldots \mu_k$ are bin centers and C is a set of bins or concepts in the feature space, n is the number of feature vectors in the feature space and

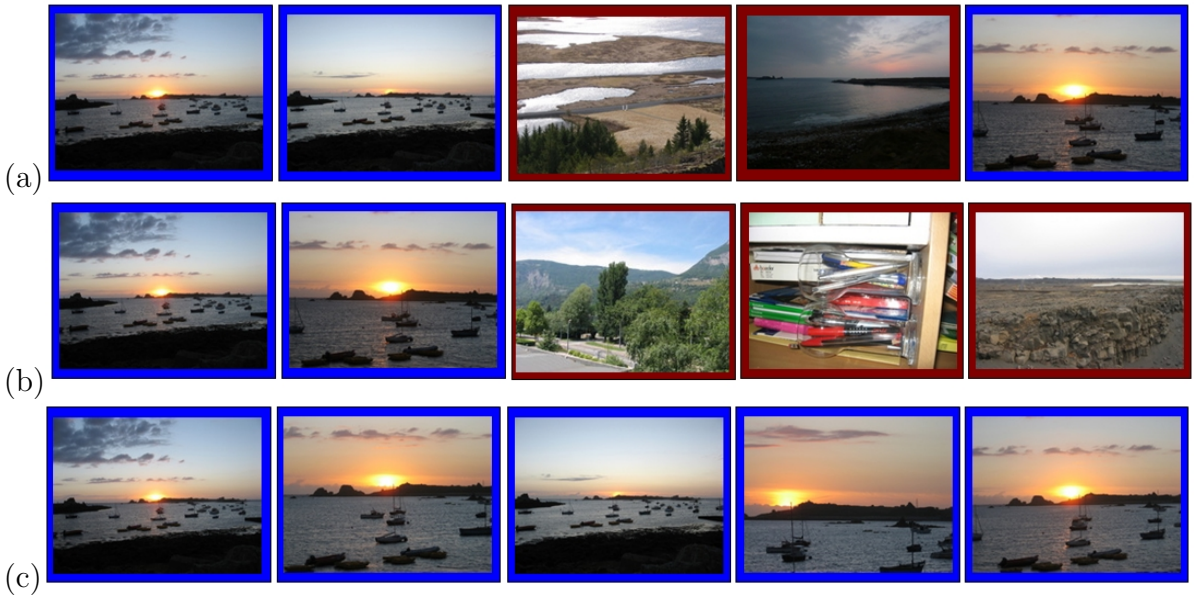

Figure 4.2: The image shows retrieval results for quantization under varying conditions. The blue boundary indicates accurate retrievals and the red boundary indicates an error in retrieval. *(a)* shows that when perceptual loss high it leads to underquantization and low precision. *(b)* shows that when binning loss is high it leads to overquantization and low recall. *(c)*shows high precision and recall for an optimal quantization

$dist(a, b)$ is a distance function then Perceptual Loss PL is given by Equation 4.1

$$PL = \sum_{i=1}^{k} \sum_{j=1}^{n} dist(p_j \in c_i, \mu^i) \tag{4.1}$$

Binning Loss: The assumption of independence of each bin in the feature space after quantization results in the loss of perceptual information between feature vectors belonging to different bins. If two feature vectors are perceptually very similar or have low dissimilarity but belong to different bins, the information of their similarity is lost in quantization[49]. If $x_1, x_2, \ldots x_n$ are feature vectors and C is the set of concepts or bins in the feature space then

Binning Loss BL is given by 4.2

$$BL = \sum_{a=1}^{n} \sum_{b=1}^{n} \{\delta : x_a \in c_z, x_b \notin c_z\} \frac{1}{dist(x_a, x_b)} \tag{4.2}$$

Under discretization and over discretization occur when these losses are mismanaged. Under discretization occurs when the number of bins the feature space is quantized into are not adequate to accurately represent the underlying data. The characteristics of this are high *perceptual loss* and high amount of *polysemy* in the words generated in the code book. High polysemy leads to poor precision as seen in Fig. 4.2(a). Over discretization, on the other hand, occurs when the number of bins are more than the number required to represent the underlying data. This leads to high *Binning Loss* and high amount of *synonymy* among the words generated in the code book leading to poor recall as seen in Fig. 4.2(b). As the feature space is more finely binned, the cardinality of the bins tends to follow the power law [56].

4.3.1 IVQ Algorithm

The design criteria for our algorithm are (i)the ability to limit perceptual loss, (ii)minimize binning loss and (iii)ability to create compact codebooks. We limit the perceptual loss with the help of a hard upper limit r(distance in the feature space). In order to meet the constraint of minimizing binloss we must accommodate all points possible in a bin that meet the constraint of perceptual loss being less than r. To minimize binloss, we allow multiple bin assignments for feature vectors where the perceptual loss is less than r. Hence, a single point in the feature space can generate multiple words in the index. We eliminate bins in the feature space and consequently words in the codebook that are not upto the quality desired, by using a density measure L. This ensures that outliers, and noise usually occurring in sparse regions of the feature space are discarded. Therefore, the maximum perceptual loss r and the minimum bin density L along with multiple bin assignments remain as the only parameters.

Algorithm: The algorithm has r and L as parameters and maintains a codebook C and

vector list V. When a new feature vector is introduced into the feature space, IVQ verifies to see wether the feature vector can belong to all bins within the codebook by calculating distance from the center of each bin and allocating the feature vector to any bin where the distance is less than r. If the feature vector is not assigned to any of the bins, we check the viability of it being used as a seed for creating a new bin. When the feature vector is used as the seed to create a new bin it must meet the criteria of the distance between bin center and feature vector being less than r. Further, from the list of feature vectors V the number of feature vectors that are closer to the bin center than r must be L for the bin to be added to the codebook. In other words the bin cardinality must be greater than L. If these conditions are not met, the new bin is not created and the codebook stays as it is. Irrespective of whether a new bin is created the feature vector is added to the vector list.

The bins do not move in the feature space and remain where they are. This makes the codebook robust to temporal factors, reducing index changes. This also ensures that all feature vectors in the feature space are represented with a perceptual loss of less than r. As mentioned earlier, each point can belong to multiple bins, and a point can belong multiple bins if and only if the distance between the bin centers is less than $2r$. This eliminates binning loss in dense areas where more descriptiveness is desired. Whenever a feature vector is associated with a new bin, only the index entry for that particular bin needs to be added to the index. Other index entries need not be modified. The membership criterion L ensures that the codebook is compact by eliminating outliers and noise from being included. Only bins whose cardinality is greater than a predetermined value L will be indexed, this is the membership criterion for being indexed. This keeps the codebook compact by not including outliers and random noise thereby improving the quality of the codebook. The choice of r needs to be made only once and does not need to be changed with the size of the image database as seen in Fig. 4.3. The factors that determine the choice or r are feature space and application. For example, a near duplicate image search engine using SIFT features would require a low r as it is essential to keep the perceptual loss to a minimum. On the other hand, a generic relevant image search of high level visual concepts would require higher value of r.

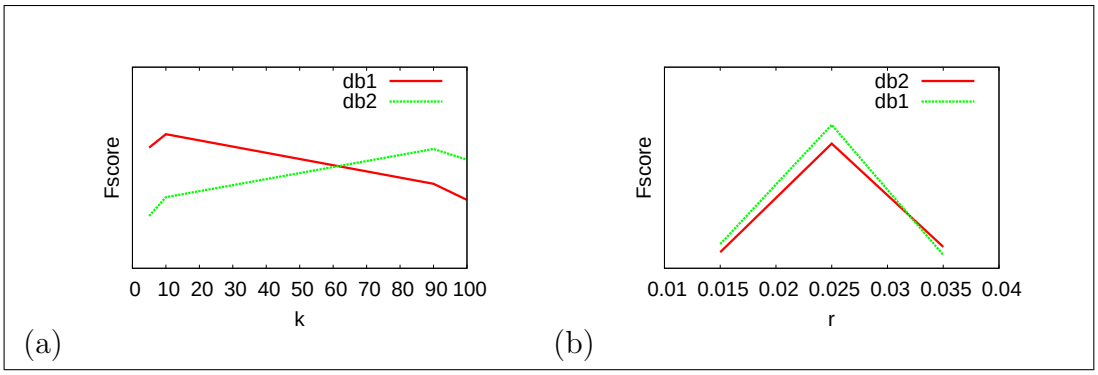

Figure 4.3: *(a)*The performance of Kmeans quantization is very sensitive to the parameter selection (K) when different feature distributions or databases are involved(Number of K's for one dataset doesn't perform in the same way for another dataset) *(b)*The performance of IVQ is not as sensitive to the parameter r even for different databases, this is due to the data independent nature of r, which is more feature space and application specific. The above experiment was carried out on two randomly generated point databases with different number of gaussians

Finally the quantization is accelerated by using locality sensitive hashing(LSH) for accessing the nearest bins in the feature space. Hence IVQ limits perceptual loss, minimizes binning loss and has the ability to control the compactness of codebooks.

4.3.2 Retrieval with IVQ

IVQ is an online quantization algorithm that runs as and when new data is introduced into the system. As seen in Fig. 4.1 the applications that require such quantization are varied and many. First, all visual data that is introduced to the retrieval system goes through a feature extraction phase. Usually high dimensional features like SIFT(128 dimensions) are extracted. These features are then fed into the IVQ quantizer, which then incrementally updates the current quantization to accommodate the new image. This incremental updation and the construction of the word histogram for the given image(single frame from a movie)

on average takes 0.44 seconds.

The next important phase in the process is the indexing of the image word histogram for retrieval. Usually this kind of indexing is done in memory. However, we chose on disk indexing using text search library called Ferret. This is due to its ability to scale indefinitely as well as being readily able to shard the index, unlike *in memory indexing* schemes. The indexing time(into and index of a million images) per image using ferret is less than 0.2 seconds. Once the images are indexed they are available to the user to be retrieved when relevant. The user retrieves relevant images from the index by sending a query to the search module. Relevant image retrieval from the index is consistently achieved under 0.2 seconds per query.

4.4 Experiments

4.4.1 Retrieval

A database comprising of a million images, each containing 10 descriptors in a two dimensional feature space is used. These images are categorized into 100 categories of 1000 images each. A concept in the feature space is described by a normal distribution where μ and σ are randomly picked, the feature space contains a 1000 such concepts. On this dataset we first ran the Kmeans algorithm with K = 1000 which according to the afore mentioned distribution of concepts in the dataset is viewed as an ideal initialization. The feature space is incrementally quantized and the time taken is measured. After quantization, the retrieval performance of the quantization for the bag of words model is calculated using precision and recall values . Then IVQ was used to quantize the feature space incrementally while measuring the time and subsequently the precision and recall. IVQ outperformed Kmeans in the amount of time taken to quantize the feature space. IVQ quantized the entire feature space in a single go in less than a second while Kmeans took over 16 minutes to do the same as seen in Fig. 4.4.

However one expects such a trade off to come at the price of reduced retrieval performance

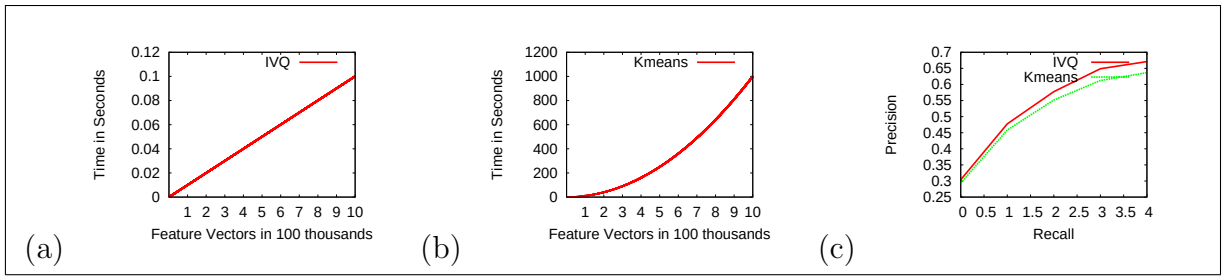

Figure 4.4: *(a)*Time taken by IVQ to quantize the feature space of different sizes, notice that the time scale is in 100ths of a second and IVQ takes nearly 0.1 seconds to quantize the entire feature space.*(b)*Time taken by Kmeans to quantize the feature space of different sizes, notice that the time scale is in seconds and it takes nearly 16 minutes to quantize the entire feature space. *(c)* Shows precision recall curves for both Kmeans and IVQ, IVQ has slightly better precision and recall characteristics than Kmeans. The precision and recall curves were calculated for all the classes and averaged to get average precision recall curves

on the part of IVQ considering the ideal initialization for Kmeans. Yet the results showed that IVQ outperformed Kmeans while creating a vocabulary that was only 10% larger, this performance improvement can be attributed to the reduction in binning loss due to multiple bin assignments in IVQ. The precision recall curve of IVQ shows better performance characteristics than that of Kmeans. IVQ's better performance was due to its soft bin assignment while the hard bin assignment of K-means and overlapping concepts in the feature space made it more susceptible to binning loss.

We then compared retrieval performance of IVQ to K-means on a standard dataset [93]. The dataset contains 500 image categories, each representing a different scene or object. The first image of each group is the query image and the relevant images are other images of the same group, in total the dataset contains 1491 images. We made extensive use of local detectors like Laplacian of Gaussian and the SIFT descriptors. Initially all the images from the dataset were downsampled to reduce the total number of descriptors, after which feature detection and feature extraction was done. Once the features were extracted the cumulative

feature space was vector quantized using both K-means and IVQ. After the quantization the vocabulary size of IVQ was truncated and retrieval performance of each quantization was measured by computing their respective mAP(Mean Average Precision) values. The results are shown in the Table 4.2. Tweaking the parameters of IVQ like r and L improved the mAP but this had to be also balanced with the vocabulary size. Very large vocabularies tend to effect the performance of the retrieval system.

4.4.2 Efficiency and Vocabulary

We use the ALOI image dataset[95] to compare the efficiency of both IVQ and K-means in an on-line quantization mode. We used a simple Locality Sensitive Hashing [80] scheme to accelerate and optimize IVQ without significant binning or perceptual loss. We quantize the entire dataset one image at a time simulating a dynamic image collection. We record the time it takes for each insertion. We also performed batch wise quantization using K-means and On-line K-means[1] where the positions of the existing means are re-calibrated in light of new data, rather than initializing them anew for every batch. In each session a batch of 100 images were added to the system using IVQ, K-means and On-line Kmeans. The recorded and projected times for quantization of some batches are shown in Fig. 4.5. IVQ outperforms Kmeans and On-line Kmeans by 4000 times and 400 times in case of the last batch.

Density Sensitivity: We plot the perceptual loss of feature vectors against their density. The density of the space around each feature vector is calculated by taking a window of size S and calculating the percentage of the points in the feature space that fall within this window. For this experiment we used custom image dataset and SIFT descriptors. We quantized the feature space using different K. We plot of perceptual loss of points in the feature space against the density of the points in the feature space, under different quantizations as seen in Fig. 4.5 *(a)*. Kmeans quantizations have a bias for high perceptual loss of low and medium

[1]There are other,efficient variants of Kmeans [49, 96]. However none of them are aimed at online quantization required for dynamic databases

(a)

Batch	IVQ	Kmeans	OnlineKmeans
1	42.7s	8.2e-06d	8.2e-05d
250	70.4s	0.1d	0.06d
500	132.0s	1.2d	0.25d
750	175.3s	4.3d	0.57d
1000	196.9s	10.1d	1.01d

(b)

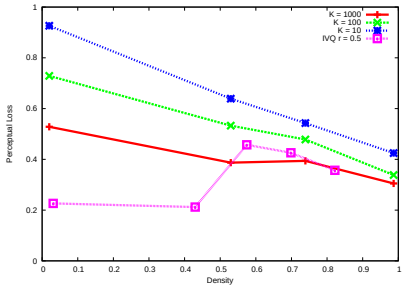

Figure 4.5: *(a)*Time taken by IVQ to incrementally quantize the feature space, notice that the time scale is in seconds and IVQ takes less than 200 seconds to quantize the last batch and time taken by Kmeans and Online Kmeans to incrementally quantize the feature space , notice that the time scale for both is in days and it takes 10 days and 1 day respectively to quantize the last batch.*(b)*Perceptual Loss with varying density in the feature space for Kmeans and IVQ, The graph shows large Perceptual loss bias in Kmeans towards feature vectors in sparse regions of the feature space

density regions, when compared to high density regions. IVQ on the other hand ensures that the perceptual loss is always below r.

Vocabulary Size: Here we intend to examine how the two parameters r and L effect the size and quality of the codebook. The size of the codebook is the number of bins in the feature space. The bigger the size of the codebook as seen in Fig. 4.6*(a)*, the greater the chance of over discretization and the smaller the size of the codebook the the greater the chance of under discretization. The quality of the codebook is calculated as the percentage of images retrieved. As r increases at a given L both the size of the vocabulary and the quality of the codebook increases because the probability of bin cardinality exceeding L increases . The increase in r improves the quality of the codebook due to the reduction of binning loss as similar feature vectors are binned together. At a given r as L increases the size of the codebook and the quality decreases as the number of qualified bins tend do decrease. This results in large parts of the feature space not being represented in the codebook there by

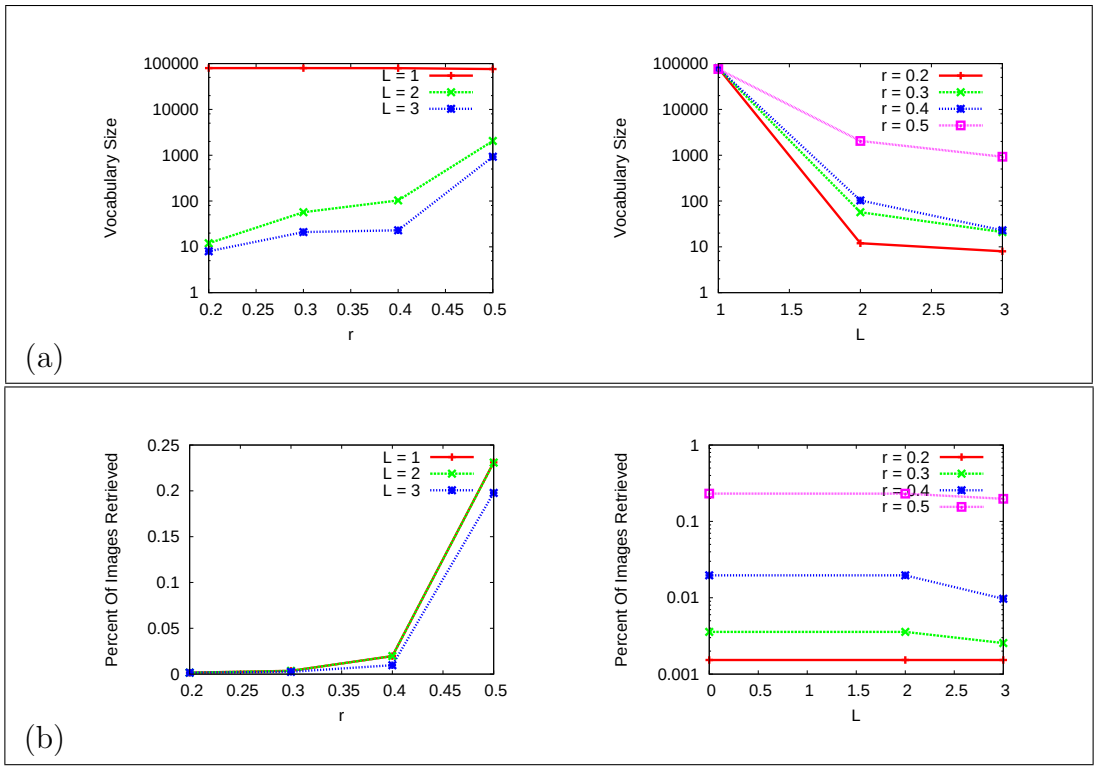

Figure 4.6: *(a)*Codebook size under varying r and varying L. *(b)* Percentage of total images retrieved under varying r and varying L

degrading it as seen in Fig. 4.6*(b)*.

4.4.3 Incremental Indexing and Retrieval of Videos

Here we demonstrate IVQ's ability to incrementally quantize at a fast pace while maintaining retrieval quality. We do this by indexing movies for retrieval. Each keyframe is processed by interest point detectors and subsequently these points are represented using the SIFT descriptor. Once each frame is processed the frame is inserted into the content server while the descriptors are processed by an incremental on-line vector quantization scheme that converts the descriptors into text based visual words. These collections of words or documents each having a relevant image in the content server are then indexed using a text search library

Ferret. When a new query frame is submitted to the system the ferret index is used to retrieve the results. The average quantization time for each image is of the order 0.44 seconds. Indexing into a TF-IDF on disk index is also extremely fast with an average rate of 5 images per second. Once the image is indexed using a search library like Ferret the searching and retrieval becomes extremely efficient returning results consistently below 0.2 seconds. IVQ without bottle necks of Disk I/O and Feature extraction can easily incrementally quantize more than 1 movie per hour(Key Frames).

Figure 4.7: The first row image shows retrieval results for a given query Only for "Father of the bride", while the second row shows the retrieval results for the same query after "Father of the bride Part II" is added to the system through incremental quantization. The Blue boundaries indicate relevant images and the red ones indicate irrelevant images. The incremental quantization increases the precision for the concept "house exterior" as the second movie is being added

To show the quality of incremental quantization we sequentially quantized movie franchises like "Father of the Bride" and "Superman". We examined to see if IVQ could improve the precision of similar concepts from different movies through incremental quantization. This would qualitatively validate the quality and adaptability of quantization through IVQ. As seen in both Fig. 4.7 and Fig. 4.8 the precision improves with incremental quantization of the sequels.

Figure 4.8: The first row image shows retrieval results for a given query Only for "Superman The Movie", while the second row shows the retrieval results for the same query after "Superman II" is added to the system through incremental quantization. The Blue boundaries indicate relevant images and the red ones indicate irrelevant images. The incremental quantization increases the precision for the concept "Superman Emblem" as the second movie is being added

4.5 Summary

We designed and presented a novel method called incremental vector quantization(IVQ) for use in image and video retrieval systems with dynamic databases. We demonstrated the quality of the codebooks as well as their adaptability and speed of creation by using various standard and generic datasets. We look at this work as a promising development towards building effective codebooks for large scale *user generated* databases where huge volumes of new visual data is continuously added. Semantic indexing for dynamic databases is also a hard problem to tackle, one has to content with large space complexities of traditional methods while being able to handle large and dynamic databases. The next chapter deals with semantic indexing in dynamic databases

```
def IVQ(V, r)
    foreach v_i in V do
       vectorAssigned = false
      foreach b_i in B do
          if dist(v_i,b_i) < r then
             AddToHash(H[b_i],v_i)
             vectorAssigned = true
             if BinCardinality(b_i) > L then
                UpdateIndex(b_i)
             end
          end
      end
       if vectorAssigned == false then
          CreateNewBin(v_i)
          InsertBinInList(b_{v_i})
          AddAllPointsWithin(r,b_{v_i})
          if BinCardinality(b_i) > L then
             UpdateIndex(b_i)
         end
       end
    end
end
```

Table 4.1: Sample implementation of *Incremental Vector Quantization*, This is only one way IVQ can be implemented

IVQ			Kmeans		
mAP	parameters	time	mAP	parameters	time
0.32	r=0.0514	783s	0.32	k=1000	5hrs
0.34	r=0.0823	717s	0.39	k=6000	25hrs
0.38	r=0.1153	656s	0.41	k=20000	82hrs

Table 4.2: Mean Average Precision Values with parameter values and time taken for both IVQ and Kmeans for the Holiday Dataset comprising of 1491 images. L=2 for IVQ. Notice that IVQ takes seconds to quantize the feature space while Kmeans takes hours to do the same

Chapter 5

Bipartite Graph Model For Semantic Indexing In Dynamic Databases

5.1 Introduction

We are interested in building scalable semantic indexing schemes for largescale, dynamic, image collections. That is, given a query, we want to retrieve the relevant images from a constantly changing database that could range in size from millions to billions of images. This implies the presence of millions of concepts and subconcepts, over which the system is required to perform efficient retrieval of relevant images without any *apriori* knowledge of the concepts present in the data. Any solution to this problem must be computationally viable without sacrificing the quality of the retrieval.

In recent years, the bag of visual words model has been adapted to vision problems [38, 39, 6, 40, 41, 42, 43, 44, 45, 46, 47] with great success. These approaches are shown to be well suited for tasks such as object categorization, object recognition, object retrieval and scene classification. The power of bag of words model to create efficient image and video retrieval systems has been explored by Sivic and Zisserman[6]. The success of bag of words model lies in its ability to quantize a very high dimensional feature space (using an algorithm like K-means)[48, 49] to build a compact codebook that encodes the similarity between descriptors

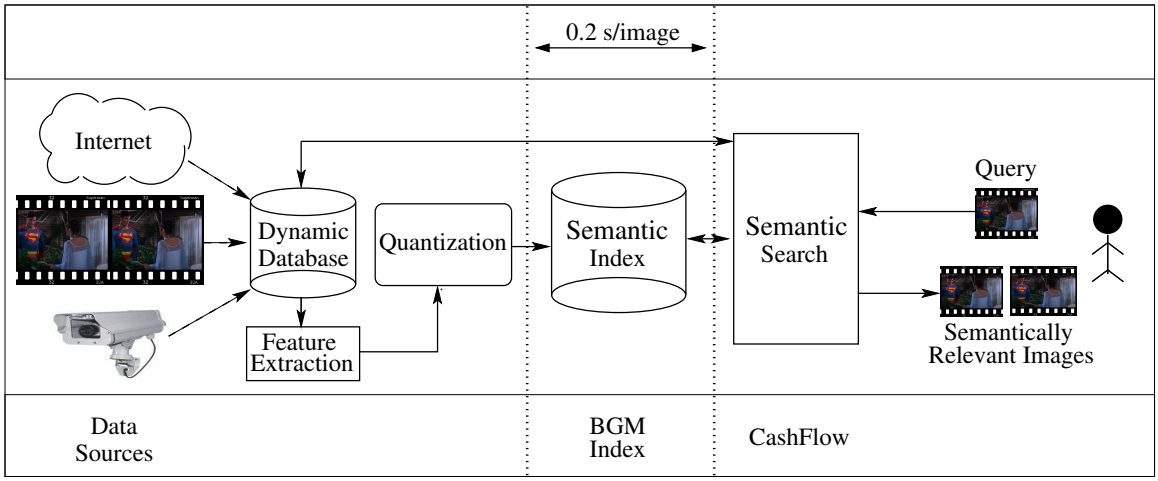

Figure 5.1: Image retrieval system for a dynamic database using BGM for indexing. The dynamic database is updated with images from data sources like the internet, movies and videos, sensors or camera feeds. The indexing time per new image is on average 0.2 seconds using a BGM index. At such pace without considering feature extraction and quantization a two hour movie with a 100,000 frames can be indexed in less than 80 seconds

and paves the way for efficient retrieval systems.

The quality of the retrieval is further enhanced with the help of semantic indexing techniques like Probabilistic Latent Semantic Analysis(pLSA)[63] and Latent Dirichlet Allocation (LDA)[64]. Semantic analysis of a document corpus can be viewed as unsupervised clustering of constituent words and documents around hidden or latent concepts in the corpus. Adaptation of PLSA and LDA to visual bag of words has provided promising results for static image databases[65, 66, 67, 68]. More recently semantic analysis is also being used in conjunction with spatial constraints for object segmentation [69, 67, 70], scene classification [47] and model learning [71, 72, 73].

Semantic indexing in a dynamic image collection poses a considerable challenge. As new images are constantly added to an image collection the semantic index is unable to accurately represent the changing database. This necessitates updation of the semantic model and indexing it at regular intervals which is time consuming and not scalable for large databases

with millions of latent concepts. As the number of images and associated concepts increases, these computations become expensive.

In this paper, we propose a Bipartite Graph Model (BGM) for semantic indexing that converts the vector space model into a bipartite graph which can be incrementally updated with *just in time* semantic indexing. We further propose a CashFlow Algorithm that traverses the BGM to retrieve relevant images at runtime. We compare the retrieval performance of BGM and pLSA using the holiday dataset[93]. We show that semantic indexing is speeded up by a factor of 100 when comparing BGM to pLSA. We, qualitatively and quantitatively compare the retrieval performance of BGM with naive retrieval(TF-IDF retrieval with no semantic indexing) and show its superiority. Finally we demonstrate the scalability, efficiency and real world retrieval capability of BGM in a near duplicate image retrieval application for more than a 1,000,000 images.

Traditional semantic indexing methods range from statistical methods like Latent Semantic Indexing [63] to probabilistic generative models like Probabilistic Latent Semantic Analysis(pLSA)[63] and Latent Dirichlet allocation(LDA)[64] and their incremental variants like incremental pLSA proposed by *Wu, et al.*[82]. One of the factors that make these methods challenging to adopt in a dynamic setting is their computational complexity. For instance, indexing a 2 hour movie comprising of 100,000 frames for near duplicate detection on off the shelf hardware(8gb memory) would require less than 80 seconds (Figure 5.7) for BGM while semantic indexing with LSI or pLSA is not possible due to their space complexity. The other factor that makes the adoption of these methods challenging is the selection of the number of global semantic topics. Even for incremental pLSA, selecting the number of latent topics in a changing database of millions of images is difficult. In such a setting where the database changes constantly a local concept threshold is appropriate due to its limited global impact on retrieval.

LSI uses Singular Value Decomposition(SVD) to factorize the term document matrix and create the semantic indexing model that identifies the relationships between the terms and concepts present in the database. LSI is based on the principle that words that occur in the

same contexts tend to have similar meanings. A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts. However, LSI is not scalable due to its resource intensive matrix operations and is sensitive to the number of dimensions over which SVD is carried out. pLSA and LDA have a more principled approach to semantic indexing with better grounding in statistics than LSI[63].

pLSA and **LDA** use Expectation Maximization(EM) to estimate a generative model to explain the observed data of words and documents, in context of the underlying latent data or concepts. These models have been applied successfully to various problems[65, 66, 67, 68, 46, 47]. However, these methods do not scale well for retrieval in large scale, highly dynamic image databases. LSI, pLSA and LDA are prohibitively costly to scale when dealing with very large datasets due to the resource intensive matrix computations needed. They can process a document corpus offline and cannot be updated in an incremental manner as is desirable in a dynamic environment where new data is constantly being added.

Incremental pLSA: there are many incremental variants of pLSA [82]. The performance of some of these methods both in terms of computation efficiency and retrieval performance are quite good. Yet they don't effectively address the issue of updating the number of global latent concepts as the database grows.

5.2 Bipartite Graph Model for Semantic Indexing

We suggest a semantic indexing model called *Bipartite Graph Model* (BGM) (Figure 5.2), that intuitively models and indexes the term document data in a scalable and incremental manner. BGM is designed to enhance the performance of large scale and highly dynamic image retrieval systems while at the same time providing an incremental concept centric indexing scheme with sublinear insertion and look-up performance.

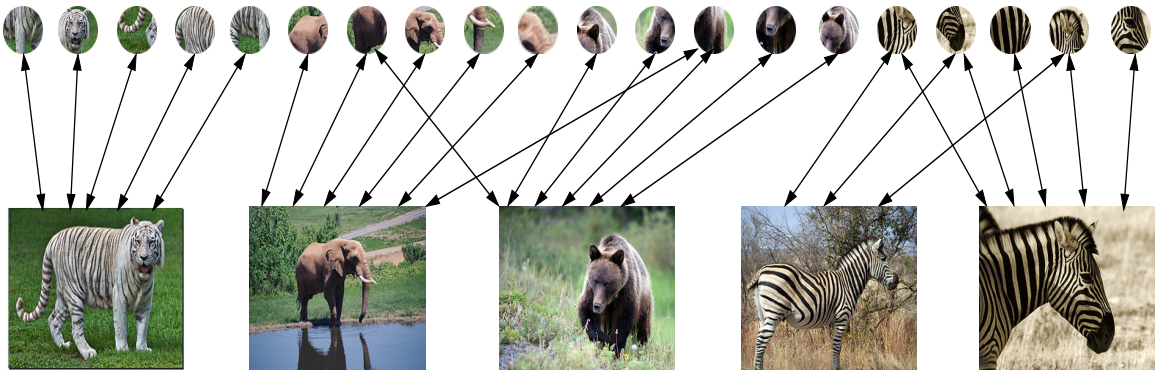

Figure 5.2: Graphical representation of the Bipartite Graph Model(BGM). The images or documents present in the corpus and are the collection of quantized feature vectors or visual words Present in the corpus. The edges connect visual words or image patches to images or documents in which they are present. You can notice that some patches are connected to more than one image, this is how co-occurrence is encoded in BGM. The greater the co-occurrence the more semantically relevant two images are. Here the two zebras are more semantically similar than the elephant and the bear

5.2.1 Semantic Similarity

Traditional semantic indexing methods calculate semantic similarity between two documents in a database by projecting the entire database into a latent concept space where the distance can be calculated. This projection encodes the co-occurrence data of terms and documents in the term document matrix. The intuition behind BGM and CashFlow algorithm is to retrieve semantically similar documents by using the afore mentioned co-occurrence information from the term document matrix rather than calculate the projection which is computationally expensive and sensitive to global parameter selection(number of latent concepts). Here we present a simple method for calculating semantic similarity. A document d_i from the global document collection D is assumed to have set of visual concepts C_i drawn from a global set of visual concepts C. The distribution B_{W_i} formed by the set of words W_i drawn from global set of words W within the document are generated by a mixture of these concepts P_{C_i}. Now

the basic retrieval problem is to retrieve documents from the database that resemble the mixture of concepts P_{C_i} in d_i as closely as possible. The intuition is that a document d_j with a word distribution B_{W_j} that is highly similar to B_{W_i} is likely to have been produced by mixture of concepts P_{C_j} that is highly similar to P_{C_i}. Here if f_{dis} is a function that calculates dissimilarity between two entities then

$$f^d_{dis}(d_i, d_j) \approx f^B_{dis}(d_i, d_j) \tag{5.1}$$

However a mixture of concepts P_{C_k} could be very similar to P_{C_i}, yet could generate B_{W_k}, where W_k is disjoint from W_i yet has a good amount of overlap with W_j. Then from Equation 5.1 it follows that

$$f^d_{dis}(d_i, d_k) \approx f^B_{dis}(d_i, d_j) + f^B_{dis}(d_j, d_k) \tag{5.2}$$

If there are m documents in the corpus the general form of Equation 5.2 could be written as

$$\sum_{x=1}^{m} f^B_{dis}(d_i, d_x) + \sum_{x=1}^{m} f^B_{dis}(d_x, d_k) + f^B_{dis}(d_i, d_k) \tag{5.3}$$

However this kind of approximation would require a transitive closure on the term document matrix, which would be prohibitively costly.

5.2.2 Term-Document Bipartite Graph

The central idea behind the bipartite graph model is that the vector space model is encoded as a bipartite graph of words and documents. The idea of converting the term document matrix into a bipartite graph is not novel and is used extensively in literature for a wide variety of tasks from semantic association of annotations to image retrieval. However, in BGM the edges are weighted with term frequencies of words in the documents as is relevant between each term and document. Each term is also associated with an inverse document frequency value.G is the bipartite graph such that

$$G = (W, D, E)$$

$$W = \{w_1, w_2 \ldots, w_n\}$$

$$D = \{d_1, d_2 \ldots, d_m\}$$

$$E = \{e_{w_1}^{d_1}, e_{w_7}^{d_2} \ldots, e_{w_n}^{d_m}\}$$

$$w_1 = IDF(w_1)$$

$$e_{w_1}^{d_1} = TF(w_1, d_1)$$

TF and IDF: In this model, the term and inverse document frequencies represent the word distribution within the document and in the corpus as a whole. These values together help in determining the importance of a word to a particular document. The term frequency representation of a document can be seen as a generative model of a document or histogram representation of a document and can be used to compute *KL-divergence* like dissimilarity. The IDF can be treated as a discriminative model of the document where the most discriminative words within a given document are given greater importance. The bipartite graph model combines both TF and IDF to be used in tandem like a hybrid generative-discriminative model. In essence the BGM encodes the co-occurrence data in the term document matrix without the need to project the database into a latent topic space.

5.2.3 Cash Flow Algorithm

We propose a *Cash Flow Algorithm* to find the semantically relevant documents in a document corpus in sublinear time using the *Bipartite Graph Model*. The main idea behind the cash flow algorithm is that, a query document(node) in the index is given cash to distribute among nodes that are relevant to it and they in turn propagate this cash distribution until the cash runs out. The higher the amount of cash flowing through a node the higher the relevance of the document(node) to the query. The cash flow algorithm is designed such that, at the time of querying, a single node or a set of nodes in the bipartite graph are infused with cash. If a node is a document node the cash is distributed among its edges in a quantity that is proportional to their flow capacity that is calculated by the normalized Term Frequency (TF) value. If the node is a word node it takes a portion of the cash it

receives(Table 5.2.3) as a *service fee* and distributes the rest like the document node based on the flow capacity of its edges. The service fee at each word node is calculated by using the Inverse Document Frequency (IDF) value of the word. Hence the cash is propagated through the system until a point when the cash flowing through a node is considered too little to justify the overhead, this is judged at each node with a *cutoff* value that is the least amount of cash needed for a node to forward the cash. At the start of every initialization each node that receives cash maintains a record of its cashflow. The end of a session is when there is no more cash flowing through the system due to the residual cashflow falling below the cutoff value. At this point the nodes that received cash are sorted based on the amount of cash that flowed through them. Total cashflow $Cash_{total}$for node N is

$$cash^N_{total} = cash^N_{previous} + cash^N_{current}$$

The two sorted node lists generated are the semantically most relevant documents and words to the given query according to the bipartite graph model.

The cutoff value along with service fee ensures that the cash flowing through the system decays over time and especially distance from point of initialization and that the algorithm eventually converges. Documents are inserted into the Bipartite Graph Model by creating a new document node and creating edges to the relevant words based on their term frequency (TF) values and updating the IDF values of the relevant word nodes. Insertions and deletions are linear in complexity to the number of words within a document. The system can be parallelized easily. The graph is thread safe allowing simultaneous reading and only requires conflict resolution when more than one thread is trying to update the IDF value of a word node. The cash flow algorithm essentially is a graph cut algorithm that divides the nodes in the bipartite graph into relevant and nonrelevant sets.

5.2.4 BGM for Retrieval

BGM is a an online semantic indexing data structure that inserts new data into itself as and when new data is presented to it. Unlike other semantic indexing methods there is no model

```
def cashFlow(G, N, cash)
    cashFlow[N] += cash
    if N.type == WORD
        cash = cash * N.idf
    end
    if cash < cutoff
        exit
    end
    foreach node in G.connectedNodes (N)
        cashFlow(G, node, cash * G.tf(N, node))
    end
end
```

Table 5.1: *Cash Flow Algorithm* for *Bipartite Graph Model*. Here both TF and IDF are normalized(less than 1)

updation required. As seen in Figure 5.1 the applications that require such quantization are varied and many. First, all visual data that is introduced to the retrieval system goes through a feature extraction phase. Usually high di- mensional features like SIFT(128 dimensions) are extracted. These features are then fed into the quantizer, which then incrementally updates the semantic index(BGM) to accommodate the new image. This incremental updation of the BGM index for the given image on average takes 0.2 seconds.

Usually this kind of indexing is done in memory. However, we chose on disk indexing using text search library called Ferret(stores TF-IDF values that are processed during search). This is due to its ability to scale indefinitely as well as being readily able to shard the index, unlike in memory indexing schemes. The indexing time per image using ferret is less than 0.2 seconds. Once the images are indexed they are available to the user to be retrieved when relevant. The user retrieves relevant images from the index by sending a query to the search

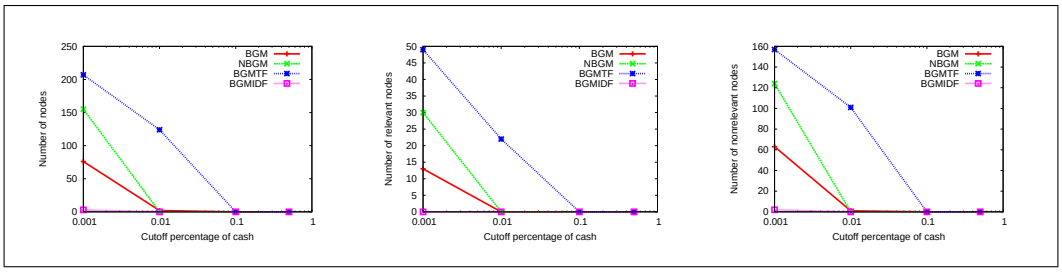

Figure 5.3: Number of nodes, relevant nodes and irrelevant nodes visited under varying cutoff

module. Relevant image retrieval from the index is consistently achieved under 2.5 seconds per query.

5.3 Experiments

5.3.1 Naive Retrieval vs BGM

First we study the retrieval performance of BGM and its variants when compared to simple retrieval without any semantic indexing involved. We make use of a Flickr sports dataset with 9 categories and a Flickr animal dataset with 5 categories both of which combined have more than nine thousand images. We extracted SIFT vectors from the images and quantize the feature space using Kmeans quantization with a vocabulary size of 10,000 and 5,000 respectively for sports and animals datasets and build a BGM as well as a simple inverted index for comparison of retrieval performance. We used four different variants of the cashflow algorithm to traverse the BGM. We measure the retrieval performance of an algorithm by calculating its F-score.

From Figure 5.4 we see that the performance of BGM algorithm compared to naive retrieval. BGM performs significantly better than simple retrieval which forms the baseline with an F-score of 0.05. As the number of cutoff nodes increases the performance increase of BGM begins to taper, this is due to the fall in recall as more and more noise from non

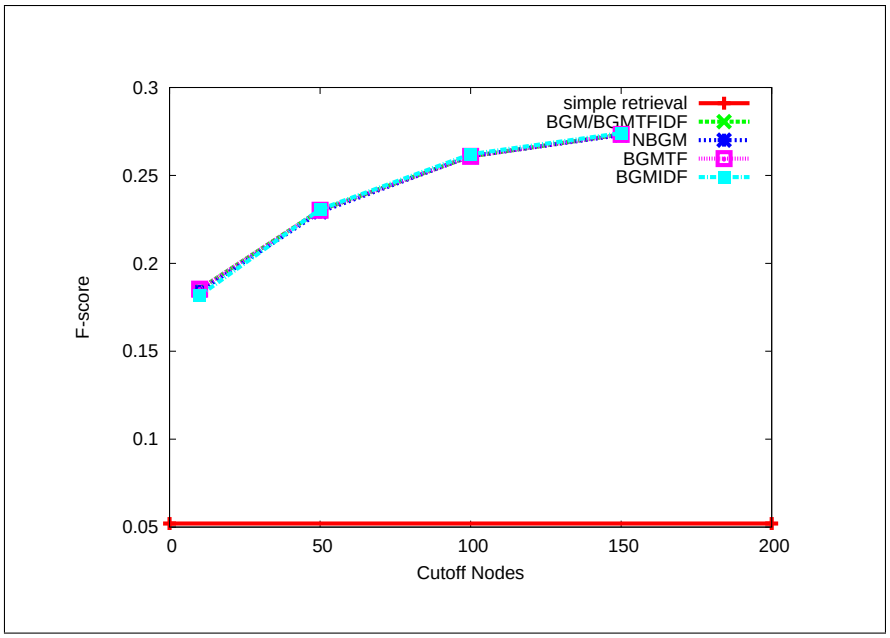

Figure 5.4: F-Score curves for BGM variants and Naive Retrieval, BGM clearly outperforms naive retrieval.

relevant image enters the system in successive iterations. Figure 5.4 and Figure 5.5 show how BGM is able to retrieve images that cannot be retrieved by simple retrieval.

Tweaking the edge flow capacities and node service fee leads to different variants of BGM. Naive BGM or NBGM does not have edge flow capacities. BGMTF has edge flow capacities and no service fee. BGMIDF has service fee and no edge flow capacities. BGMTFIDF or BGM has both edge flow capacities and service fee. Since the number of nodes traversed by the different cashflow algorithms for the same cutoff varies drastically as seen in Figure 5.3, we used number of nodes traversed as the cutoff condition to compare the different algorithms.

5.3.2 pLSA vs BGM

The objective of this experiment is to compare the offline retrieval performance of pLSA with that of the on-line retrieval performance of BGM. For this experiment we have used holiday dataset[93] , it contains 500 image groups, each representing a different scene or object. The first image of each group is the query image and the correct retrieval is the other images of the same group, in total the dataset contains 1491 images. We made extensive use of local detectors like Laplacian of Gaussian(LoG) and the SIFT descriptors[97]. Initially all the images from the dataset were downsampled to reduce number of interest points, after which feature detection and SIFT feature extraction was done. Once the features were extracted the cumulative feature space was vector quantized using K-means. With the aid of this quantization the images were converted into documents or collection of visual words.

For pLSA each image was represented as a histogram of visual words. Aggregating these histograms the term document matrix was represented by A of the order $M \times N$ where M is the vocabulary size and N is the document corpus size. Here $A(w_i, d_j)$ is the term frequency of the term w_i pertaining to the document d_j. This term document matrix is used for pLSA where a hidden aspect variable Z_k is associated with each occurrence of a visual word w_i in an image d_j. The conditional probability $P(w|d)$ is

$$P(w_i|d_j) = \sum_{k=1}^{K} P(z_k|d_j)P(w_i|z_k)$$

where $P(z_k|d_j)$ is the probability of the topic z_k occurring in the document d_j and $P(w_i|z_k)$ is the probability of the word w_i occurring in a particular topic z_k. The pLSA(EM) model generates $P(z)$,$P(z|d)$,$P(w|z)$. The EM model was initialized with latent 500 topics which is similar to the number of categories in the dataset. Once the model converges all the topic probabilities for all the documents in the corpus are generated. For retrieval the Euclidean distance of the documents over topic probabilities was used to retrieve the 10 most similar images.

For BGM each image was represented as a document comprising of visual words. Then a term document matrix was created where each row representing m_i representing the term

frequencies of the relevant document was normalized. Then all the terms in the matrix were updated with their inverse document frequency values. This term-document matrix was then converted into a bipartite graph between the set of terms and documents as described by the BGM model. For each of the 500 query images the cash flow algorithm was used over this graph to retrieve the 10 most similar images.

Retrieval results for the both BGM and pLSA were aggregated and the evaluation code provided for the holiday dataset was used to calculate the Mean Average Precision(mAP) in both cases. The mAP results show that BGM performs very comparably to pLSA. However, when one looks at the memory usage and time taken for creating the semantic indexes(training) in both cases one can clearly notice the difference. Here, BGM outperforms pLSA by the order of 100. However, the real advantage of BGM is noticed when adding another image to the index only takes a few milliseconds while for pLSA the computation of the entire semantic index needs to be done again incurring high time and memory costs.

Model	mAP	time	space
Probabilistic LSA	0.642	5473s	3267Mb
BGM + CashFlow	0.594	42s	57Mb

Table 5.2: Mean Average Precision for both BGM and pLSA for the holiday dataset, along with time taken to perform semantic indexing and memory space used during indexing

5.3.3 Retrieval Performance

Text retrieval systems and search engines have become a commodity with large numbers of *off the shelf* and opensource systems available. These can be easily scaled to handle billions of documents and millions of queries with ease. This kind of scalability has always been a challenge for image retrieval systems, but bag of words model enables one to build such image retrieval systems[57]. Can this gap be eliminated by using text retrieval systems for image retrieval. We explore this possibility by building a full scale image retrieval system

by using a text search engine. Accomplishing this means access to proven technology from basic text indexing schemes to advanced crawling, index sharding, index optimization and ranking algorithms and implementations. In order to accomplish this we must first convert image documents to text documents. We achieve this using a simple hash function that converts codebook bins into text strings and subsequently images into text documents. We build our image retrieval engine using the Ferret search library which is a ruby port of the *Apache Lucene* project.

BGM was used in conjunction with the Ferret index(Which indexed and stored the relevant TF and IDF values) to achieve semantic indexing. The space complexity of PLSA is of the order$O(TN_z)$ where N_z is the number of nonzero elements in the document term matrix and T is the number of topics. Thus 10 million non zero elements in the document term matrix would necessitate a memory requirement of no less than 10GB. At this scale pLSA takes a few hours to compute. Both space and time complexity of PLSA make it an impractical choice in a dynamic environment. BGM, on the other hand is a data structure that is resident on disk, which makes updating BGM highly efficient due to absence of any significant computation. In order to put BGM and the Ferret index through their paces we adjusted vector quantization parameters to create a large and descriptive vocabulary of more than 6 million words. Each image in the dataset on average has 110 visual words across 100,000 images. The average time taken to insert an image into BGM is of the order 0.0134 seconds the same as the time it takes for an image to be inserted into the ferret index. The average response time for a query for Ferret is 0.29 seconds while the average response time for a BGM query is 2.42 seconds. The discrepancy in response times can be attributed to the multiple levels of graph traversal by the Cashflow algorithm in case of BGM. Even though BGM improves retrieval performance (Figure 5.3 and Figure 5.4) by a large margin, the discrepancy in retrieval time is very low as clearly seen in Figure 5.6. The response time of BGM and Ferret can be improved by sharding the index across multiple machines while at the same time providing high scalability.

5.4 Near Duplicate Detection

Near duplicate detection in videos and images involves finding images that are almost similar to the query image with only slight changes, like successive frames in a video. It is a challenging problem for bag of words based image retrieval methods. Some of the interesting problems that need to be tackled involve scalable and efficient vector quantization and semantic learning. Here we discuss the application of BGM over a large dataset.

The data for the application comes from frames of various motion pictures. Each frame is processed by interest point detectors and subsequently these points are represented using the SIFT descriptor. Once each frame is processed the frame is inserted into the content server while the descriptors are processed by an incremental on-line vector quantization scheme that converts the descriptors into text based visual words. These collections of words or documents each having a relevant image in the content server are then indexed using a text search library Ferret. When a new query frame is submitted to the system first the new document is treated as the node initiating the flow in BGM and the TF-IDF index is used to boost the terms and submitted as a query to the Ferret index. This same process continues at every subsequent document node until sufficient number of duplicates with the relevant scores are retrieved. Even with such massive amounts of data like movie frames the system is able to scale very effectively. The indexing time for 1000 images after nearly a million images are already present in the index is of the order of 100s of seconds. Similarly the retrieval time is on average less than 2 seconds over the entire index(Figure 5.6). BGM significantly outperforms naive retrieval by discovering and retrieving a varied range of near duplicate frames than Naive retrieval(Figure 5.8 and Figure 5.9).

5.5 Summary

We proposed a method and a datastructure that tackle representation of the term document matrix and on-line semantic indexing where the database changes. We introduced a bipartite graph model (BGM) which is a scalable datastructure that aids in on-line semantic indexing,

which can be incrementally updated. We also introduced a cash flow algorithm that works on the BGM to retrieve semantically relevant images from the database. We examined the properties of both BGM and cash flow algorithm through a series of experiments. Finally, we demonstrated how they can be effectively implemented to build large scale image retrieval systems in an incremental manner. In the final chapter, the thesis is summarized and concluded with a look at what the future directions of the work could be.

Figure 5.5: Relevant images retrieved(a) with an inverted index of bag of words model for a zebra image query and additional relevant images(b) retrieved by BGM for the same query. BGM significantly outperforms Naive retrieval

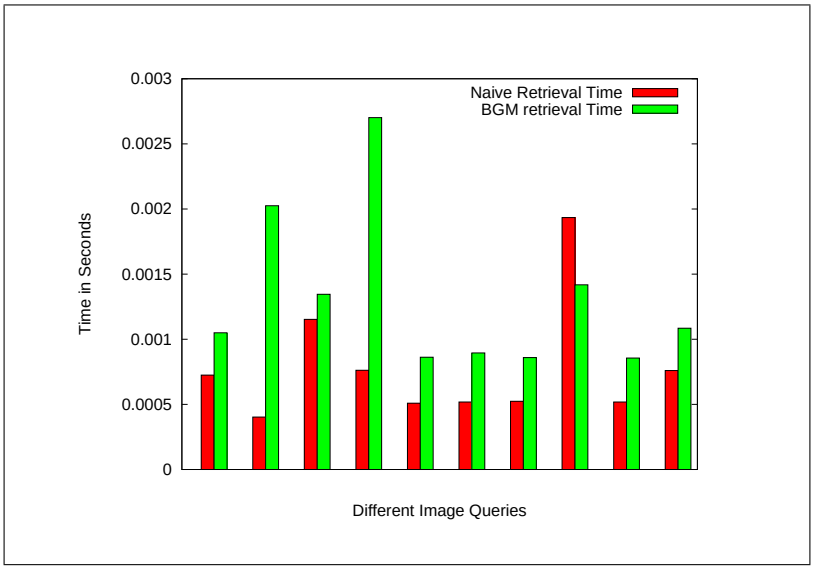

Figure 5.6: Query response times across 10 queries for Ferret and BGM, One can clearly notice that the retrieval times are very comparable to one another

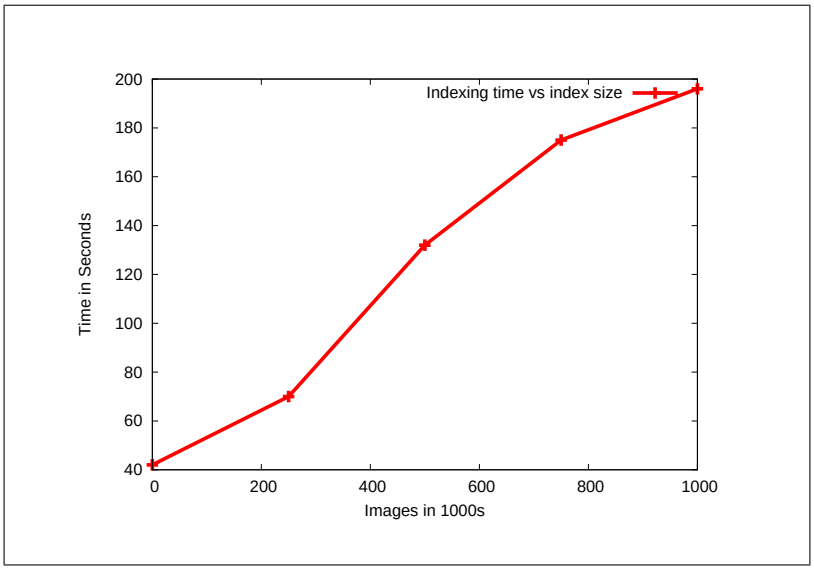

Figure 5.7: indexing time vs size of the index in 1000's of images, even at a million images the time taken for inserting a batch into the semantic index is under 200 seconds

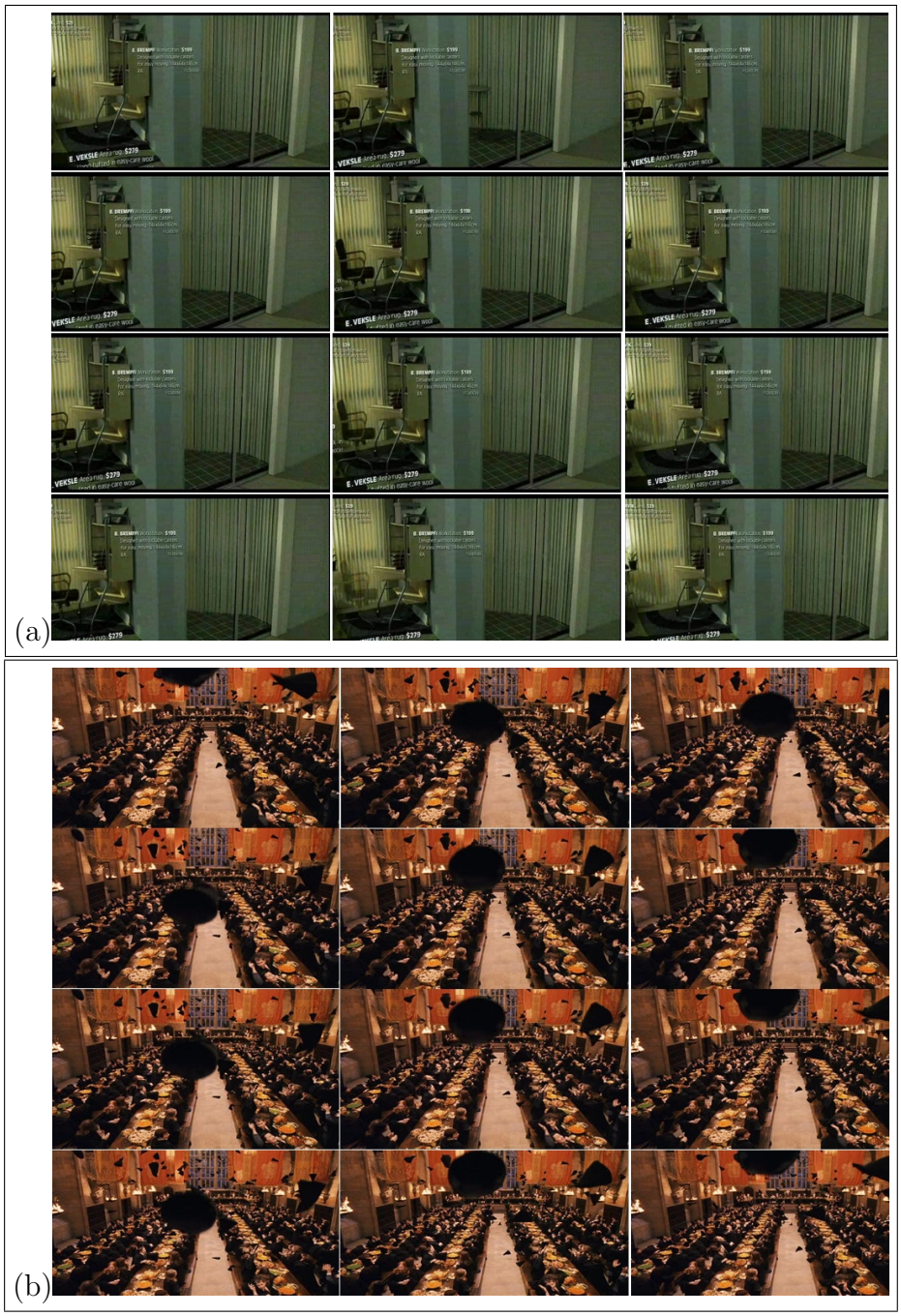

Figure 5.8: Near duplicates detected for a frame in the movie Fight Club(a) and Harry Potter(b) respectively, one can notice that the frames only differ slightly from each other.

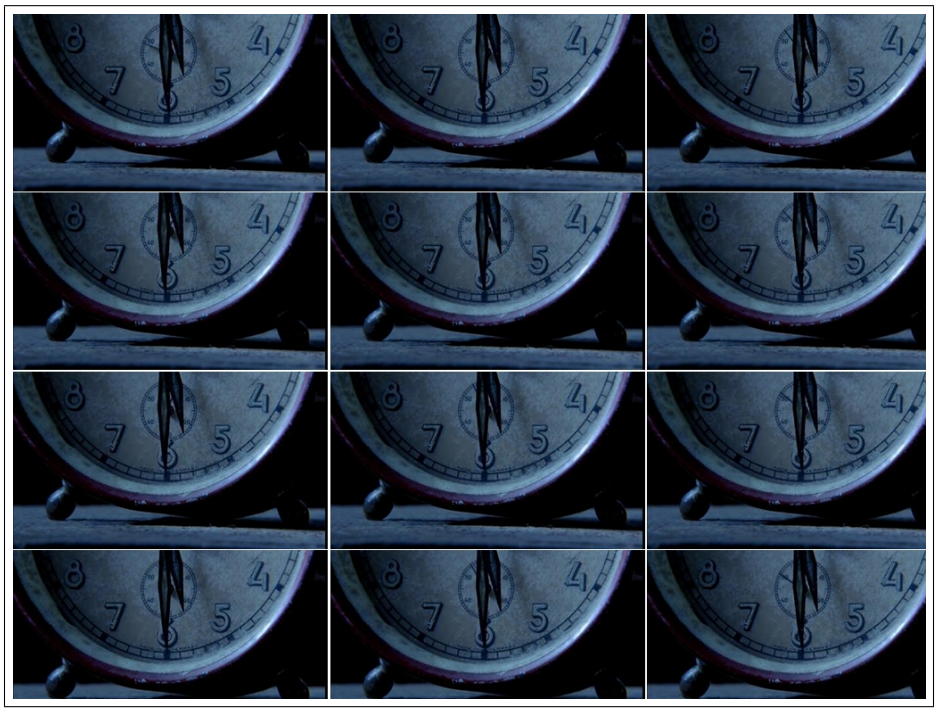

Figure 5.9: Near duplicates detected for a frame from the movie The Fastest Indian, BGM is able to retrieve a larger number of near duplicate frames than Naive retrieval.

Chapter 6

Conclusion

The growth of multimedia content and the maturing social web are leading to emergent needs from multimedia content. Our lives are becoming more and more multimedia driven from IPTV to the ubiquitous *always on* devices like camera cell phones. The need to organize, streamline and present the relevant information from the mountains of data both multimedia and otherwise is more pertinent than ever. Towards this end one must first eliminate roadblocks that stand in the way of deploying innovative solutions like CBIR and multimedia retrieval. We have made contributions in addressing the roadblocks of scalability over large amounts of data(large scale databases) as well as adaptability over ever changing data (dynamic databases).

In case of CBIR we have studies the nature of traditional systems and their shortcomings when it comes to real world deployability. We have proposed methods that address the scalability concerns, without being agnostic to retrival performance of the system through relevance feedback. We have also proposed a datastructure and relevance feedback scheme to improve the scalability and retrieval performance of the system.

Though bag of words model based image retrieval has better scalability charatechteristics than traditional CBIR we have found that contemporary bag of words methods do not adapt well to dynamic image databases. Vector quantization is one such bottle neck, which we propose to mitigate with incremental vector quantization. We show the performance

charecteristics of IVQ when compared with Kmeans and demonstrate how effective image search engines can be built using off the shelf text search libraries.

Another important problem to adress with respect to Bag of words based image retrieval is semantic indexing. In our survey of traditional semantic indexing methods like LSI, PLSA, LDA and Incremental PLSA we have found various shortcomings. We propose a new just in time semantic indexing method which works on the Bipartite Graph Model datastructure constructed from terms and documents.

6.1 Future Work

The goal of effective information retrieval is always a moving target and work is constantly needed to keep up with it. Image retrieval is no exception. Following are some challenging and promising areas where the work in this thesis leads to.

Multimodal Retrieval BGM and Cash Flow algorithm can be readily used with image as well as textual cues. However the prominence of Text vs Image features for a given query must be adjuged at runtime. Should text search be used to retrieve images and image search to filter them or the other way around are some the questions that need to be asked and explored.

Multiple Vocabularies As there are innumberable subjective visual concepts each with a wide range of visual representation, effective large scale image understanding and retrieval would require many specialized vocabularies and not just one. For example a vocabulary for understanding and representing faces, a vocabulary for vehicles, a vocabulary for backgrounds, etc. Feature engineering, feature fusion and feature diversity are valid concerns to be addressed here.

Appendix A

List Of Publications

1. Suman Karthik, C.V. Jawahar, "Analysis of Relevance Feedback in Content Based Image Retrieval", Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV),1-6, 2006, Singapore.

2. Suman Karthik, C.V. Jawahar, Virtual Textual Representation for Efficient Image Retrieval. Proceedings of the 3rd International Conference on Visual Information Engineering(VIE), 26-28 September 2006 in Bangalore, India.

3. Suman Karthik, C.V. Jawahar, "Efficient Region Based Indexing and Retrieval for Images with Elastic Bucket Tries," icpr, vol. 4, pp.169-172, 18th International Conference on Pattern Recognition (ICPR'06) Volume 4, 2006

4. Suman Karthik, Chandrika Pulla and C.V. Jawahar, "Incremental On-line semantic Indexing for Image Retrieval in Dynamic Databases,"4th International Workshop on Semantic Learning, CVPR 2009

Bibliography

[1] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *CVPR*, pp. 1–8, 2008.

[2] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Past, present, and future," in *International Symposium on Multimedia Information Processing*, 1997.

[3] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Third International Conference on Visual Information Systems*, Springer, 1999.

[4] Y. Du and J. Z. Wang, "A scalable integrated region-based image retrieval system.," in *Proceedings Of International Conference of Image Processing (1)*, pp. 22–25, 2001.

[5] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[6] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, vol. 2, pp. 1470–1477, oct 2003.

[7] H. Tamura and N. Yokoya, "Image database systems: A survey," *Patt. Recog.*, vol. 17, no. 1, pp. 29–43, 1984.

[8] A. Hsu and S. Chang, "Image information systems: Where do we go from here?," in *IEEE Trans. on Knowledge and Data Eng.*, vol. 4, pp. 431–442, Oct 1992.

[9] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[10] O. Marques, L. M. Mayron, G. B. Borba, and H. R. Gamba, "On the potential of incorporating knowledge of human visual attention into cbir systems," in *ICME*, pp. 773–776, 2006.

[11] S. Karthik and C. V. Jawahar, "Analysis of relevance feedback in content based image retrieval," in *ICARCV*, pp. 1–6, 2006.

[12] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts.," in *Proceedings of the International Conference on Computer Vision*, pp. 1331–1338, 2005.

[13] A. Bar-Hillel, T. Hertz, and D. Weinshall, "Efficient learning of relational object class models.," in *Proceedings of the International Conference on Computer Vision*, pp. 1762–1769, 2005.

[14] M. Fritz, B. Leibe, B. Caputo, and B. Schiele, "Integrating representative and discriminative models for object category detection.," in *Proceedings of the International Conference on Computer Vision*, pp. 1363–1370, 2005.

[15] J. M. Winn, A. Criminisi, and T. P. Minka, "Object categorization by learned universal visual dictionary.," in *Proceedings of the International Conference on Computer Vision*, pp. 1800–1807, 2005.

[16] Y. Rui, T. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in mars," pp. II: 815–818, 1997.

[17] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, April 2008.

[18] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984* (B. Yormark, ed.), pp. 47–57, ACM Press, 1984.

[19] S. Berchtold, D. A. Keim, and H.-P. Kriegel, "The x-tree : An index structure for high-dimensional data," in *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pp. 28–39, Morgan Kaufmann, 1996.

[20] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r*-tree: An efficient and robust access method for points and rectangles," in *Proceedings of the 1990 ACM SIGMOD*, pp. 322–331, ACM Press, 1990.

[21] K.-I. Lin, H. V. Jagadish, and C. Faloutsos, "The tv-tree: An index structure for high-dimensional data," *VLDB J.*, vol. 3, no. 4, pp. 517–542, 1994.

[22] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation by image exploration," in *ECCV*, vol. I, pp. 40–54, 2004.

[23] D. Lowe, "Distinctive image features from scale-invariant key-points," *Intl. Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[24] A. S. Salgian, "Using multiple patches for 3d object recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–6, 2007.

[25] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pp. 1150–1157, 1999.

[26] G. Dorkó and C. Schmid, "Selection of scale-invariant parts for object class recognition," in *International Conference on Computer Vision*, vol. 1, pp. 634–640, 2003.

[27] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 264–271, June 2003.

[28] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pp. 525–531, 2001.

[29] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 530–534, May 1997.

[30] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Lausanne, Switzerland), pp. 226–231, October 2002.

[31] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

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

[33] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. 506–513, 2004.

[34] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[35] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded-up robust features," in *9th European Conference on Computer Vision*, (Graz, Austria).

[36] M. S. Sarfraz and O. Hellwich, "Head pose estimation in face recognition across pose scenarios," in *VISAPP (1)*, pp. 235–242, 2008.

[37] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 370–377, 2005.

[38] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision*, pp. 59–74, 2004.

[39] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," vol. 1, pp. 370–377, 2005.

[40] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *ECCV*, pp. IV: 490–503, 2006.

[41] D. Larlus and F. Jurie, "Latent mixture vocabularies for object categorization and segmentation," *Journal of Image and Vision Comput.*, vol. 27, no. 5, pp. 523–534, 2009.

[42] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *ECCV*, pp. 464–475, 2006.

[43] F. Schroff, A. Criminisi, and A. Zisserman, "Single-Histogram Class Models for Image Segmentation," in *ICVGIP*, 2006.

[44] R. Fergus, *Visual Object Category Recognition*. PhD thesis, University of Oxford, Dec. 2005.

[45] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in *ICCV*, 2005.

[46] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *ECCV*, 2006.

[47] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification using a hybrid generative/discriminative approach," *PAMI*, vol. 30, no. 4, 2008.

[48] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and tf-idf weighting," in *BMVC*, 2008.

[49] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.

[50] D. Nistr and H. Stewnius, "Scalable recognition with a vocabulary tree," in *CVPR*, pp. 2161–2168, 2006.

[51] H. Jegou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," in *CVPR*, jun 2007.

[52] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *ICCV*, 2007.

[53] W. Zhong, G. Altun, R. Harrison, P. C. Tai, and Y. Pan, "Improved k-means clustering algorithm for exploring local protein sequence motifs representing common structural property," *NanoBioscience, IEEE Transactions on*, vol. 4, no. 3, pp. 255–265, Sept. 2005.

[54] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering," in *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, (San Francisco, CA, USA), pp. 91–99, Morgan Kaufmann Publishers Inc., 1998.

[55] K. Wagsta, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *Proceedings of 18th International Conference on Machine Learning (ICML-01)*, pp. 577–584, 2001.

[56] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *ICCV*, pp. 604–610, 2005.

[57] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[58] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*, June 2008.

[59] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," in *BMVC*, 2008.

[60] T. Yeh, J. Lee, and T. Darrell, "Adaptive vocabulary forests br dynamic indexing and category learning," in *ICCV*, pp. 1–8, 2007.

[61] F. Moosmann, B. Triggs, and F. Jurie, "Randomized clustering forests for building fast and discriminative visual vocabularies," in *NIPS*, nov 2006.

[62] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," pp. 1–8, 2008.

[63] T. Hofmann, "Probabilistic Latent Semantic Indexing," in *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, (Berkeley, California), pp. 50–57, August 1999.

[64] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[65] F.-F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, pp. 524–531, 2005.

[66] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. V. Gool, "Modeling scenes with local descriptors and latent aspects," in *ICCV*, pp. 883–890, 2005.

[67] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *CVPR*, 2006.

[68] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in *ICCV*, 2005.

[69] L. Cao and L. Fei-Fei, "Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes," in *ICCV*, pp. 1–8, 2007.

[70] X. Wang and E. Grimson, "Spatial latent dirichlet allocation," in *NIPS*, vol. 20, 2007.

[71] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros, "Unsupervised discovery of visual object class hierarchies," in *CVPR*, 2008.

[72] L.-J. Li, G. Wang, and null Li Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," *CVPR*, vol. 0, pp. 1–8, 2007.

[73] J. Philbin, J. Sivic, and A. Zisserman, "Geometric LDA: A generative model for particular object discovery," in *BMVC*, 2008.

[74] T. Hofmann, "Probabilistic Latent Semantic Indexing," in *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, (Berkeley, California), pp. 50–57, August 1999.

[75] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in *Proceedings of the International Conference on Computer Vision*, 2005.

[76] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pp. 525–531, 2001.

[77] J. Zobel and A. Moffat, "Inverted files for text search engines," *ACM Comput. Surv.*, vol. 38, no. 2, p. 6, 2006.

[78] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *CVPR*, pp. 1–8, 2008.

[79] Y. M. Wong, S. C. H. Hoi, and M. R. Lyu, "An empirical study on large-scale content-based image retrieval," in *Multimedia and Expo, 2007 IEEE International Conference on*, pp. 2206–2209, July 2007.

[80] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, (New York, NY, USA), pp. 253–262, ACM Press, 2004.

[81] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, (New York, NY, USA), pp. 604–613, ACM Press, 1998.

[82] H. Wu, Y. Wang, and X. Cheng, "Incremental probabilistic latent semantic analysis for automatic question recommendation," in *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, (New York, NY, USA), pp. 99–106, ACM, 2008.

[83] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *Proceedings of the International Conference on Computer Vision*, 2005.

[84] F. Jing, M. Li, L. Zhang, H. Zhang, and B. Zhang, "Learning in region-based image retrieval," in *Jing, F., Li, M., Zhang, L., Zhang, H.J., Zhang, B.: Learning in region-based image retrieval. In: International Conference on Image and Video Retrieval, Lecture Notes in Computer Science, vol. 2728, Springer (2003) 198–207*, 2003.

[85] CalPhotos, "Corel image database." at, http://elib.cs.berkeley.edu/corel/.

[86] P. E. Black, "Elastic bucket trie, dictionary of algorithms and data structures."

[87] J. Yang, Y. G. Jiang, A. G. Hauptmann, and C. W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, (New York, NY, USA), pp. 197–206, ACM, 2007.

[88] F. Fraundorfer, H. Stewenius, and D. Nister, "A binning scheme for fast hard drive based image search," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1–6, 2007.

[89] T. Deselaers, L. Pimenidis, and H. Ney, "Bag-of-visual-words models for adult image classification and filtering," in *International Conference on Pattern Recognition 2008*, (Tampa, Florida, USA), 08/12/2008 2008.

[90] M. Marszaek and C. Schmid, "Spatial weighting for bag-of-features," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2118–2125, 2006.

[91] Y. G. Jiang, C. W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, (New York, NY, USA), pp. 494–501, ACM, 2007.

[92] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in *ICCV*, 2007.

[93] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, vol. I of *LNCS*, pp. 304–317, oct 2008.

[94] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Second International Conference on Knowledge Discovery and Data Mining* (E. Simoudis, J. Han, and U. M. Fayyad, eds.), pp. 226–231, AAAI Press, 1996.

[95] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The Amsterdam library of object images," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 103–112, 2005.

[96] A. M. Dan Pelleg, "Accelerating exact k-means algorithms with geometric reasoning (extended version)." Technical Report CMU-CS-00-105, April 1999. Technical Report CMU-CS-00-105.

[97] G. Dorkó and C. Schmid, "Object class recognition using discriminative local features," Rapport de recherche RR-5497, INRIA - Rhone-Alpes, February 2005.