Learning in Large Scale Image Retrieval Systems

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

Master of Science
(by Research)
in
Computer Science and Engineering

by

Pradhee Tandon
200607020
pradhee@research.iiit.ac.in
http://research.iiit.ac.in/~pradhee

International Institute of Information Technology
Hyderabad, India
June 2009

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Learning in Large Scale Image Retrieval Systems" by Mr. Pradhee Tandon, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Advisor: Dr. C. V. Jawahar
Date	Advisor: Dr. Vikram Pudi

To my Dad,
Shri R. N. Tandon
and my Mom,
Smt. Dr. Sheela Tandon

Do much or say nothing at all.

Acknowledgments

I would like to thank Dr. C. V. Jawahar for his support and guidance during the past four years. I was fortunate enough to have him as my advisor. I sincerely appreciate all his help, generosity, patience and deep insights over perspective solutions to the problems. I would also like to express my sincere regards for his ever untiring urge to help me become an accomplished individual beyond a successful researcher.

I thank Dr. Vikram Pudi for his untiring guidance and encouragement at all stages of my masters. I feel fortunate to have him also, as my advisor. I am grateful to him for all the painstaking efforts he put in to help me triumph even if it meant considerable discomfort for him.

I also thank Dr. P. J. Narayanan, Dr. A. Namboodiri and Dr. J Sivaswamy for various references and guidance in different subjects related to the stream. I sincerely thank Mr. Piyush Nigam for all his support throughout with discussions on all aspects of the system, especially the implementation and look and feel. It has been a pleasure to learn from my seniors, Suman Karthik, Visesh, Paresh, Pavan, Tarun, Jagmohan, Vidit, Gopal and Karteek. Their clever comments and witty criticisms have many a time saved me from embarking on a wrong path. I would also like to thank Pooja, Anand, Pawan, Jyotirmoy, Neeba, Avinash, Praveen and Himanshu for proofreading my thesis. Being surrounded by so many intelligent and creative lab mates, always encouraging, ready to talk, argue, and shoot down ideas, was certainly the best thing about my entire graduate degree.

Finally, I would like to appreciate the patience and support from my parents, grand parents and my sister, Kritika. 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

Recent explosive growth in images and videos accessible to any individual on the Internet have made automatic management the prime choice. Contemporary systems used tags but over time they were found to be inadequate and unreliable. This has brought content based retrieval and management of such data to the fore front of research in the information retrieval community.

Content based retrieval methods generally represent visual information in the images or videos in terms of machine centric numeric features. These allow efficient processing and management of a large volume of data. These features are primitive and thus, are incapable of capturing the way humans perceive visual content. This leads to a semantic gap between the user and the system, resulting in poor user satisfaction. User centric techniques are needed which will help reduce this gap efficiently. Given the ever expanding volume of images and videos, techniques should also be able to retrieve in real time from millions of samples. A practical image retrieval approach in summary is expected to perform well on most of the following parameters, (1) acceptable accuracy, (2) efficiency, (3) minimal and non-cumbersome user input, (4) scalability to large collections (millions) of objects, (5) support interactive retrieval and (6) meaningful presentation of results. Most of the noted efforts in CBIR literature have focused primarily on providing answers to only subsets of the above. A real world system, on the other hand, requires practical solutions to nearly all of them. In this thesis we propose our solutions for an image retrieval application, keeping in mind the above expectations.

In this thesis we present our system for interactive image retrieval from huge databases. The system has a web-based interface which emphasizes ease of use and encourages interaction. It is modeled on the query-by-example paradigm. It uses an efficient B+-tree based dimensional indexing scheme for retrieving similar ones from millions of images in less than a second. Perception of visual similarity is subjective. Therefore, to be able to serve these varying interpretations the index has to be adaptive. Our system supports user interaction through feedback. Our index is designed to support changing similarity metrics using this feedback and is able to respond in sub-second retrieval times. We have also optimized the basic B+-tree based indexing scheme to achieve better performance when learning is available.

Content based access to images requires the visual information in them to be abstracted into some numeric features. These features generally represent low level visual characteristics of the data like colors, textures and shapes. They are inherently weak and cannot represent human perception of visual content, which has evolved over years of evolution. Relevance feedback from the user has been widely accepted as a means to bridge this semantic gap. In this thesis, we propose an inexpensive, feedback driven, feature relevance learning scheme. We estimate iteratively improving relevance weights for the low level numeric features. These weights capture the relevant visual content and are used to tune the similarity metric and iteratively improve retrieval for the active user. We propose to incrementally memorize this learning across users, for the set of relevant images in each query. This helps the system in incrementally converging to the popular content in the images in the database. We also use this learning in the similarity metric to tune retrieval further. Our learning scheme integrates seamlessly with our index making interactive accurate retrieval possible.

Feature based learning improves accuracy; however it is critically dependent on the low level features used. On the other hand, human perception is independent of it. Based on the underlying assumption, that user opinion on the same image remains similar over time and across users, a content free approach has recently become popular. The idea relies on collaborative filtering of user interaction logs for predicting the next set of results for the active user. This method performs better

by virtue of being independent of primitive features but suffers from the critical cold start problem. In this thesis we propose a Bayesian inference approach for integrating similarity information from these two complimentary sources. It also overcomes the critical shortcomings of the two paradigms. We pose the problem as that of posterior estimation. The logs provide a priori information in terms of co-occurrence of images. Visual similarity provides the evidence of matching. We efficiently archive and update the co-occurrence relationships facilitating sub-second retrieval.

Studies have shown that the user refines his query based on the results he is shown. Studies have also shown that quality of retrieval improves with the effectiveness of learning acquired by the system. Presenting the right images to the user for his feedback can thus result in the best retrieval. A set of results which are similar to the query in different ways can help the user narrow down to his intended query at the earliest. This diversity cannot be achieved by similarity retrieval methods. In this thesis, we propose to efficiently use skyline queries for effectively removing conceptual redundancy from the retrieval set. Such a diversely similar set of results is then presented to the user for his feedback. We use our indexing scheme to extract the skyline efficiently, a computationally prohibitive process otherwise. User's perception changes with the results, so should the nature of the diverse set. We propose the idea of preferential skylines to handle this. We use the user preference, based on feedback, to adapt the retrieval to the user intent. We reduce the diversity and include more similarity from the preferred attributes. Thus, we are slowly able to tune the retrieval to match the user's exact intent. This provides improved accuracy and in far fewer iterations.

We validate all the ideas proposed in the thesis with experiments on real natural images. We also employ synthetic datasets for other computational experiments. We would like to mention that in spite of the considerable improvements in accuracy with our learning approaches, the effectiveness our solutions is still limited by the features used for encoding the human visual perception. Our methods of learning and other optimizations are only a means to reduce this gap. We present extensive experimental results with discussions validating different aspects of our expectations from the proposed ideas, throughout this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent explosive growth in images and videos accessible to any individual on the Internet have made automatic management the prime choice. Contemporary systems used tags but over time they were found to be inadequate and unreliable. This has brought content based retrieval and management of such data to the fore front of research in the retrieval community.

Of late many popular image management systems have sprung up like, Yahoo's Flickr [1], Google's Picasa [2] and Facebook [3]. Most of them still rely on user tags for organization and thereby restricting information access. Content based retrieval of images is the only option in such scenarios.

Content based image retrieval or CBIR as it is popularly known as, typically tries to represent the information in an image using numeric features. To handle huge collections, such as millions and billions of images special indexing schemes have been proposed [4, 5, 6, 7]. Most of theses schemes [4, 5, 6] are rigid and do not scale up well with database complexity. We need adaptive, efficient indexing solutions like the one in [7].

(a)

(b)

Figure 1.1: Semantic gap: In (a) Color based features are unable to differentiate between these images representing different concepts. In (b) Shapes cannot differentiate between the different roses.

The features used for representing images automatically, are generally composed of primitive

visual features like colors, textures and shapes. The way we perceive visual content can be very different from what the machine can understand. This results in a *semantic gap* between the two as can be seen in the image in Figure 1.1. Learning has been explored as an option for the system to bridge this gap in a mechanism shown in Figure 1.2. Complex learning schemes require prohibitive computational resources, specially when the data is huge. The need is of efficient learning schemes which can reduce the semantic gap appreciably. This learning is absorbed as relevance feedback on the set of results displayed to the user. These schemes can learn to use combinations of low level features. Given the immense user base of the Internet, schemes efficiently using behavior patterns for improved retrieval are also interesting. Behavior logs are free of dependencies on features, by making use of all the information available, which makes them popular. These approaches depend on large volumes of logs for acceptable performance. This makes hybrid approaches combining the two worth exploring. Users generally mark only a very small set of images. This results in unreliable

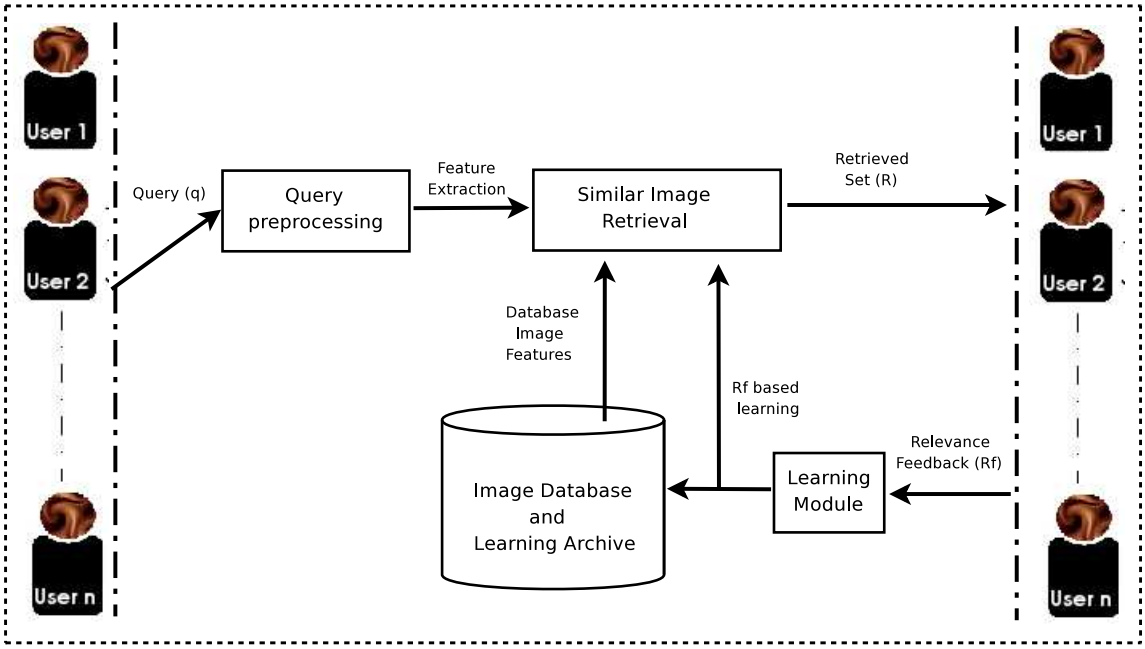


Figure 1.2: Conceptual representation of learning in image retrieval

or weak feedback learning. Methods to encourage the user to provide more feedback should be incorporated. Though simpler interfaces help yet active learning and effective presentation of results is more useful.

Motivated with some of these opportunities we attempt to build a system for interactive image retrieval in this thesis. It supports learning from relevance feedback and retrieves efficiently from a database of millions of images. We also propose our efficient algorithm for learning the image content for improved retrieval. We also propose a Bayesian inference framework for integrating learning from feature relevance and user logs. We propose to use skyline queries from database research for retrieving a diversely similar set of results for the user. Throughout this thesis, we validate the algorithms and ideas with extensive experiments using real and synthetic data supported by detailed discussions.

1.1 Background

In a typical content-based retrieval system, the contents of the media in the database are extracted and described as multi-dimensional feature vectors, also called descriptors. To retrieve desired data, users submit query examples to the retrieval system. The system then represents these examples with feature vectors. The distances (i.e., similarities) between the feature vectors of the query example and those of the media in the feature dataset are then computed and ranked. Retrieval is conducted by applying an indexing scheme to provide an efficient way to search the media database Finally, the system returns the most similar few to the user. Around this methodology researchers have proposed numerous enhancements for achieving better performance, in terms of accuracy, efficiency and usability. In brief, these would broadly include user interaction, learning and presentation. We shall next, briefly review progress in some of these otherwise independent areas in the following sections.

1.1.1 Image Retrieval Systems

Over the history of CBIR research many systems have been designed catering to some or many of the basic needs of a CBIR engine [8]. While many remain confined to research laboratories, some have been made public too. Before exploring the contributions in this thesis it would be interesting to briefly review the features of some of these. Detailed analysis on the past, present and possible future directions of research can be found in some noted surveys [9, 10, 11, 12] in CBIR.

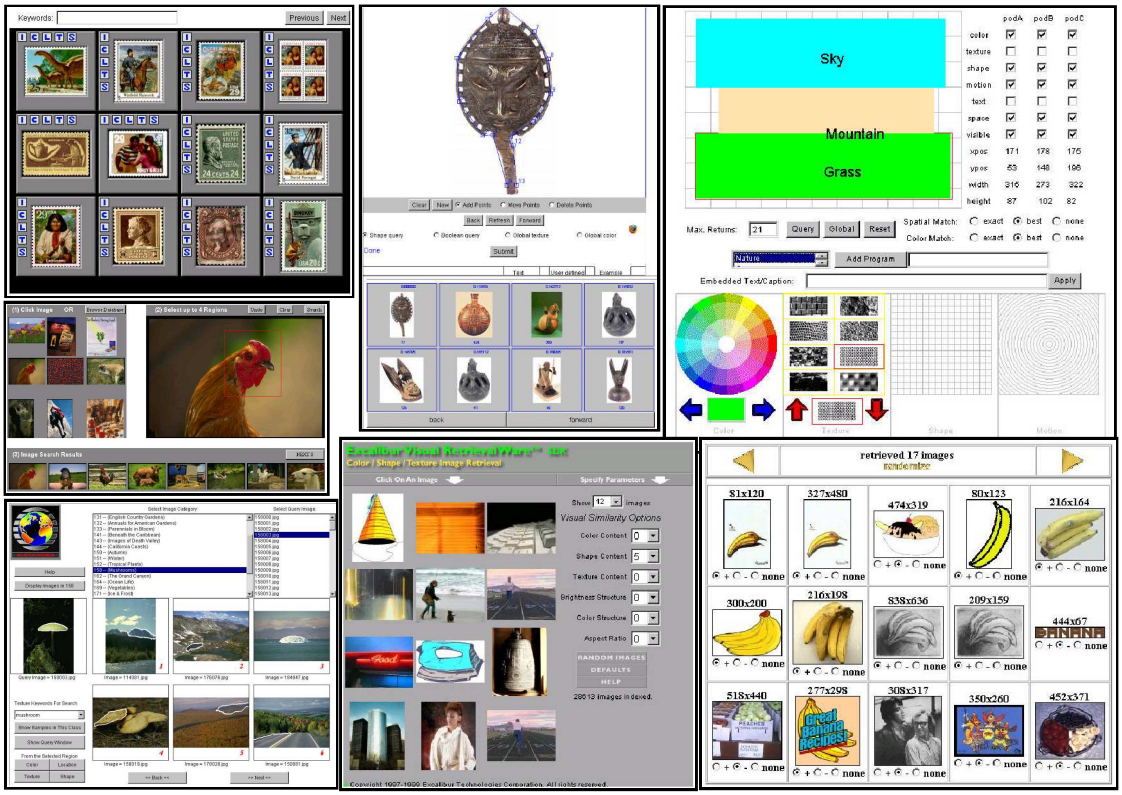


Figure 1.3: Historical image retrieval systems, (clockwise from top-left) QBIC, MARS, VisualSEEK, WebSEEK, RetrievalWare, Netra and Sypanse.

The most prominent ones from history would be QBIC, Virage and RetrieverWare in the com-

mercial domain Figure 1.3. IBMs QBIC system [13] is the best-known among all. It offers retrieval by any combination of colour, texture or shape as well as by text keyword. Image queries can be formulated by selection from a palette, as an example image, or by sketching. Retrieval uses an R*-tree index for efficiency. The VIR Image Engine from Virage, Inc [14] is another well-known system. It supports modular development and is available as Oracle DB add-ons. It is used to power the Photo Finder system from Alta Vista. Excalibur Technologies' Visual RetrievalWare offers a variety of their proprietary indexing and matching techniques. It is best known for use with Yahoo! Image Surfer.

A number of academic systems were also developed over time. Photobook system [15] from MIT proved to be one of the most influential of the early CBIR systems. It computes information preserving features to support reconstruction of visual information, in addition to the usual visual features. It is used by US Police with the FaceID project. Another known early system was Chabot [16], which provided a combination of text-based and colour-based access to a collection of digitized photographs held by Californias Department of Water Resources. The system has been renamed Cypress, and incorporated within the Berkeley Digital Library project at the University of California at Berkeley (UCB). The VisualSEEk system [17] offers searching by image region colour, shape and spatial location, as well as by keyword. Users can build image queries by specifying areas of defined shape and colour at absolute or relative locations within the image. The WebSEEk system [18] clusters by text and then queries inside clusters using colors. Relevance feedback is also supported. MARS project at the University of Illinois [19] brought the user to the fore. Relevance feedback is an integral part of the system and is used for learning feature weights, and if necessary to invoke different similarity measures [20]. Surfimage system from INRIA, France [21] is similar to MARS and offers combinations of features and supports relevance feedback. The Netra system [22] uses colour texture, shape and spatial location information with image segmentation to provide region-based searching based on local image properties. Synapse [23] on the other hand, is an implementation of retrieval by appearance using whole image matching.

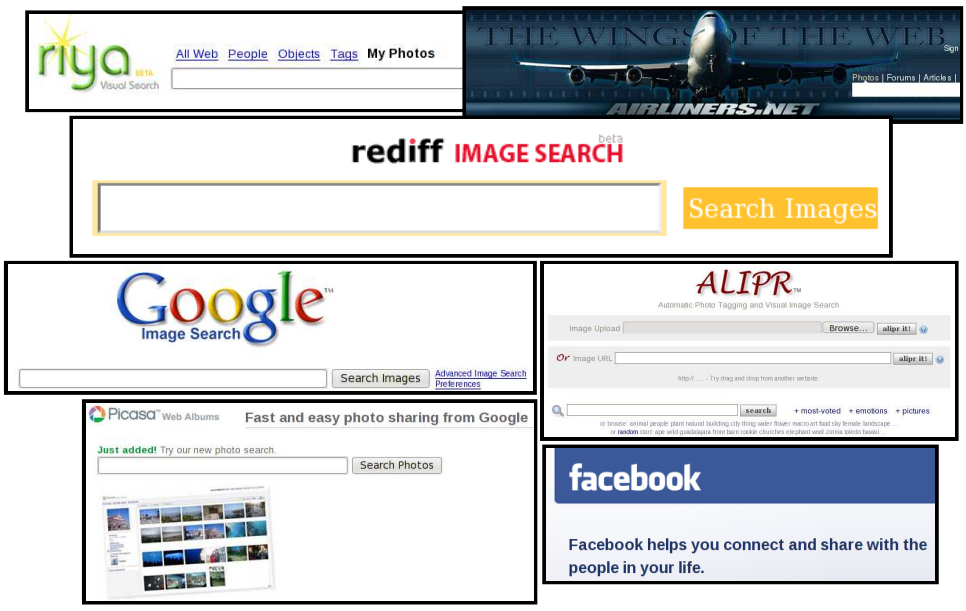


Figure 1.4: Recent image search systems, (left to right, top to bottom) Riya, Airliners.net, Yahoo Image Search, Google Image Search, ALIPR, Picasa and Flickr.

Google Image Search [24] and Yahoo! Image Search [25] were the only major public image search

engines deployed in the last decade. They primarily use surrounding meta data. Riya, a public-domain search engine allows searching for pictures of products and people. CBIR technology is being applied to diverse domains like family album management, botany, astronomy, mineralogy, and remote sensing [26, 27, 28, 29, 30]. Airliners.net was released as a public search tool for 800,000 airline-related images by [31]. GlobalMemoryNet [32] integrates similarity search functionality into a large collection of art and cultural images. Terragalleria [33] incorporates image similarity with a massive picture travel pictures archive. Automatic Linguistic Indexing of PicturesReal-Time (ALIPR), an automatic image annotation system [34, 35], has been designed for validating computer generated tags. Another good work is a Web image search system [36] that exploits visual features and textual meta data. Very recently, Google-labs has added an image-to-image similarity based retrieval feature on top of their primarily meta data based image retrieval backbone.

Systems discussed above summarize the growth of CBIR. Given the major hindrance to CBIR's popularity is the gap between user expectations and system performance we shall next briefly review the two, before proceeding to our ideas.

1.1.2 User Experiences and Expectations

User satisfaction being the prime goal, human centered approach is crucial. Understanding the users can be beneficial. User can be effectively categorized based on their intents. These intents define how should the system react to the query, what are the expectations from it etc. *Browser* is a user looking for pictures with no clear end-goal. Her session would consist of a series of unrelated searches jumping across multiple topics. Her queries would be incoherent and diverse in topic. *Surfer* is a user surfing with moderate clarity of an end-goal. Her actions may be somewhat exploratory in the beginning, with increasing clarity. *Searcher* is very clear about what she is searching for. Her session would be typically short and coherently leading to a result. While the browser values ease of use, a surfer appreciates help in forming her query, the browser values completeness and clarity of response.

In image retrieval, complexity of querying is an important parameter for gaging the interaction. This deals with the different modalities which can be used for querying like, text, features, visual examples or graphics. Though querying by example is the most appealing, it may not be the most convenient. Composite queries with text and visuals with feedback, would allow maximum flexibility to the user. Some recent innovations support sketch-based retrieval of color images [37] and also 3D model [38] based querying. An interesting work in multi-modal systems involves hand gestures and speech for querying and feedback [39]. Certain new methods statistically model the users interest [40], or help the user refine her queries with cues and hints [41, 42].

Although, querying is at the core, but efficient retrieval with acceptable accuracy are also equally important for user satisfaction. Quality of retrieval is still dependent on the capabilities of representation schemes and effective learning schemes.

1.1.3 Learning in Image Retrieval

Over the last decade of CBIR research, many new representational schemes have been proposed and become popular but still the major emphasis has been on better, more useful learning schemes rather than better representation. Technically, researchers have concentrated on bridging the semantic gap with better learning schemes while trying to best utilize user interaction. This has primarily been motivated by the sheer numbers of web based user always available to provide feedback, implicitly or explicitly. We shall briefly review the trends in learning in CBIR here.

As the target user base is on the web, learning needs to adapt techniques suitable for the web, in

terms of user interaction, response times, utility and usability as discussed above. Learning techniques can be broadly classified as in [12], into classification, clustering and relevance feedback techniques. Analysis shows that classification is able to achieve fast accurate response by pre-processing information using tools like SVMs, statistical models, Bayesian classifiers, trees [43, 44, 45]. This pre-processing requires prior availability of reliable training data which can introduce unwanted biases. This technique involves very little user involvement and is completely non-interactive and thus unadaptive to his needs. Clustering on the other hand allows meaningful result visualization, efficient storage and processing using methods like k-means, kernels, metric learning [44, 46, 47, 48]. Though the user effort is minimized yet the complete lack of adaptability reduces its usefulness to our user base. Relevance feedback is user interaction centric and fits the adaptive requirement well. It relies on simple feature re-weighting techniques and memory learning, boosting [20, 40, 41, 44] etc. methods to support interactive retrieval. It is able to reliably produce user specific results and semantics but relies primarily on low level features. It needs support from efficient indexing and effective learning methodologies to serve the web user base.

Relevance feedback (RF) is a query modification technique which attempts to capture the users precise needs through iterative feedback . In the absence of means of learning higher level semantics, it provides an effective tool for capturing user semantics. Comprehensive reviews can be found in [20, 49, 50]. We present a short overview of recent work in RF, and the various ways in which these advances can be categorized, here. We group them here based on the nature of the advancements made, resulting in (possibly overlapping) sets of techniques. These include: (a) learning-based, (b) feedback specification, (c) user-driven, (d) probabilistic, (e) region-based, and (f) other advancements.

Based on the users relevant feedback, learning-based approaches are typically used to appropriately modify the feature set or similarity measure. However, in practice, a users RF results in only a small number of labeled images pertaining to each high-level concept. This has encouraged techniques in one class learning, manifolds and active learning. A discriminant-EM algorithm is proposed to make use of unlabeled images in the database for selecting more discriminating features [51]. Positive samples are often organized more consistently in the data space. This leads to a natural formulation of one-class SVM for learning relevant regions. Among other techniques, a principled approach to optimal learning from RF is explored in [52]. We can also view RF as an active learning process, where the learner chooses an appropriate subset for feedback from the user in each round based on her previous rounds of feedback, instead of choosing a random subset. Active learning using SVMs was introduced into RF in [53]. Extensions to active learning have also been proposed [54, 55]. Unlike traditional RF, recently feedback based directly on an image's semantics characterized by manually defined image labels, appropriately termed semantic feedback, was proposed in [56].

A well-known issue with feedback solicitation is of user patience. To quicken the process, user logs of earlier feedback sessions can be used in query refinement, thus reducing the user iterations, as in [57]. Innovation has also come in the form of the nature by which feedback is specified by the user. [58] introduces the concept of multiple queries in the intermediate steps. In order to address this asymmetry during RF when treating it as a two-class problem, a biased discriminant-analysis-based approach has been proposed in [59]. While most algorithms treat RF as a two-class problem, it is often intuitive to consider multiple groups of images as relevant or irrelevant [57, 59, 60]. For example, a user looking for cars can highlight groups of blue and red cars as relevant, since it may not be possible to represent the concept of car uniformly in a visual feature space. Another novelty in feedback specification is the use of multilevel relevance scores to indicate varying degrees of relevance [61]. Recently, providing cues and hints to the user has been considered in [41, 42]. A similar search paradigm proposed in [40, 57] by making use of logs that contain earlier feedback

given by that user.

Probabilistic models have found increasing patronage for performing RF in recent years. Probabilistic approaches have been taken in [62, 63, 64]. In [62], the PicHunter system is proposed, where uncertainty about the users goal is represented by a distribution over the potential goals, following which the Bayes rule helps in selecting the target image . In [63], RF is incorporated using a Bayesian-classifier-based re-ranking of the images after each feedback step. Another RF approach is based on the intuition that the systems belief at a particular time about the users intent is a prior, while the subsequent user feedback is new information obtained.

Besides the grouped sets of methods, there have been a number of isolated advancements covering various aspects of RF. For example, methods for performing RF, using visual as well as textual features (meta data) in unified frameworks, have been reported in [65, 66, 67, 68]. A tree-structured SOM has been used as an underlying technique for RF in a CBIR system in [69]. A well-known RF problem regarding query specification is the fact that after each round of user interaction, the top query results need to be recomputed following some modification. A way to speed-up this nearest neighbor search is proposed in [70]. The use of RF for helping to capture the relationship between low-level features and high-level semantics, a fundamental problem in image retrieval, has been attempted using logs of user feed backs in [71].

Relevance feedback provides a compromise between a fully automated, unsupervised system and one based on subjective user needs. While query refinement is an attractive proposal when considering a very diverse user base, there is also the question of how well the feedback can be utilized for refinement. New approaches such as [40, 41] have started incorporating the evolving query of the user in mind in the RF process. The lack of popularity of relevance feedback techniques in real world systems is mainly due to the feedback process. Memory based retrieval and other techniques should help in future.

1.2 Scope of this thesis

1.2.1 Motivation

Traditionally, content based image retrieval relies on low level representation schemes for efficient retrieval. These are generally incapable of expressing the user's high level concepts and as a result both, accuracy and user satisfaction suffer. Relevance feedback is believed to be the best input for bridging this semantic gap. This approach of relevance feedback has been extensively studied in CBIR literature over the last decade.

Most of the techniques discard this expensive user feedback once the session is over. Some recent works have tried to use this learning to benefit subsequent users or queries. This learning if accumulated incrementally, can also be interpreted as popular content in the images. It is validated by many user over a period of time so the popularity is also reliable. This learning can directly be used for improving retrieval for all subsequent users and queries. Most of the techniques discussed in literature are computationally expensive. For real time interactive systems efficient approaches are required. Even though such approaches would commendably reduce the semantic gap yet they are inherently dependent on the feature representation. User judgment is independent of the representation. As a result, his co-relevance judgments can be used as semantic relationships among images. These can be used for retrieving results when similar patterns of feedback are presented to the system. But the accuracy of such approaches is critically dependent on quantity and quality of logs. Small set of marked images, further compounds problems with both the classes of learning. Effective integration of knowledge from these two complimentary paradigms would result in better retrieval performance.

User query is known to be vague at the start and forms iteratively based on the results shown. Presenting him results which are most informative or cover maximum variation in his possible query concepts would help. Set of diversely similar results would help as the user would be able to quickly identify his intended concept. This would also improve interaction by reducing the number of feedback iterations required from the user.

Most of the popular CBIR approaches address only some of the critical issues plaguing the community. We propose to present some solutions so some of the key problems of efficiency and quality of retrieval in CBIR.

1.2.2 Objective

We strongly believe in the need for a content based image retrieval system which should be able to retrieve in perception time with acceptable accuracy. It should encourage and make effective use of user interaction in terms of his feedback. It should use this feedback to optimally benefit the user by bridging the semantic gap with low level features. It should also utilize the implicit relationships embedded in user feedback patterns. These should be efficiently integrated into the interactive system for practical utility. The systems should also be able to help the user with a more intelligent presentation of results rather than suggesting just the top few. These could take into consideration qualities like diversity or lack of redundancy as primary expectations.

We would like to propose solutions for efficient and accurate CBIR, while keeping it user centric.

1.2.3 Focus

In this thesis we have focused on building an image retrieval solution using visual content. We have concentrated our efforts on designing a user friendly web based system to optimize user interaction by maximizing information gain from minimal user effort.

Our system retrieves results in perception time using our efficient indexing approach. We have also focused on best absorbing and using the user feedback on the relevance of the results presented to him with our proposed learning methods. We have also worked on a novel method of improving the quality of retrieved results across users and queries using feedback both by learning visual content and using user behavior patterns. These result in improved user experience in time (iterations and effort) and accuracy of results returned to him. Paucity of feedback for reliable inferences is another key issue of our focus. We view this coupled with the issue of improved user experience in terms of the informativeness of the set of results presented to him. We have also proposed a unique approach for achieving this mix.

In the broader perspective, we have focused on some of the most problems facing image retrieval by content. These constitute problems which are the root cause of such systems being commercially non-viable. We hope that our proposed techniques and those like ours would ensure CBIR systems are not made by scientists for scientists only and they does not fade into isolation of research.

1.2.4 Contributions

In this thesis we have proposed some solutions to some important problems in content based image retrieval. We have designed an image retrieval system called **FISH** for **F**ast **I**mage **S**earch in **H**uge databases. The system has a web based interface which supports query-by-example. The system supports feedback collection with minimal user effort. The system uses a B+-tree based dimensions indexing scheme to achieve interactive response times.

We also propose our novel approach for efficient long term content memorization. We use relevance feedback from the user to learn the popular content in his query as well as the relevant

images in the database. We incrementally memorize this learning across users and queries. Our learn and use it across users and queries. Its inclusion results commendably reduces the semantic gap resulting in improved accuracy in interactive response times.

Feedback patterns and logs are user evaluations of semantic similarity and independent of the low level features. Collaborative filtering of such logs has been researched by general information retrieval community extensively. We propose to use these relationships among images in addition to the feedback based judgment of the important features for retrieval. We propose a Bayesian inference approach for integrating these complimentary forms of relevance knowledge. We pose it as a posterior estimation problem given the log based prior relationships and feature based visual similarity evidence. We show that our scheme achieves much improved performance by overcoming their respective dependencies on logs and representation.

Paucity of feedback is a key issue in image retrieval. Subjectivity of user feedback and interpretation further compounds the problem. We believe that presenting the user a more informative set can improve is interaction. This becomes important in light of the belief that user query too evolves iteratively with the results. Conceptual diversity in the retrieved set would be a helpful improvement. We propose to use skyline queries from database research community to achieve this in an efficient algorithmic manner. We also propose the use of our efficient indexing scheme for efficiently computing the skyline set over a set of similar results for the user query.

1.2.5 Organization of the thesis

We have organized the work in thesis in a manner where each contribution adds to, and seamlessly fits with all the previous ones. We first discuss in detail the design and features of our **FISH** system. We present extensive analysis of the index structure and the performance related aspects of **FISH** in Chapter 2. We experimentally validate the ideas and claims on the capabilities of **FISH** as a scalable image retrieval system. We also elucidate the system building aspects of **FISH**.

Next, in Chapter 3, we present our proposed technique for long term learning and memory. We discuss the formulation and present extensive experimental results using multiple metrics and datasets with different characteristics. We show results from **FISH** based experiments to validate the practical utility of our proposed approach. We also present our novel content learning technique where we are able to over sessions ascertain the popular content in the images and use it for improved retrieval.

Then, in Chapter 4, we propose our Bayesian framework for integrating the *a priory* knowledge acquired from the feedback logs and the evidence from the visual similarity. We compare the advantages of our approach with a pure feature based learning CBIR. We present both quantitative and qualitative proof of our concept.

In Chapter 5, we present our novel approach for achieving diversity in similarity retrieval. We discuss our approach of using skyline for the same. We extensively prove the efficiency of our approach using our indexing scheme, over real and synthetic datasets. We also present results of queries showing diversity.

We conclude in Chapter 6 with some remarks on the proposed framework and learning algorithms. We also discuss some limitations and interesting future directions leading from our work.

Chapter 2

The FISH System

2.1 Introduction

Since the last few decades, huge amounts of multimedia data are being generated and stored digitally. The reasons for this are the widespread use of good multimedia capture devices and the availability of virtually infinite storage capacity. Even personal collections have become manually unmanageable. Content-aware automatic multimedia data management systems are needed to address this problem. One approach is to annotate multimedia data with textual tags representing the semantics of the data. However, the sheer volume makes annotation impractical for most scenarios. Subjectivity of the interpretation also renders this approach inadequate.

This necessitates the use of machine-centric data features which can be automatically extracted, instead of tags. These can then be used for indexing, search, retrieval and comparison of multimedia data. All these features are not equally relevant for all the data objects. Depending on the semantics captured in the data, different features must be given different weights. In a typical multimedia retrieval system, similarity is computed based on the features and the weights for each stored object. The retrieved objects are then presented to the user for his feedback.

The systems should ideally scale to millions of objects and the retrieval is expected to happen in interactive time. Exhaustive retrieval from millions of samples is in itself prohibitively inefficient. Good indexing schemes are needed which support changing metrics and perform acceptable approximate retrieval.

However, a multimedia object retrieval system has various requirements that are not well-supported by most of the current state-of-the-art systems. Although the current systems incorporate novel and elegant ideas, most of them are built mainly as a proof-of-concept, and not to be deployed on a large scale. Practical systems have the following broad requirements:

- **Simple Interface:** A useful system must aim for maximizing information capture while minimizing user effort. Literature talks about querying methods ranging right from query-by-example and sketching to low level numeric feature description. The feedback collection too should be as implicit to the user as possible rather than asking the user to rate every result on a scale of relevance. These aspects make the querying and iterations tedious for the user and are quite detrimental from the usability perspective.

- **Acceptable Accuracy:** Any system to be usable needs to retrieve with acceptable accuracy. Therefore an image retrieval system should support efficient and effective use of learning to bridge the semantic gap between the machine centric representation and the conceptual intent of the human users. The learning schemes should adapt seamlessly to highly scaled up setups.

The systems should be able to support learning across sessions and even specific individual users.

- **Dynamic Databases:** Multimedia data collections are almost always dynamic in nature. Hence such systems must seamlessly handle new objects inserted into the collection.

- **Adaptive Indexing:** A good index structure is required to efficiently retrieve multimedia objects that are similar to a query object [4, 5, 6, 13]. While many schemes exist in the literature, most of them are designed to work with a *fixed* similarity metric. These do not suit our environment since the similarity metric depends on the weights of the features, which continually change with fresh queries and user-feedback.

- **Interactive response:** Practical systems must be optimized for efficiency to the point where multimedia retrieval sessions become interactive for a user. The response time must be suitable for web-based search engines, where it would be frustrating for users to wait for more than a few seconds at most.

- **Scalability:** Typical multimedia collections are not small by any standards. Web-based systems for search and retrieval are huge – spanning to *millions* of objects [72]. In contrast, most of the existing techniques have been demonstrated only for much smaller collections involving thousands of objects. Practical systems should rely on schemes which seamlessly scale while retaining interactive response times. These should ideally be inexpensive resource based approaches making effective use of distributed computing.

- **Modular extensible design:** In addition to incorporating the above requirements, the resulting system must be easily extensible to new unforeseen requirements. This leaves us in favor of systems that are *simple* to understand, design, implement and modify. In contrast, most existing techniques, while being mathematically robust, are complex and hard to "break into pieces" – a quality essential for extensibility.

As a part of this thesis, we have built a system for interactive image retrieval. **F**ast **I**mage **S**earch in **H**uge databases or **FISH**, has been designed as a system for interactive image retrieval from huge real world image collections. It relies on an efficient multi-dimensional indexing scheme. The interface encourages user interactions and the indexing also allows seamless integration of relevance feedback based learning methods for improved performance, both in response time and accuracy. A demonstration version of the system is available on the web at the URL: *http://cvit.iiit.ac.in/fish*.

We focus on image collections, although the techniques we discuss are also applicable to general multimedia data collections. We perform an extensive experimental study to validate these claims on both real and synthetic datasets. Our results show that the developed system *easily scales to millions of complex real images while still maintaining interactive response time*.

2.2 Related Work

Any real-world image retrieval system must effectively incorporate most of the ideas enlisted in Section 2.1 to make them practically usable. Many of these have, in isolation, been addressed in the past. However, the combination of all the listed criteria has so far not been achieved in a single system. In this section we briefly review the research efforts closest to our method in various aspects of the retrieval pipeline. A detailed review of the prominent systems for image retrieval can be read in Section 1.1.1.

Representation of visual content of the images has received considerable attention throughout the history of image retrieval [73, 74, 75]. Methods have explored features ranging from simple global color histograms to color layouts and structures, from point based shape matching to domain centric structural information and from moment based textures to wavelet transforms, all the way to robust highly descriptive point-based features such as SIFT [76]. The choice of representation is effected by different factors like the special characteristics favored by the domain of deployment [31] and even the system's sensitivity to computation overheads with complex features.

The retrieval of images in response to a query should match the user's intent. This requires the systems to incorporate methods to absorb and use semantic input from some external guide, optimally the human user, and improve retrieval. This human input is primarily acquired in the form of his feedback on the relevance and irrelevance of the images in the set returned to him in response to his query. These approaches primarily improve the performance for the present query only, better known as intra-query or short term learning approaches as studied in [20, 49]. Though most of the approaches discard the expensive and invaluable feedback from the user after every query session, some researchers have explored learning from one query to benefit the subsequent ones. This category of techniques is known as inter-query or long term learning methods [77, 78].

Scalability to huge datasets is a key in real world systems. The problem of retrieving the exact k nearest neighbors from a large dataset is prohibitively time consuming. Thus there is a need to best approximate the actual k neighbors. A large body of work explores index structures for supporting similarity search in large datasets. These involve approaches like k-d trees [4], R-tree [5], SS-trees [6] and their variations. An important aspect to keep in mind when choosing an indexing scheme is that most of the popular learning driven retrieval approaches effect the comparison metric in some way. This necessitates that the indexing scheme should be adaptive to changing similarity metrics while the above solutions assume a fixed metric at the time of indexing. They tune the indexing according to the metric to approximate the k nearest neighbors and are thus unsuitable in our scheme of things. There has been some recent work in the area of efficient search with changing metrics [79, 80]. These approaches primarily use a branch and bound methodology for their purpose, which can degrade to exhaustive retrieval over the entire dataset. Some recent proposals explore the use of B+ trees [7, 72] for efficient indexing. These schemes though effective fail to take into account the inherent nature of the data to form clusters based on a few out of the entire set of features. The data is generally clustered tightly over this small subset of dominant dimensions. This pattern of relative dominance corresponds to a concept in the image, so there can be many of them in an image. This multiplicity can be handled with some modifications to the scheme proposed in this work.

User interaction with the system being the key element in terms of his feedback on the retrieved images, the usability of the interface becomes a very important consideration from the perspective of developing a complete solution. The interaction stretches from the querying mechanism through the display of retrieved results to the method of feedback collection. For keeping the querying interface simple, Query by Example (QBE) and sketching methods should be preferred over methods requiring low level specification in terms of features and their weights [81, 82, 83]. Feedback should also be simplified to optimize learning by increasing the number of user iterations.

In this thesis, we have proposed some new techniques and adapted some proposed in literature seamlessly into our framework. Our proposal uses a set of standard visual descriptors with a highly adaptive indexing scheme which supports the inherent organization of data into similarity clusters. We also adopted a set of popular relevance feedback approaches from literature. We proposed a highly suited scheme for inexpensive inter-query learning. We validate all our claims on a huge real image dataset using our system, developed with a usable and effective user interface.

2.3 FISH Architecture and Retrieval Process

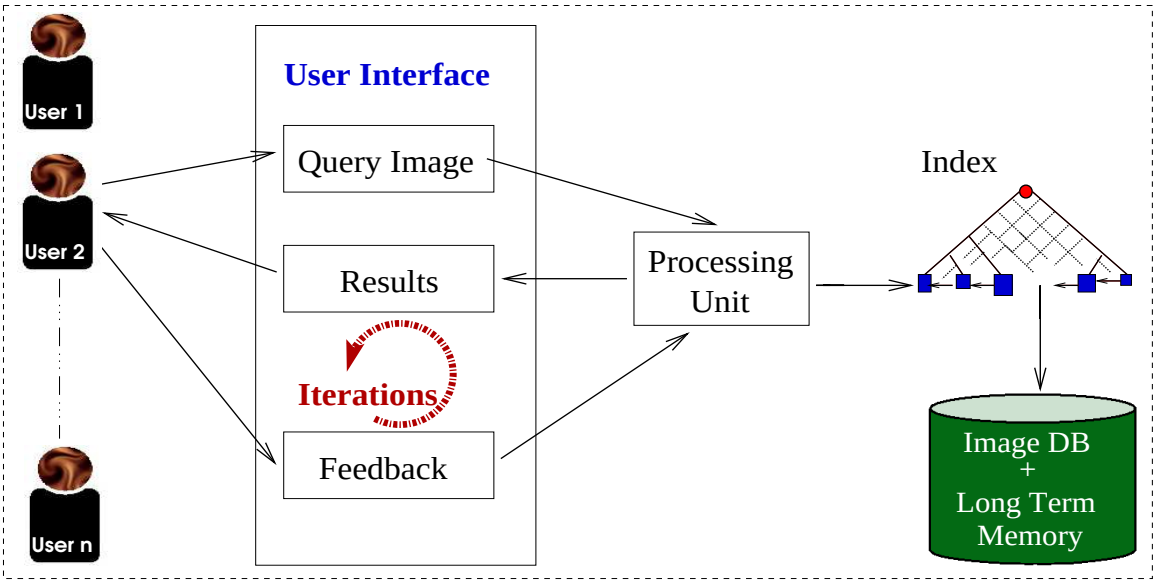


Figure 2.1: FISH Architecture: A Conceptual Presentation

Fast **I**mage **S**earch in **H**uge databases or **FISH** was designed to answer some of the core issues plaguing the image retrieval community. We kept efficiency, usability and utility at the fore-front. **FISH** as a result is able to efficiently present acceptably accurate results from huge databases in interactive time by making the most of minimal user effort. **FISH** was designed as a web-based system where users can simply upload an image of their choice and interactively deal with the set of similar images presented to them, from our collection. The system incorporates features like limited number of results, pagination, tab-key control etc. to make human interaction effortlessly enjoyable. The overall architecture of the system is shown in Figure 2.1. Through the user-interface, which is a web-front end, users (shown on the left) provide query images to the system.

Once received, the system processes the query image into an internal representation and searches for similar images in a large database. The search for similar images is made faster by the use of an appropriate index structure. We use a B+-tree to index features from every dimension. We approximate the k-NN images to the query by efficiently retrieving t ($t >> k$) closest samples from every dimension. The system is able to respond in sub-second or perception time with a set of similar images. The retrieved similar images are then shown back to the user over multiple pages of results.

The user then has a chance to give feedback to the system as to whether each retrieved image is indeed similar to his query. To further reduce user effort, all images are assumed to be non-relevant by default, so the user only has to click and select the images he feels are relevant in order to provide useful feedback. The system uses this feedback to learn and interpret the user's intent better and provide a more relevant set of results in the next iteration. These iterations are continued on user's interest.

The ability of the system to provide better results by learning the intent of the user within a session is referred to as short-term learning. This learned knowledge is represented succinctly and stored in the "long-term memory" for providing better results in later queries also. **FISH** maximizes the utilization of expensive user feedback by using it for both the active query as well

as subsequent users and queries.

2.4 The System

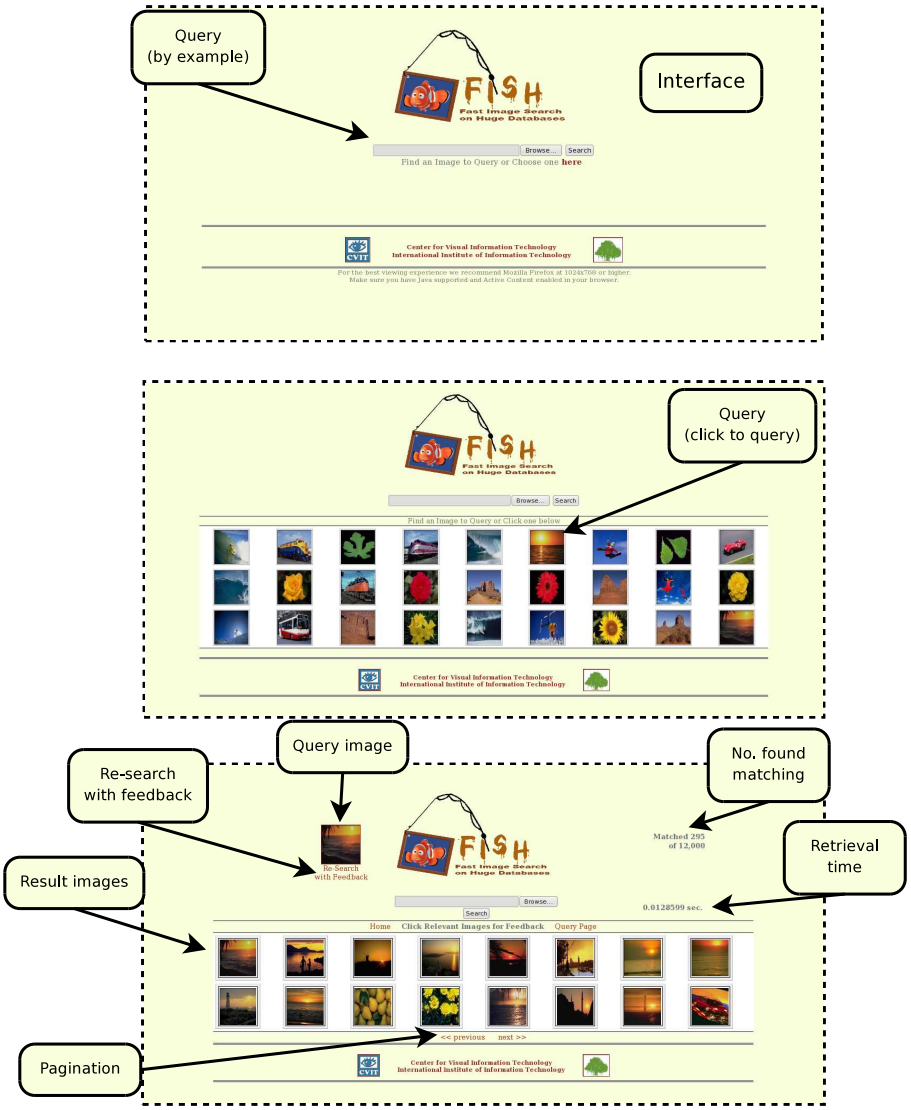


Figure 2.2: The image highlights the functions of various aspects of the **FISH** interface. It shows how the user can either upload a query image or select one from the random set of suggestions. This helps especially if he is there only to test the system. The interface also presents inter sting statistics of performance like the total number of samples found relevant to the query (based on a threshold cut-off) and the retrieval time for the query. The interface presents results paginated over multiple pages to help visualization.

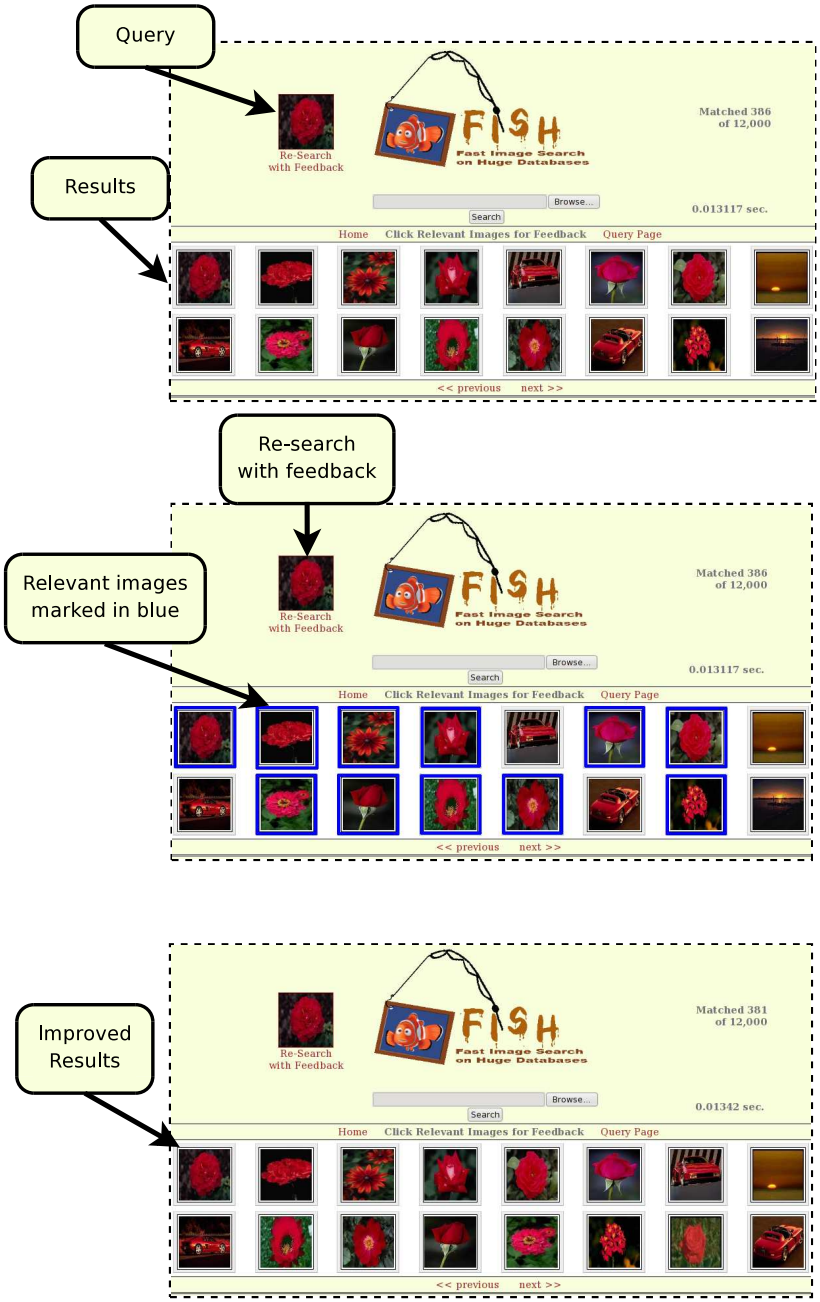


Figure 2.3: The image shows how query and retrieval proceed in presence of relevance feedback and learning. First the user is presented with a set of results in response to his query. Then he is expected to mark the relevant images by clicking them (shown in highlight) and click re-search. The system uses this input to learn and retrieved results with higher accuracy.

2.5 Indexing

In this section we discuss the use of index structures to quicken the retrieval of similar images. In Section 2.5.1 we discuss the selection of an index structure for the system and it details, briefly. We then improve upon it in Section 2.5.2.

2.5.1 Index Structure Selection

Image retrieval systems need to retrieve images from a dataset one by one and compare them with the query image for similarity. In principle this process can be quickened with the help of a nearest-neighbor index structure that can retrieve images in the neighborhood of a query image.

However, most existing indexing schemes require a fixed similarity metric with which the index is built [4, 5, 6]. Such schemes do not suit us as our metric changes with queries and feedback. Some schemes support changing metric like those in [79, 80]. However, most of these enumerate a large number of candidate images as they treat all features uniformly. Real-life datasets have inherent clusters in them resulting in highly skewed biases towards specific features. Our indexing scheme [7] takes advantage of this inherent characteristic of multimedia data and flourishes in this environment.

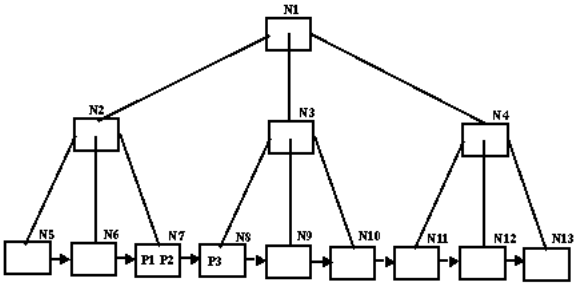


Figure 2.4: Schematic representation of our B+ - tree based index.

The Working

The index structure (as shown in Figure 2.4) is simple: For each of the D dimensions, a list is maintained that contains all the data points sorted along that dimension. The lists can be stored on disk and be implemented as B+-trees. Insertion and deletion of points from the index structure can be accomplished in $O(DlogN)$ time for each point, where D is the number of dimensions.

The retrieval operation is designed to efficiently (but approximately) retrieve the k nearest neighbors of a query point. The pseudo-code of this operation is shown in Figure 2.5.

The algorithm takes as input k: the number of desired nearest neighbors, t: the number of candidate neighbors to consider along each dimension, X : the query point and M : the index structure. The output consists of the k nearest neighbors (approximately). The neighbors of the query point is initialized to the empty set (in line 1 of Figure 2.5). Next, the dimensions are enumerated in decreasing order of their weights (line 2) and the nearest t neighbors along each dimension d are retrieved (line 3). This is done by searching for the query point in the list for dimension d in the index structure. This search will retrieve the point closest to the query point.

[0]Our index structure is built on top of the work of Nataraj Jammalamadaka, done for his B. Tech Honours at IIIT Hyderabad [7]

Retrieve k-NN(k, t, X', M):

1 $neighbors = \{\}$
2 for each dimension d (in non-increasing order of weights):
3 $\quad R = t$ nearest neighbors in dimension d
4 $\quad neighbors = k$ nearest neighbors of X' among $(neighbors \cup R)$
5 return $neighbors$

Figure 2.5: Approximate k-Nearest Neighbor based Retrieval

Then, a linear traversal along the list from that point in both directions will retrieve the closest t points. The t points obtained along each dimension are candidate points to be considered for being among the k nearest neighbors of X. These points are compared with the nearest neighbors so far obtained to determine whether they are to be retained in the k nearest neighbor set, or to be discarded (line 4). Finally, the nearest neighbors obtained after enumerating points along all dimensions are output (in line 5).

Complexity Analysis

Consider N, D dimensional points in the database. Insertion is an offline process and can be done efficiently. In the search operation Step 3 is $O(log(N) + D)$ and Step 4 is $O(D)$. Thus the search operation takes $O(DlogN + kD^2)$. Since $log(N) >> D$ we have the complexity of the search operation to be $O(DlogN)$. A more detailed analysis of the index can be found in [7].

The rationale behind the index design

We wanted to use a simple and flexible index structure that can be used for similarity search based on the weighted Euclidean distance measure. During the user session the similarity measure changes rather than the index. The key operation at the time of retrieval is to obtain the neighbors of the query point along each dimension. In our index we just need to reach the closest point in $O(logN)$ and then traverse to enumerate samples. These neighbors do not change with weights. The retrieve operation enumerates dimensions in non-increasing order of their weights. This means that the most important dimensions are enumerated first. This makes it likely that most of the true nearest neighbors are retrieved very early during the execution of the algorithm. This can be advantageous, especially in situations where the user is interested in all neighbors within a specified threshold. Our index and algorithm retrieves only approximately, i.e. the nearest neighbors from a dimensions may not be in the final list at all. We use a large $t >> k$ to compensate for this approximation. Our experiments show that good accuracy is obtained for reasonable values of t. In Section 2.8.2 we validate the approximation accuracy with the help of Table 2.3 and Table 2.4.

Comparison with other approaches

The common approaches adapted for retrieving from large databases are either flat-file based or MySQL indexing based. We compared our proposed approx. kNN approach with these experimentally. We compared the effect of increasing database size and number of dimensions on the retrieval time for the three. We averaged the retrieval time for a set of 5 queries which were randomly picked from the database. The same set of queries were used for all the experiments.

We first compared the effect of database size using a synthetic database of 10 dimensions. As can be seen from the pattern in Table 2.1, MySQL is much faster than the flat-file structure. The difference increases as the size of the database increases. MySQL is slower with very small database

sizes as the overheads involved are more than those with flat-files. But our approach is consistently faster than both flat-files and MySQL. The basic reason for our approach performing better than the other two lies in the inherent similarity ordering of data in our case. Flat-files have no indexing at all so there the retrieval is always exhaustive leading to high retrieval times. We do better in comparison to MySQL, which also has a B tree based indexing scheme because in our B+ trees the feature values for each of the dimensions are inherently arranged ordered on similarity. As a result when we want to retrieve the k nearest neighbors from a dimension (B+ tree), the interesting samples lie in the nearby nodes. This is not so in the case of MySQL where a sort like operation has to be performed on the exhaustive set of samples for selecting the nearest k. The inherent ability of our approach to maintain ordered arrangement within feature dimensions makes it optimal for our similarity based retrieval approach. Table 2.1 lists the average retrieval times for different database sizes for the three approaches.

DB Size	**Flat Files**	**MySQL**	**Approx. kNN**
25K	0.0148	0.0189	0.00721
1L	0.0777	0.0305	0.00757
2.5L	0.2025	0.0473	0.00773
5L	0.4147	0.0587	0.00779
7.5L	0.6325	0.0753	0.00782
1M	0.8730	0.0987	0.00787

Table 2.1: The table compares retrieval times (in secs.) at different DB sizes for Flat-files, MySQL and our approach. The number of dimensions is kept fixed at 10 for this experiment. Flat-files retrieve quickly for smaller sizes owing to the overhead involved with both MySQL and our approach but the trend quickly shifts in favor of our approach. It even reports better performance than MySQL as unlike in MySQL, along the dimensions samples are inherently ordered in our case.

As visible in Table 2.1 retrieval time increases *l*ogarithmically with the database size. Detailed discussion can be found in [7] but since each tree is a B+-tree the average retrieval time varies as the logarithm of the number of samples. We next compare the effect of increasing the number of feature dimensions on the retrieval times for flat-files, MySQL and our Approx. kNN approach. We used a fixed size of the database, *0.1 Million* samples and varied the number of dimensions. Table 2.2 lists the statistics for the experiment.

Dimensions	**Flat Files**	**MySQL**	**Approx. kNN**
10	0.0777	0.0305	0.00757
20	0.1489	0.0630	0.01330
50	0.3657	0.1590	0.03678
100	0.7829	0.3274	0.07100

Table 2.2: Table comparing retrieval times (in secs.) at different number of dimensions for Flat-files, MySQL and our approach The database size is kept fixed at *0.1 million* data points. At low dimensionality, flat-files perform well but the performance deteriorates quickly with increase in dimensionality.

As can be clearly noted from the data in Table 2.2, the retrieval time increases linearly with the number of dimensions in the data. This can be intuitively understood based on the retrieval process. k-NN samples are retrieved from every dimensions B+-tree in logarithmic time. So with addition of more dimensions or B+-tree the retrieval time also increases linearly.

Scalability of content based multimedia retrieval approaches for large datasets of images has not received the due attention. A large body of work exists on the study of index structures for similarity search. These algorithms involve building a spatial access tree, such as an R-tree, k-d tree, SS-tree or their variants. The index structures presented in these papers were novel, elegant, and useful. However they are not applicable in our study as they take a specific similarity measure as input and build index structures tuned for that measure. Recent attempts on this problem focused on scenarios where the similarity measure itself is not fixed, but continuously being refined. These algorithms have taken a branch and bound approach, which may degrade to searching most of the tree structure. We focus on efficiently retrieving the data, with bounds on the time taken, when similarity measure is varying continuously.

2.5.2 Selective Dimensional Retrieval

In this thesis, we further propose that even an exhaustive retrieval from all the feature dimensions is not required. We propose and experimentally validate that using only the most important few dimensions, retrieval accuracy at par with the use of all of them can be achieved. The number of dimensions to be used for a particular query iteration is not fixed. We propose an adaptive scheme for dimensionality reduction which provides tremendous gain in efficiency.

Our formulation uses the feature weights for the present query to estimate the optimal number of dimensions. This problem in our case is posed in a pretty simple manner. Unlike scenarios where inherent manifolds need to be estimated using computationally expensive techniques, our learning scheme provides us a weight vector which favors the more relevant features(dimensions). This allows us to estimate the reduced number of dimensions(features) to be used with a simple change monitoring method.

In our proposed dimensionality reduction approach, we traverse the dimensions in non-increasing order of weights and merge the samples suggested by each dimension into a master-list. For every new dimension we first check if merging the samples suggested by this dimension results in any change in the master list. If not, then we stop the traversal and return the current list as the final result set.

This simple scheme saves a lot of computations especially in later iterations when the learning guided metric is heavily biased towards a few good features. We validate our claims experimentally in Section 2.8.2 using the Figures 2.10, 2.11, 2.12 and Table 2.3 and Table 2.4.

2.6 Image Representation in FISH

Each image in the system is represented as a vector of numeric feature values $X_1, X_2, \ldots, X_D$. The space of possible vectors constitutes a multi-dimensional space in which each image is a point.

The general features used in any image retrieval system are color, texture and shape descriptors. Color descriptors though weak in description, allow flexibility in use through variations ranging from the global histogram to the color layout descriptors. Texture is generally highly dependent on the homogeneity and regularity of the patterns in pixels. Shape is difficult to extract and represent.

We have predominantly used color descriptors for our features. We have experimentally selected a weighted combination of third order color moments and some selected MPEG-7 descriptors [75]. We have chosen to use mean, variance and skew color moments which capture the orders of variations of colors in the image. To incorporate texture information we have included three components from the *Texture Browsing Descriptor(TBD)* in MPEG-7 Standard and also the *Edge Histogram*

Descriptor(EHD). We have also included the *Color Layout Descriptor* and the *Color Structure Descriptor* from MPEG-7.

The *Color Layout Descriptor(CLD)* [84] captures the layout of the colors in the image by incorporating a DCT transform based representation. The color layout descriptor operates in the YCbCr space by breaking up the image into 8×8 or 64 blocks. Then it computes the dominant color for each of these blocks. To gain computational efficiency, we use the average color for this. This forms a 8×8 pixel representative image. Next, we compute the DCT transform of this representative image and compute the 8×8 DCT coefficients for each of the three channels separately. We then perform a quantization of these three matrices one for every channel and select a few of most representative coefficients from the three channels in a zig-zag scan order to make our feature vector. Based on MPEG-7 standards we use the top 6 coefficients from the Y channel and 3 from the other two. We use a weighted combination of these coefficients based on scan order.

The *Color Structure Descriptor(CSD)* [85] is a 32-bin quantized representation. It slides an 8×8 window over the entire image and computes an occurrence histogram for the colors in the image. This captures the spatial layout of the colors in the image. The result is a spatially augmented color histogram of the image.

The *Edge Histogram Descriptor* [86] captures the dominant edge pattern in the image. It looks for four edges namely, vertical, horizontal and the two 45° orientations. The histogram of edges when applied on grid based regions captures the local structure well.

The *Maximum Response Filters(MR8)* [87] filter bank consists of 38 filters but only 8 filter responses. The filter bank contains filters at multiple orientations but their outputs are "collapsed" by recording only the maximum filter response across all orientations. This achieves rotation invariance. The filter bank and consists of a Gaussian and a Laplacian of Gaussian (these filters have rotational symmetry), an edge filter at 3 scales and a bar filter at the same 3 scales. The latter two filters are oriented and occur at 6 orientations at each scale. We use all the 38 responses in combination with the other descriptors mentioned above rather than directly selecting the maximum responses.

The process of feature extraction can be pictorially summarized as in Figure 2.6.

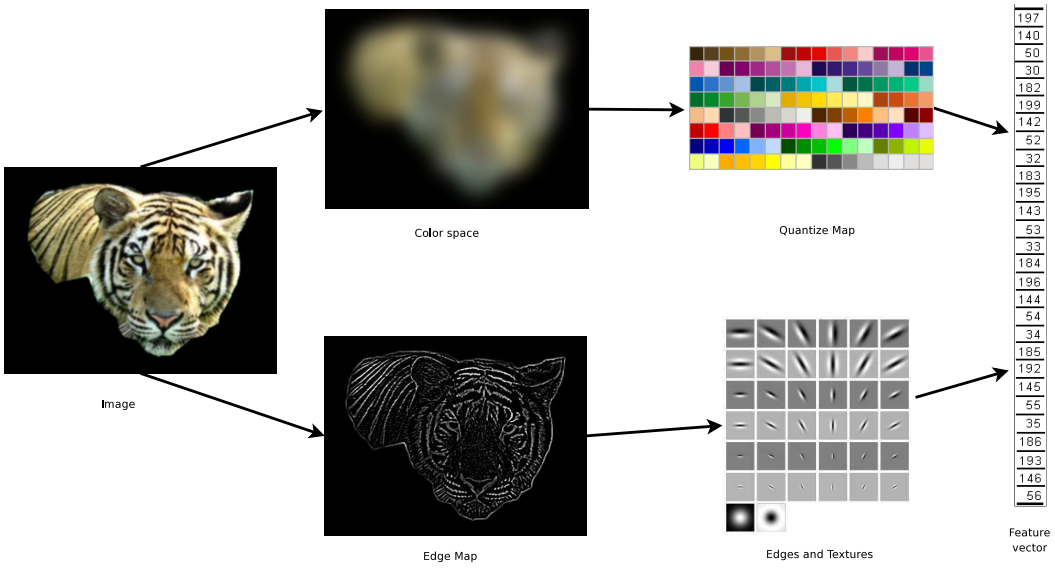


Figure 2.6: Feature extraction in FISH

We use a combination these descriptors for our experiments. The dataset of images is indexed on these features. Feature extraction and indexing on dataset images is done offline while user-queries are processed online in real-time.

2.7 Scalability of Learning Schemes in FISH

Even the most complex features which can be extracted from images are far from satisfactory when it comes to semantic retrieval capabilities. This necessitates external input of semantics into the retrieval loop. As the human user is the best interpreter of visual content, Huang [20] proposed the use of feedback from the user on the relevance of images displayed to him in response to his query. We propose to use a standard relevance feedback approach, where typically. image semantics are inferred as weights for features, using the relevant and irrelevant images [49]. These are then used with the similarity metric for tuning the retrieval. Some proposals also modify the query to better represent the intended concept.

Learning preference for feature dimensions inherently suit our approach, especially in view of our arguments of improved accuracy using only the best few dimensions, in order. We use techniques (discussed in depth in Section 3.4) for estimating the relative importance of a feature j to the query concept based on user feedback as s_j. We use s_j across user iterations for incrementally updating weights $\mathbf{w}$ for the features as in Equation (2.1).

$$w_j^t \;=\; \gamma\, w_j^{t-1} \;+\; \beta\, s_j \tag{2.1}$$

where w_js represent the weights for the j^{th} feature after the $(t-1)^{th}$ and the t^{th} iterations. γ and β control the learning.

This allows us to adaptively learn the user's preference for his intended concept in his query and use them for improved retrieval. We believe that the weights represent the visual concept which made these images semantically relevant to the query. So when the user session is over, we use the weights $\mathbf{w}$ to update our knowledge of the important content in the relevant images, also expressed as weights for the features, $\mathbf{c}$ as in equation,

$$\mathbf{c} = \mathbf{c} \;+\; \rho\, \mathbf{w} \tag{2.2}$$

ρ again controls the learning.

This incremental update ensures smooth convergence of the content learning for images. Our dissimilarity metric used for selecting the top N results for the user biases feature level dissimilarity with query centric weights, $\mathbf{w}$ and image centric weights, $\mathbf{c}$ to retrieved with improved user centric accuracy as in Equation (2.3).

$$d_i = f(x_i,\, q,\, w,\, c_i) \tag{2.3}$$

We discuss learning approaches in detail in Sections 3.4 and Section 3.5. This brief discussion on how FISH incorporates learning can be expressed compactly as an algorithm as below.

New images inserted into the database are effectively included by absorbing their first session's weights as their long term learning. Experiments discussing the benefits of incorporating learning in the **FISH** framework are discussed in detail in Section 2.8.2

2.8 Performance Study

In this section we present our performance model for validating our claims on our system. Our experiments fall into four categories, namely, (1) System, (2) Relevance feedback, (3) Long Term Learning, and (4) Optimizations.

Require: Query as a feature vector q
Ensure: List $\mathbf{R}$ of images similar to the query as per user intent
  for all Iterations, t, of the user **do**
    if $t = 1$ **then**
      Initialize $\mathbf{w} = \{1.0\}$, no-preference
    end if
    Initialize $\mathbf{R} = \{\}$
    Sort $\mathbf{w}$
    for all Dimensions, d , in order of decreasing $\mathbf{w}$ **do**
      Retrieve k-NN members from q in dimension d
      Merge them into the master set $\mathbf{R}$ to have N closest members using Equation (2.3)
    end for
    Present top $\mathbf{R}$ to the user for feedback
    Use Equation (2.1) to learn $\mathbf{w}$
    Re-initialize $\mathbf{R}$ and repeat the process as long as the user iterates.
  end for
  Use Equation (2.2) to update long term memory

Algorithm 1: Retrieval with changing similarity metric with feedback and weights

The *system* related experiments validate the performance claims in terms of scalability and response times. Next we discuss a series of experiments which showcase the *accuracy* of our system as a result of effective and efficient use of relevance feedback from the human user. We show our improvements in performance using *precision* as the main performance metric. Following it we present a series of results showcasing the effectiveness of the proposed framework for inter-query or long term learning through precision gain. In the last part, we present some results on the optimizations we have incorporated in the system to improve performance. We experimentally validate our claim on dimensionality reduction by restricting the comparisons only to the top few dimensions rather than exhaustively across all the features.

2.8.1 Datasets

We have used two types of datasets for our experiments. For the first of these datasets, we collected around 12000 real images by mixing the Corel dataset, images crawled from Flickr with research permissions, Caltech datasets and some other freely available small collections. This set is manually annotated. The data set is a mixed set of 58 concepts like flowers, trains on tracks, horses in the fields, cars, buildings, mountains, surfers in sea, ships at sea, meadows, airplanes, guns, cycles, buses and animals. The selection of samples does not showcase any inherent visual discrimination among classes. We used this annotated set for all our accuracy validation experiments.

For our system related experiments we populated a huge dataset of *1 Million* feature vectors. Approximately $400,000$ are above features extracted from real images collected from the above sources while the rest were synthesized.

The experiments involving response times and scalability were performed on a machine powered by Intel Xeon 1.6 Ghz processor and 8 GB RAM, running on Fedora Core 5 64-bit operating platform.

2.8.2 Experimental Results

Scalability and Speed Scaling of the database in terms of the number of samples, feature dimensions and user sessions all add a computational cost. As expected, this leads to slow response for the query. Through our experiments, we show that our index structure efficiently adapts to the scale-up without much change in response time characteristics. The change in response to an increase in the number of samples in the database is very gradual and low (as shown in Figure 2.7). The response time in this graph has been averaged over a sets of 5 queries picked randomly from the synthetic dataset of 10 dimensions. They were used to estimate average response after insertion of every 25000 samples till all the *1 Million* samples were inserted.

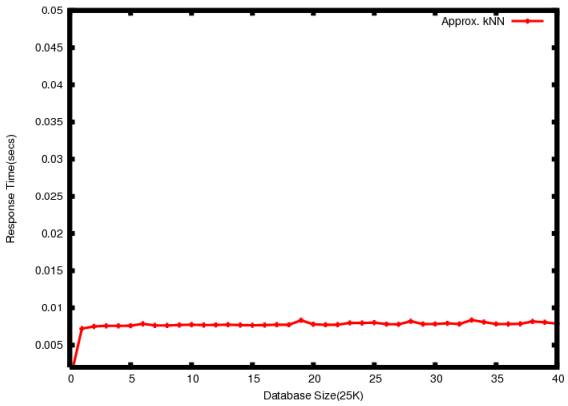


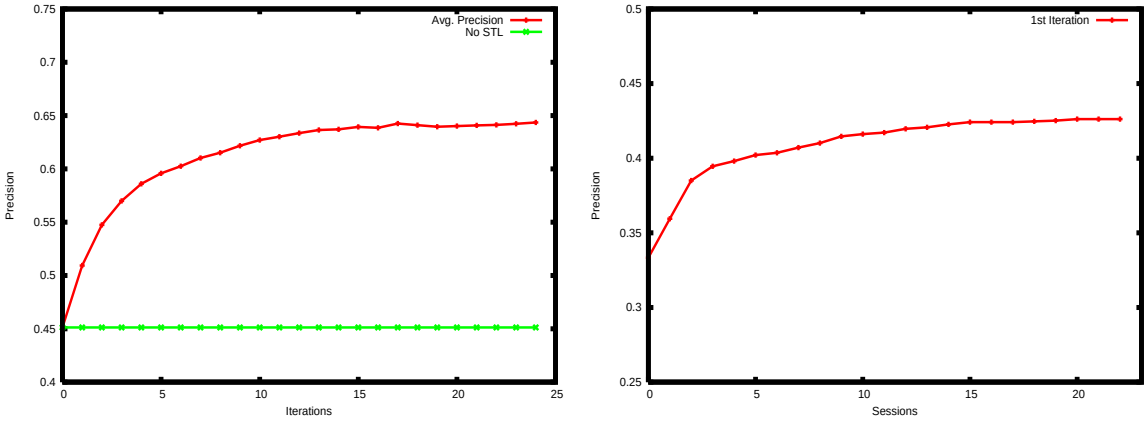
Figure 2.7: Plot shows the avg. retrieval time with increasing database size for our Approx. kNN approach

Scalable Learning in FISH Figure 2.8 shows the improvement in accuracy, measured as average precision, with subsequent feedback iterations over the same query. The experiment was run on the 12,000 real image dataset (described in Section 2.8.1). A set of randomly picked 25 queries was used for averaging accuracy for all iteration. In each iteration the top 48 results received feedback. The simulation ran for 25 iterations. As is evident from the sharp gain in accuracy over the initial few iterations, the system readily absorbs relevance feedback from the user and iteratively tunes the retrieval to the his intent. The slope reduces over iterations and finally flattens out showing convergence of learning.

We next used the same 25 queries and measure average precision across sessions. Each query was treated with 25 sessions of synthetic feedback of 5 iterations each. The precision for the first iterations in each session was averaged across queries and is plotted in Figure 2.8. The commendable gain in accuracy can be seen in the figure.

We have also included results for a query presented to **FISH**. The rows of 'train on track' images shown in Figure 2.9 show retrieval and it's improvement with feedback.

Performance Optimizations and Improvisations The basis of all our claims on performance and effectiveness is the belief that the data is clustered along only a few features for any given category of images. This is best expressed by the weight bias learned by the system towards a few of the features. We conducted an experiment observing the feature weights over iterations. We show the results in Figure 2.10 for only a couple of randomly picked samples due to space constraints. The plots clearly show that the initially-unbiased weights curve reaches a highly skewed bias in favor of a few features in the first few iterations itself.



(a) (b)

Figure 2.8: Plot (a) shows the improvement in average precision with user feedback over iterations for the same query. Plot (b) shows how the average precision for the first iteration improves using long term learning in FISH

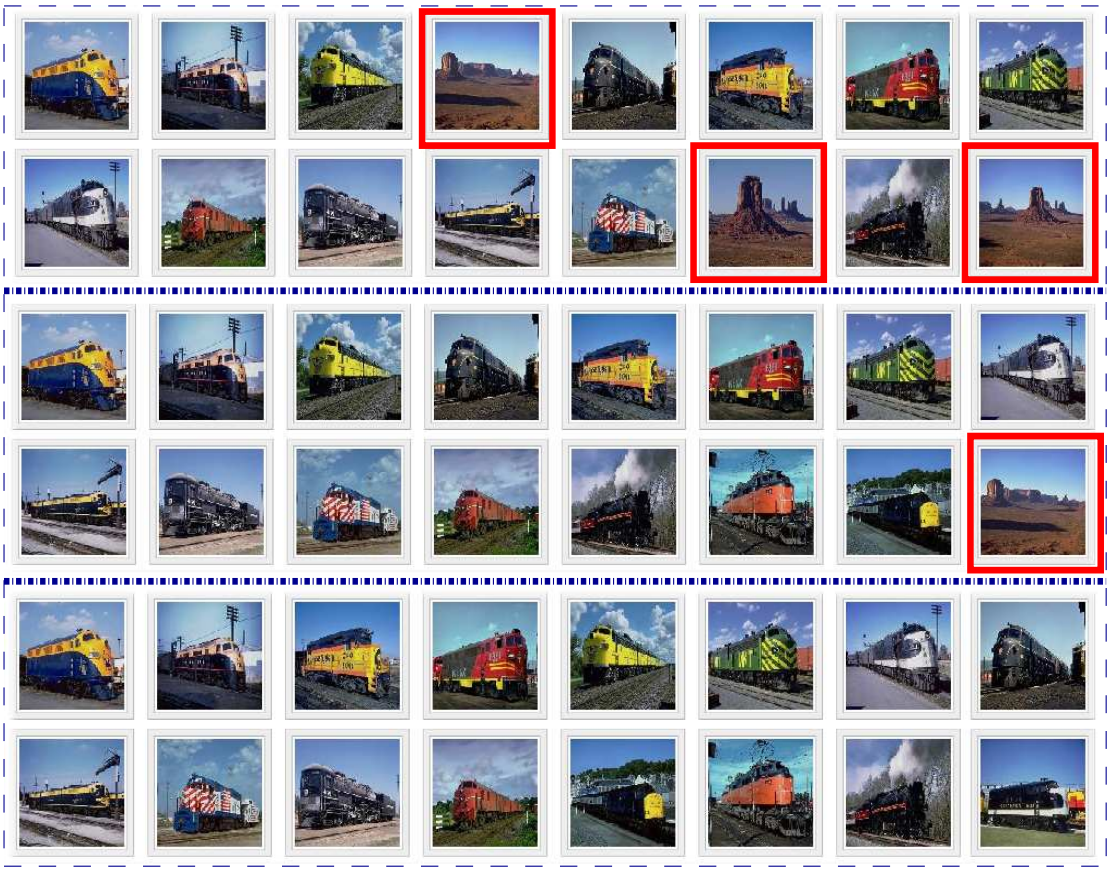
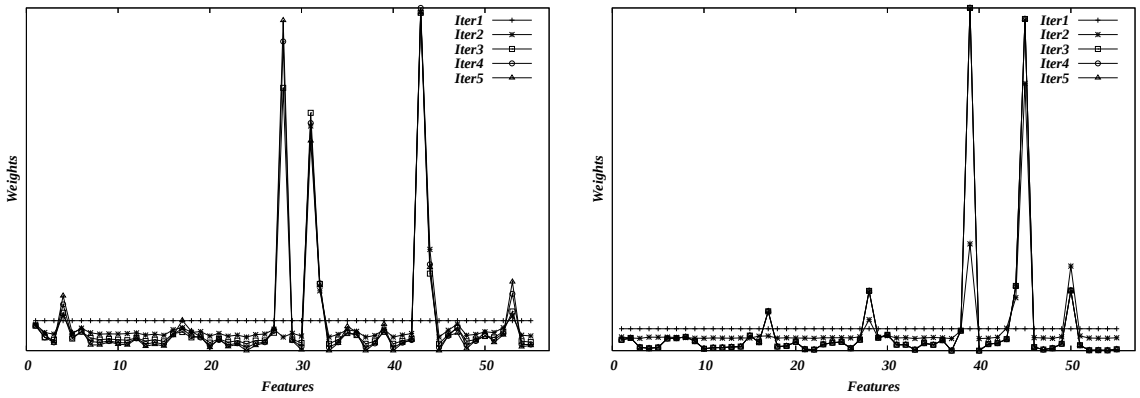


Figure 2.9: Set of ranked results returned by the System in first iterations of the 1^{st}, 2^{nd} and 3^{rd} sessions (Irrelevant results highlighted). The improved relevance of the set is evident from the results. As can be seen the rock images(false positives), recede with feedback based learning.



(a) (b)

Figure 2.10: Plot (a) shows how a few of the features become much more important than others with feedback iterations. Plot (b) shows another example of skewed feature weights as iterations progress.

We gain tremendous efficiency with a light trade-off in accuracy by performing an *approximate* $k-NN$ retrieval. We achieve this by retrieving only a small set of samples independently from each of the feature dimensions and merging them to get the final set. The graph in Figure 2.12 shows that our approx. $k-NN$ approach closely matches the accuracy of the exhaustive approach. It also shows that the approximation improves with iterations, as learning biases the retrieval towards the more important dimensions. The graph in Figure 2.11 shows that when we retrieve in non-increasing order of importance of dimensions the number of relevant samples added to the final set decreases and finally stops after the first few dimensions. This means that we can reach reasonable accuracy even if we retrieve only from the top few most important dimensions in any iteration for the query. The graphs in Figure 2.11 and Figure 2.12 further our claim in Section 2.5.2 that even an exhaustive search spanning all the features is excess. We present two table here validating our claims.

Top 1	Top 2	Top 4	Top 5	Top 10	All
Iter 2	42	48	49	51	53
Iter 4	43	50	51	54	55
Iter 5	44	51	54	56	57
Iter 10	49	53	56	58	58

Table 2.3: The table shows the comparison of precision achieved using only a few top dimensions with precision achieved using all dimensions. As seen, precision using just the top few dimensions matches that achieved using all the dimensions. The pattern of learning based improvement in precision further strengthens this argument. Recall was constant at 100.

Table 2.3 shows the precision Vs iteration values using increasing number of features. The other Table 2.4, presents the response time Vs iteration values for the same scenario. As can be seen from Table 2.3 the accuracy with very few of the most important features nearly matches that achieved by using all the features while at the same time the gain in response time is tremendous as can be seen in Table 2.4. This tremendous gain in response time for a possible yet negligible loss in accuracy showcases our claims on the proposed method for dimensionality reduction.

The results discussed above validate our claims on our optimized approach for approx. $k-NN$

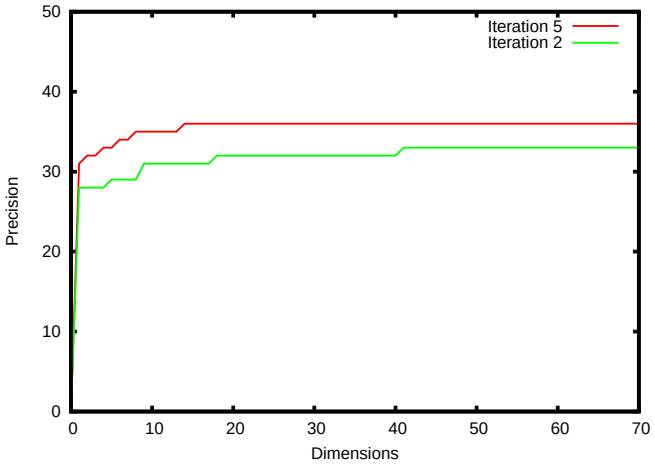


Figure 2.11: Graph shows how in decreasing order of weights every dimension adds fewer relevant samples. The graph also shows that the trend becomes more and more visible as iterative learning improves.

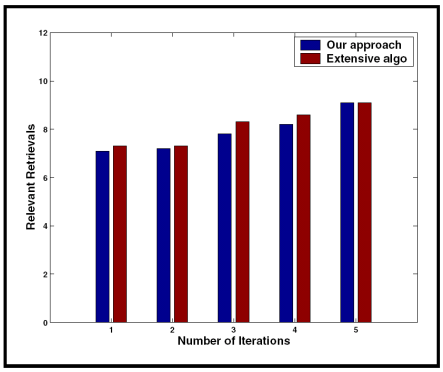


Figure 2.12: The chart shows how across iterations the number of relevant samples retrieved by our approx. approach are nearly same as that retrieved by the exhaustive approach. We evaluated the top 30 retrieved samples for relevance in each feedback iterations. As can be noted from the chart, the approximation error becomes negligible over iterations as a result of better learned dimensional ordering, using feature weights.

Top 1	Top 2	Top 4	Top 5	Top 10	All
Time	1.3 ms	2.4 ms	6.1 ms	12 ms	51 ms

Table 2.4: The table shows the gain in retrieval time achieved when only some of the top dimensions are used instead of the exhaustive set. The pattern of gain in retrieval times hold true across iterations, corresponding to the pattern in the precision in Table 2.3

retrieval using B+ - trees for dimensions. They also show how learning based improvement in accuracy can be achieved effortlessly with **FISH**.

2.9 Implementation of FISH

We designed **FISH** as a user centric system. This required **FISH** to support efficiency, accuracy, interactive response, and user friendliness. We also wanted **FISH** so serve as a platform for researchers working on different aspects of image retrieval to be able to use it to apply, test, prove and improve their algorithms. Keeping these objectives of consumption we designed **FISH** using the some of the most simple and commonly used building blocks. We list them in briefly below for reference.

- **FISH** has been completely developed using open source software. It confers to the LAMP philosophy of software design. It has been developed as an open source platform available for research purposes. It extensively uses C/C++, Perl, PHP, HTML and Apache for development. In order to achieve goals of easy extensibility OOP principles have been adhered to as far as possible. We use MySQL database for all our archival and concurrency needs.

- **FISH** has a web interface which is capable of accepting queries in most of the common image formats like JPEG, TIFF, GIF, PNG, PPM, PBM, PGM. It supports numerous user friendly features like query-by-uploaded-example, pagination, click based navigation and relevance feedback.

- **FISH** uses a secondary memory algorithm for managing the B+-tree based indexing scheme. We have adopted the implementations available in the TPIE toolkit developed at the Duke University. It allows FISH to efficiently handle its own memory requirements. FISH design and development are completely independent of TPIE, we chose it for its simplicity and known worthiness.

- **FISH** has been developed as a modular framework. Based on research goals in CBIR community it can be considered as being composed of three macro modules, the interface, the retrieval system and the learning system. Each share a clear interface with the others and can be replaced with minimal impact on the rest. Micro modules inside each of these macro ones can also be easily experimented with.

- **FISH** has been designed to seamlessly extend to distributed querying and indexing. The retrieval module communicates through the network with the learning and the interface. The interface can also schedule queries or parts of the queries (distribute dimensions) across machines and seamlessly aggregates the results. This further enhances the capability of interactive retrieval.

- Long term learning of image content is controlled to guard against rogue feedback. It is archived in MySQL and the updates are designed as offline deferred processes to avoid any visible impact on retrieval time.

We have included many such trivial yet effective features in **FISH**'s design in order to achieve appreciable and reliable performance.

2.10 Limitations of FISH

Though **FISH** is extensively designed, yet it does have some constraints. We shall review them at length here.

Semantic Gap
: In **FISH** we have incorporated simple but advanced methods of bridging the semantic gap between the low level visual features used for representation and the high level perception of content by human users. But even with the most advanced tools for learning driven retrieval we still are dependent to some extent on the features. The learning can improve the accuracy but only if the features capture some basic semantics. In cases of no semantic relation between the data and representations even the most advanced feature based systems tend to fail. We would like to bridge this gap by feature independent approaches which are left as a future enhancement beyond the purview of this thesis.

Constrained to the use of Low Level Features
: In **FISH** we have designed optimized indexing, retrieval and learning schemes keeping the low level feature relevance in mind. The system maximally utilizes the low level information and performs retrieval comparable to that with high level semantic information like tags, regions etc. But it will require modifications to the core scheme of things to adapt it to handle higher levels of features which are not combinations of the low level ones, like regions, tags, relations etc. Some of these can be incorporated with minimal modifications and are left as future enhancements to **FISH**.

Query by Content with Text
: **FISH** does not have the capability yet to query by text tags over a database of visual features. It can only query given an image as a query. The capability to convert a text query into visual concepts or query will be an interesting pursuit for a future version.

Resource intensive
: **FISH** was not designed to be used as a desktop application. The goals were efficient, accurate interactive retrieval from large databases. The system is both processor and memory intensive. The secondary memory management improvises the query response but it is still intensive on processor and memory at that instance. It can be scaled down to perform as a desktop utility but not without compromising on basic design goals.

Concurrency is difficult
: In the present version of **FISH** concurrent queries are not handled. This constraint comes from the TPIE toolkit as it disallows multiple interactions on the same ports. We tried to threads for separate queries but only with limited gain in concurrency.

URLs not supported
: In **FISH** the goal was to allow the users to experiment with the system by testing with their

own queries and those in the database. So we provided the option of uploading an image by browsing your local machine. Querying by pasting an image URL was not provided in an effort to reduce the complexities on the interface.

FISH definitely has some constraints which should be factored in by users and researchers alike. But apart from a few like low level feature dependency others are open enhancements which we believe will improve usability and utility without much impact from the algorithms and research perspectives.

2.11 Summary

We have developed and presented a system for interactive image retrieval. Our framework seamlessly scales to huge databases. It uses relevance feedback for improving intra and inter query retrieval. Our interface optimizes the interaction by minimizing the load on the user. We have shown that there is minimal if any loss of accuracy with our approx. k-NN approach and the performance to accuracy trade-off is in favor. We have experimentally validated all our claims on performance while reiterating the belief that though feature dependent, retrieval improves commendably with *short* and *long* term learning. We have also validated our claim on more efficient retrieval using learning and how feature relevance based improvisations benefit efficiency and accuracy. The system has been designed and developed on the principles of modularity allowing for effortless modifications.

Chapter 3

Feature Relevance Learning

3.1 Introduction

In CBIR, typically, an image is represented as a feature vector, and retrieval is done by finding the most similar images in the database. Especially, in the early years major research efforts focused on feature identification and expression for the best representation of the content in images. Notable contributions were made to effective representations using colors, shapes, and textures. Researchers then, typically adopted the "machine centric" mechanism to design the CBIR systems. In the "machine centric" CBIR systems, users are first required to select the features they are interested in and then also specify the weights of the features in accordance with their preferences. The CBIR systems then, return the most similar images by ranking on similarity based on the selected features and their corresponding weights. However, those early CBIR systems with heuristic feature selections and fixed weighting schemes did not achieve satisfactory performance. Later, researchers noticed and recognized the challenges in CBIR, as the semantic gap between high-level concepts and low-level features, and the subjectivity of human perception.

For example, consider the flower in Figure 3.1. The challenge before CBIR researchers is of designing algorithms which can automatically identify and extract the popular content from any given image. This can then be used for improving retrieval accuracy. In this example, it is the yellow flower and not the hedge which is more popular, among most of the users who saw this image. "Machine centric" CBIR systems are not powerful enough to capture the high-level concepts and

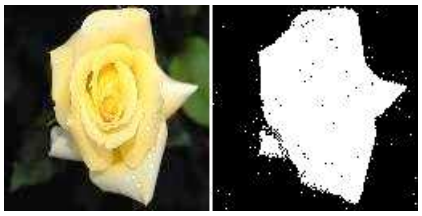


Figure 3.1: (a) A flower in the hedge and (b) The learned content

overcome the subjectivity of human perception.

In order to overcome the limits of the early "machine centric" CBIR systems, human interaction was brought into the retrieval process in an interactive mechanism in recent CBIR research [19]. Under this circumstance, relevance feedback was introduced into CBIR as a powerful technique to attack these challenges. Relevance feedback comes originally from traditional text-based information retrieval, in which it has been proved as an effective technique to improve the retrieval

performance. The CBIR system first displays the user a few image samples simply employing some kind of similarity metric. Based on the initial returned results, the user marks labels on the images to indicate which of them are relevant and which are irrelevant. A relevance feedback module is employed to refine the user's query concepts based on the feedback and return a set of better results to the user. By engaging in such a relevance feedback mechanism, the user can retrieve his/her desired images round-by-round through the interactions with the computers.

Various aspects of feedback have been extensively studied in literature [49]. They vary from techniques which change the feature metric or the feature space towards more important features as in [88] to those which try to improve feature based classification as in [89]. These approaches primarily improve the performance for the present query only and are better known as *Intra-query* or *Short Term Learning* approaches. Though most of the approaches discard the expensive and invaluable feedback from the user after every query session, some researchers have explored learning from one query to benefit the subsequent ones. This category of techniques is known as *Inter-query* or*Long Term Learning methods*. Most of these approaches rely information processing of logs of user feedback [89], while some use computationally intensive techniques like factorization (LSI) and Neural Networks etc. [77, 90, 91]. Although, a powerful tool, use of relevance feedback has limited popularity. The poor scale-up of techniques with volume of data and users and over reliance on logs etc. can be considered the prime reasons.

3.2 Related Work

Relevance feedback has been long advocated as a means to resolve query ambiguity through user interaction. Short term learning as a filed studies the utility of relevance feedback for the active query. Probabilistic methods are the most common, where conditional probabilities model the next retrieval sequence. People have also explored methods which modify the query to a virtual point based on feedback. Feature weighting to favor the more relevant features and suppressing the non-relevant ones has been extensively researched in literature. Classification based approaches have become popular recently. Density estimation approaches like GMMs have also been considered. Of late hybrid approaches as combinations of many simple ones are being proposed for handling complex queries. Table 3.1 summarizes the trends in relevance feedback based short term learning research.

As per literature, inter query learning can be classified into two. First are the term-document methods of relationships between retrieval sessions and second are, methods based on feature vector models. The first assume on similar feedback patterns for similar queries. This assumption is extended to considering that similar feedback for images means they are semantically similar. The second, change the scales of vector co-ordinates, thereby bringing similar images closer. These use methods ranging from weighting schemes to kernel operations for transforming the feature space according to the feedback. Table 3.2 summarizes the two approaches.

Core Technique	Paper	Contribution
Probabilistic Methods	Cox et al. [62], Vasconcelos and Lippman [64], Yin et al. [92]	Use probabilistic models in relevance feedback to capture the information from the retrieval pattern
Query Point Movement based	Celentano and Di Sciascio [93], Muller et al. [94], Huang et al. [95],Rui et al. [96]	Move the query point closer to the relevant and farther from the non-relevant images.
Re-weighting based	Squire et al. [97], Huang et al. [95], Rui et al. [20], Fournier and Cord [98], Aksoy et al. [99], Brunelli and Mich [100], Ishikawa et al. [101], Doulamis and Doulamis [102], Schettini [103], Wu [104], Minka and Picard [105]	Weight the important features more, even using multi-layer feedback and self organizing maps.
Classification based	Vapnik [106], statistical discriminant analysis [59], Muller et al. [107], Hong et al. [108], Zhang et al. [109], Tao and Tang [110], 1-SVM [111], Xie and Ortega [112], Wu et al. [51], Srinath et al. [113], Wu et al. [114], Zhou et al. [59], Wu et al. [104], Tao et al. [115], MacArthur et al. [116], Guo et al. [117], Tieu et al. [118], Qian [119]	Methods use SVMs, kernel based SVMs, statistical discriminant techniques, decision trees, boosting and even RBFs for classifying the images are relevant or non-relevant.
Density Estimation based	Nastar et al. [120], Meilhac and Nastar [121], Dong et al. [122], Najjar et al. [123], Yoon et al. [124], Qian et al. [125]	Density estimation based approaches, which generally assume that the positive samples mostly follow a uni-modal Gaussian distribution. GMMs, Parzen windows, and EM have also been explored.
Hybrid methods	Yin et al. [92]	It was argued that not one single scheme but combinations of different relevance feedback techniques are required for most retrieval situations.
Region based methods	Jing et al. [126], Jing et al. [127], Jing et al. [128], Jing et al. [129], [130]	Proposed the use of relevance feedback for fine tuning queries comprised of region-based visual features.
Manifold based methods	Zhou et al. [131], He et al. [132], Jin et al. [133]	Recently, manifold ranking has also been applied to relevance feedback in image retrieval systems but the approach is computationally expensive.

Table 3.1: A summary of the development of relevance feedback over the last decade.

Core Technique	Paper	Contribution
Basic Term-Document Retrieval Pattern based	Fournier and Cord [134], Dong e al. [122], Nastar et al. [120], Srinath et al. [113], Heisterkamp [135], Koskela et al. [136], He et al. [137], Chen et al. [138], Li et al. [139], Hoi et al. [140], Tomo et al. [141], Gosselin and Cord [142], Gosseling et al. [143]	This method applies the term-document retrieval model to the different query session. However, it does not analyze and explore queries which are indirectly related to each other.
Intermediate Term-Document Retrieval Pattern based	Han et al. [71], Han et al. [144], Yin et al. [145]	This approach often analysis the relationship further using a more abstract format. It tries to construct the semantic relationship, also known as the hidden relationship, between each query.
Hierarchical Term-Document Retrieval Pattern based	Jiang, Er and Dai [146]	This extends the second one by organizing into a structured format. This structure can often be used to analyze the more complex relationship between each query session.
Feature Vector Model based	Laaksonen et al. [147], Chan et al. [148], Gondra and Heisterkamp [149], Gondra, Heisterkamp and Peng [150]	SVMs have dominated feature vector learning approaches. Self organizing maps and Tree-Structured SOMs have also been used for feature weighting. They lack adaptability to the users needs.

Table 3.2: A summary of the development of long term learning techniques over the years.

3.3 Feature Relevance Learning in FISH

In this chapter, we explore the possibility of a long term learning scheme which can seamlessly merge with the traditional CBIR with relevance feedback. We would like to retain (or use) the valuable user inputs obtained through user feedback, to learn across sessions. Such a learning scheme should be: (i) Incremental in nature. It should improve with every session, but with minimal computations. (ii) It should be content-specific. Learning should result in enhancing the image representations and thereby reducing the semantic gap. (iii) Transparent to the retrieval. Such a learning should be transparent to the retrieval process. Retrieval should be possible even when the content is not learned completely.

Our solution is motivated by some of the simple, but powerful techniques in text retrieval. While indexing text documents, not all words are found to be equally relevant or useful for the retrieval tasks. There exist effective techniques for capturing the key words, given a collection of documents [151]. Borrowing this idea, we find a set of features which could be informative for the retrieval task. From the pattern classification point of view, these are the more discriminative of the features. However they are calculated with computationally efficient techniques which avoid factorizations and eigenvector computations. In addition, when the retrieval takes place (with or without relevance feedback), the relevance of the features to a specific image is learned, across sessions, to further refine the keywords (or discriminative features).

One of the key advantages of our work is that, it allows to discover the content in the images over time. Given an image of a flower in a garden, our approach allows to learn the pixels corresponding to the flower without any explicit segmentation or user interaction at the pixel-level, provided that multiple users have retrieved this image, while searching for the flower. Figure 3.1 gives an example of the content learned over time. The brightness of the pixel is proportional to the utility of the content.

In brief, when a query is presented to the user in a traditional CBIR, it is processed to extract low level features. Then a retrieval system compares this query feature vector with all the feature vectors stored in the database. It then selects the few most similar images and presents them to the user. In our approach we encourage the user to provide us with feedback on the relevance of the images presented to him. Based on this feedback we estimate the relative importance of the features to the user query. At the end of every query session We also use it for learning the importance of the features to the images which were relevant to the query. This is done under the assumption that these images are related semantically to each other and the query by these feature weights. We then use both these weights in our dissimilarity metric to favor the more important features and retrieve more accurately. We incrementally learn the importance of the features to the query over iterations which can be continued on user's interest. We memorize the learning for the relevant images incrementally to benefit subsequent queries. Our incremental method is resource unintensive but we still use deferred updates to ensure interactive user experience.

In the rest of this chapter we shall discuss our adaptations of some of the existing useful approaches for absorbing relevance feedback. We will then discuss our novel incremental, inexpensive approach for learning and refining the discriminative features in the images.

3.4 Learning the Query Concept

Given the user feedback on the relevance of images presented to him by the system in response to his query, numerous techniques have been proposed in literature for estimating his intent. This is generally accomplished in terms of weights or preferences for some of the features in feature

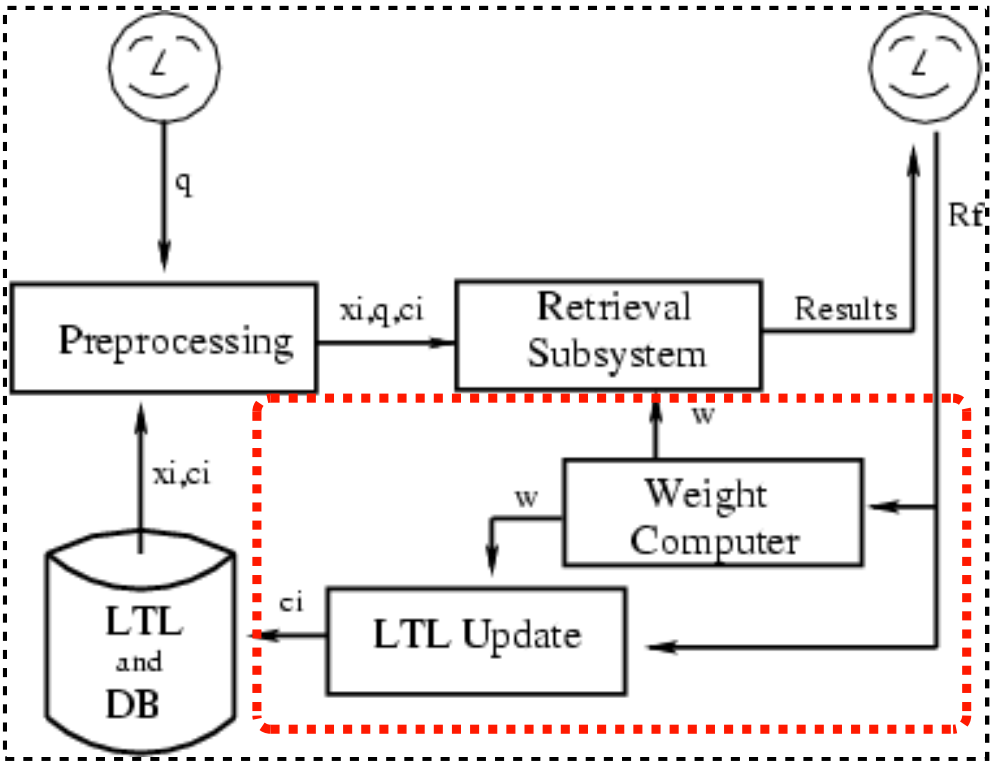


Figure 3.2: Learning in retrieval with the portion of the pipeline benefiting from our contribution being highlighted in color

based image retrieval. Other techniques reviewed earlier use it in methods like spectral analysis etc. In our proposed efficient model for feature based image retrieval our aim was to select a scheme which would best capture the user intent in terms of the features and yet not be computationally prohibitive. Additionally it must seamlessly integrate with our indexing scheme which performs better with more accurate estimates of feature preference. In view of these requirements we explored some schemes from early feedback literature. We briefly discuss here in order to understand the core idea and the associated issues with such approaches. This we believe would help in a better understanding of our proposed approach which we discuss in the following section.

We in general use $\mu_{j,p}$ and $\mu_{j,n}$ to represent the means of the positive and the negative samples for the j^{th} feature and $\sigma_{j,p}$ and $\sigma_{j,n}$ their variances. The x_j^is denote the corresponding features for the ith sample. n_{p} and n_{n} denote the relevant and the irrelevant subsets of R.

Delta Mean

$$s_j = \frac{|\mu_{j,p} - \mu_{j,n}|}{\sigma_{j,p} + \sigma_{j,n}} \tag{3.1}$$

If p and n sets are separated it returns a high score for the feature as discussed in [50, 152]. It fails if the unimodal constraint on the distribution in the p and n sets is violated as generally happens over the n set.

Inverse Sigma Here only the p set is constrained and the peaking of the p distribution is valued as in [50, 152]. It fails to utilize the n set.

$$s_j = \frac{1}{\sigma_{j,p}} \tag{3.2}$$

Discriminative Variance This also constrains both the p and n sets but adds a discriminative edge to *Inverse Sigma* above, also discussed in [50, 152].

$$s_j = \frac{\sigma_{j,n}}{\sigma_{j,p}} \tag{3.3}$$

KL - Divergence This represents the entropy based dissimilarity between the relevant (n_p) and negative (n_n) distributions, discussed in [152]. The *Asymmetric* form is directional in nature.

$$s_j = \sum_{i=1}^{N} (p_{i,j} \log \frac{p_{i,j}}{qi,j}) \tag{3.4}$$

The Symmetric form can be defined as,

$$s_j = \sum_{i=1}^{N} (p_{i,j} \log \frac{p_{i,j}}{q_{i,j}}) + \sum_{i=1}^{N} (q_{i,j} \log \frac{q_{i,j}}{p_{i,j}}) \tag{3.5}$$

Query Point Movement (QPM) This technique discussed in [50], iteratively modifies the query vector to be a weighted average over the relevant images.

$$q'_j = \frac{1}{n_p} \sum_{i=1}^{n_p} * x_{ij} \tag{3.6}$$

Most these schemes result in estimating the relative importance s_j which can then be used for improving the retrieval accuracy. Some, QPM, modify the query directly rather than biasing the metric. Most of these techniques have been adopted from text literature for presenting a better result set to the active user by reducing the semantic gap.

3.5 Discriminative Long Term Learning

In CBIR, an image is represented using a feature vector of automatically extracted visual characteristics. Let $\mathbf{x_i} = [x_{i1}, \ldots, x_{ij}, \ldots x_{iN}]^T$ be the feature vector corresponding to image i in the database. When an example image (query) is given, its corresponding feature representation $\mathbf{q}$ is computed, and a set $\mathcal{R}$ of images with minimal distance ($d(\mathbf{x_i}, \mathbf{q})$) to $\mathbf{q}$ is treated as optimal for retrieval. It has been argued that not all feature dimensions are equally useful for the distance/similarity computation. Relevance feedback based approaches estimate the importance of features to the query concept in terms of weights for each dimension. This relevance ($\mathbf{w}$) is obtained through continued user interactions. This is then used in the distance computation $d(\mathbf{x_i}, \mathbf{q}, \mathbf{w})$.

The success of relevance feedback comes from the fact that not all features are relevant for a given query. However, a relatively unnoticed fact has been that not all features are helpful in characterizing the semantic content of a given image. For example in Figure 3.1 (a), the yellow

flower is the useful (or popular) content rather than the green leaves around it. We argue that such content can be automatically characterized from the history of interactions, and there after used in image retrieval.

Let $\mathbf{c_i}$ be the relative importance of features of image i. Then a better semantic similarity can be computed as

$$d_i = f(\mathbf{x_i}, \mathbf{q}, \mathbf{w}, \mathbf{c_i}) \tag{3.7}$$

There are two possible clues which could allow us to build an estimate of $\mathbf{c_i}$: (a) the similarities and dissimilarities of images within a database (b) Past user preferences in terms of acceptable and non-acceptable images to a given query. Given a collection of images, there is some amount of inherent information in it describing the content in the constituent images. The features which are prominent in one image and those that are not prominent in other images would be more relevant and should be weighted higher while computing the semantic similarity. Such an estimate of the relevance of features to images can be *a priori* computed and used for image retrieval, as $\mathbf{c_i}$ for the i^{th} image. In the case of relevance feedback, user feedback results in two sets of images: relevant and irrelevant images. The goal is then to emphasize features which selectively prefer relevant images and then remember and reinforce them for the future sessions. This relative importance is captured well by the consistency of features across the relevant samples, and discriminability of irrelevant examples from relevant examples. The consistency in features is characterized by their relative low variance. Therefore the idea is to emphasize those features which show high consistency over the relevant set and high variance over other images.

There are two possible modes in which image retrieval typically takes place. In the first category, a query (text or image) is given and a set of relevant images are retrieved. User accepts (selects) some of the images and thus gives an indirect feedback. The second popular approach is to use relevance feedback and along with query-by-example. In this case, user gives explicit feedback to identify positive and negative images. Both these methods can be understood as a process of splitting the retrieved images $\mathcal{R}$ into two subsets $\mathcal{P}$ and $\mathcal{N}$. Literature [88, 49] then talks about many different ways of accomplishing this goal. These methods generally try to capture the utility of a feature based on its uniqueness to the relevant set $\mathcal{P}$ as compared to the irrelevant set $\mathcal{N}$.

We argue that a ratio of the inverse of the variance (or any other similar measure of dispersion) of a feature over the relevant set to the inverse of its variance over the irrelevant set captures the utility of the feature for the retrieval task. Given a set of relevant and irrelevant images, such a measure could be computed for each individual feature as

$$s_j \;=\; \left[\frac{\sigma_{j\mathcal{N}}}{\sigma_{j\mathcal{P}}}\right] \tag{3.8}$$

where σ captures the measure of dispersion. Here s_j is only an instantaneous estimate of the importance of the feature. It changes with user feedback. Note that this is computed for individual features.

$$\sigma^2_{j\mathcal{N}} = \frac{1}{|\mathcal{N}|}\sum_k (x_{kj} - \mu_j)^2$$

where $\mathbf{x_k} \in \mathcal{N}$. One could also think of using other similar measures. A similar approach has been shown to be promising in text retrieval research. There, the idea has been to select key words [151], i.e., the terms which selectively or dicriminatively describe the current document with respect to others.

In addition to selecting such key words, there is another aspect which is very relevant to the text processing community. It is the removal of *stop words* from text. Stop words are words which

are common in a majority of the documents and lack any descriptive capacity. They could come from the *a priori* information coming out of languages or the statistics/distribution of words in the database. In images, these correspond to features which show similar variation over all images and thus should be de-emphasized by the formulation with there weights ideally set close to *zero*. This can be efficiently accomplished by using a modified formulation where we take the *logarithm* of the ratio of inverse of variances discussed earlier. The *logarithm* ensures that the features which show similar variation over the relevant set and the other images are weighted to zero. The modified formulation can be presented as:-

$$s_j \;=\; \log\left[\frac{\sigma_{j\mathcal{R}}}{\sigma_{j\mathcal{P}}}\right] \tag{3.9}$$

First we explain, how the relative of importance of features $\mathbf{s}$, which gets accumulated over iterations, can be used for computing the weight $\mathbf{w}$ in a relevance feedback framework, and then how to incrementally use them for computing $\mathbf{c_i} = [c_{i1}, c_{i2}, \ldots c_{iN}]^T$. The estimate of the importance of the feature s_j can be used for incrementally updating the weight, w_j, for the corresponding feature as in the equation below, which can then be used for tuning the comparison metric,

$$w_j^t \;=\; w_j^{t-1} \;+\; s_j \tag{3.10}$$

where w_j represent the weight for feature j after the iterations t and $(t-1)$. One could also add a learning rate to control the rate across iterations. Here we had shown how the relative importance can be captured within a user session. In a non-iterative mode, this relative importance can be directly computed as we explain below. However, our objective is to use these weights for computing the content vector $\mathbf{c_i}$ corresponding to the image i.

At the end of the session the weight vector is used for updating the content weights for the relevant images of this session as long term learning from this query and is then memorized for future use as in

$$c_{ij} = \; c_{ij} \;+\; \rho\, w_j \tag{3.11}$$

Here w_j is the weight after the final user iteration, c_{ij} represents the relative importance of feature j in image i and reflects the relevant content learned by the system over all previous queries. ρ slows the learning based on the number of past sessions. When many users access and accept an image for a specific feature or features, we indirectly conclude that these features are important for that image.

However, in scenarios where iterative feedback from the user is missing, there is no incremental learning of the features relevant to the query. Such is the case even with popular web-based image search engines. Here, when the images are indexed by surrounding textual content, this method can be employed. This is, in a way, similar to the standard text processing scenario where given a database or a collection of documents the selective terms for all documents are to be estimated. Here the idea is to emphasize those features which better discriminate the relevant samples with respect others. On the lines of the method discussed above for the iterative feedback based approaches, we expect the consistency of the features over the relevant set and their variation over the irrelevant images makes them relevant for the images and vice-versa.

With little adaptation our earlier formulation for iterative feedback in Equation 3.10 fits the requirements as in

$$w_j \;=\; \log\left[\frac{\sigma'_{j\mathcal{R}}}{\sigma'_{j\mathcal{P}}}\right] \tag{3.12}$$

where $\sigma'_{j\mathcal{R}}$ and $\sigma'_{j\mathcal{P}}$ denote the dispersions of feature j over the $\mathcal{R}$ and $\mathcal{P}$ sets. The weights thus learned are then used for incremental long term learning as in Equation 3.11.

Our incrementally improving approach to long term learning allows flexibility in the learning, controlled in rate by factors like ρ. This ensures convergence of long term learning to the generally acceptable content in the image. Our incremental learning approach has a distinct advantage in terms of its computational expense, allowing efficient long term learning. The incremental nature makes it independent of available archived logs of feedback etc. It is also independent of the query so it performs irrespective of the availability of iterative relevance feedback. This allows it to perform independent of the retrieval approach employed by the system thus making it a highly portable approach for long term learning. Such unique characteristics of our approach make it an inexpensive and effective approach for long term learning of content.

3.6 Performance Measures

The system is initialized with some rudimentary concept (system weights or system parameters) when the system retrieves the images, the user gives feedback about the discrepancy between his and the systems concept. Then the system re-estimates its concept based on the feedback to represent as closely as possible the users concept. This concept update is done by changing the system weights. The point at which the systems concept and the users concept are fairly similar and consistently stays so is called the point of convergence. How fast this convergence takes place signifies how fast the system adapts to a user. In essence the difference in concept, which is the difference between weights, reduces until convergence. Hence the difference of weights at each iteration forms a good metric for measuring the speed and efficiency of the system.

The number of relevant images retrieved can also be used as a performance measure. This is an intersection of the set containing the M images and the set containing N images which are all the relevant images in the database. *Average Precision* is popular for characterizing the performance. The total number of relevant images is calculated by comparing the query image with all the images in the database and the user assigning score to each of them, and finally threshold the scores to get the total number of relevant images to the concept the user is looking. The number of relevant images is calculated by cross checking with the relevant images in the database.

Though average precision captures the overall gain with learning, yet in ranked sets many times the precision may not improve but the ranks of the relevant images in the recall set may improve. This cannot be captured in the general precision-recall curves. We introduce a novel approach for capturing this change. We call it *Rank Convergence (RC)*. Rank of the first image is 1 and that of the k^{th} is k. In this approach we monitor the change in ranks of the top N relevant images in every iteration. Ideally the first N images should be all from the query category so the sum of their ranks would be $N(N+1)/2$. This sum in every non-ideal retrieval would be more than that in the ideal case. With learning this sum is expected to continuously decrease tending towards the ideal case. It is able to capture learning at a much finer resolution as compared to average precision and recall.

We have used these measures in our experiments for validating the capabilities of our proposed approach. We present the results using graphs and tables and also support them with visual examples in the next section.

3.7 Implementation of Learning

We have designed our algorithms to work as an addition to **FISH** in Section 2.3. The approach can be summarized as in the Algorithm 2.

Require: Query as a feature vector q
Ensure: List $\mathbf{R}$ of images similar to the query as per user intent
  for all Iterations, t, of the user **do**
    if $t = 1$ **then**
      Initialize $\mathbf{w} = \{1.0\}$ for no bias
    end if
    Initialize $\mathbf{R} = \{\}$
    for all Dimensions, d , in order of decreasing **w do**
      Retrieve k-NN members from d
      Merge them into the master set $\mathbf{R}$ to have N closest members using Equation (3.7)
    end for
    Present top $\mathbf{R}$ to the user for feedback
    Use Equations (3.9) and (3.10) or Equation (3.12) to learn $\mathbf{w}$
    Re-initialize $\mathbf{R}$ and repeat the process as long as the user iterates.
  end for
  Use Equation (3.11) to update long term memory

Algorithm 2: Retrieval with Learning

We used a version of the **FISH** system designed for simulated experiments. It retains all the features of the original system except the web pages which are bypassed using scripts. We used such simulated runs of **FISH** with our for our experiments using synthetic feedback for both these sets.

The two datasets differ in the characteristics and complexities of visual content. They thus pose different challenges to the retrieval system in terms of both accuracy and performance. The first smaller set consists of concepts with limited visual overlap so allows better evaluation of performance qualitatively. The second set is larger and more complex and is better suited for metric evaluations and quantitative experiments. Both these sets are completely manually annotated and we use them for synthetic feedback on the retrieved sets of images.

The first is a diverse set of 1000 real natural images with about 100 from each of the categories including trains, surfers, hills, cars, sunset images, flowers etc. and thus they vary in their visual content. We represent these using MPEG-7 visual features as discussed in Section 2.6. We use Mahalanobis metric for estimating (dis)similarities over this dataset. The covariance matrix is pre-computed. The feature vector comprises of CSD and EHD descriptors from Section 2.6.

The second is a much more general set consisting of around 12,000 images collected from Flickr, Caltech datasets and COREL. This set has 58 categories and exhibits considerable visual overlap among the categories. These are represented using a combination of MPEG-7 descriptors and texture filter banks as discussed in Section 2.6. Given the complex information represented in the feature vector we chose to use a weighted asymmetric Kullback-Leibler Divergence (KLD) based metric instead of a Euclidean or a Mahalanobis as KLD is found to compare patterns in distributions.

3.8 Experiments and Discussion

We have conducted extensive experiments to validate the applicability of the proposed approach for learning. We have used the measures of performance as discussed in Section 3.6 for validating our approach for learning. As as baseline, we compare our performance with a system which has

no learning. We have used the **FISH** system (Section 2.3) for conducting all our experiments.

3.8.1 Feature Relevance Learning Experiments

We have used the improvement in precision percentage across sessions, with fixed recall, to demonstrate the effectiveness of our proposed long term learning method. We have conducted experiments to show the performance in both presence and absence of iterative relevance feedback. We have chosen the 1000 image dataset to show this result.

(a) Sunset

(b) Rocks

(c) Flower

Figure 3.3: Top 7 results for 3 queries for 3 different sessions.

For the iterative feedback based experiment we have randomly picked set of 20 queries while ensuring equal representation from all categories. We experimented for 20 sessions with 5 iteration for each. In each iteration, we gave feedback on the top 48 retrieved images. The system estimates the feature weights and performs retrieval using them. After 5 iterations, these weights update the long term learning for the relevant images. This should result in a characteristic gain in precision

in the first iteration of next session. We averaged the percentage precision over the queries and have included some randomly sampled instances in Table 3.3. As expected, our approach shows improvement in performance as sessions progress, in comparison to the approach without LTL.
In the next experiment, we show how our approach improves retrieval even in absence of incremental

Session	**1**	**2**	**5**	**10**	**20**
noLTL	58.7	58.7	58.7	58.7	58.7
Ours	58.7	63.7	69.5	71.6	73.8

Table 3.3: Percentage precision with iterative relevance feedback.

feature learning. We use the same random set of 20 queries and run the system for 20 sessions each with only one round of feedback, the first one. As a result relevant and irrelevant subsets of the retrieved set are formed. Using these, our approach computes the update to the content vector in long term learning. The average percentage precision for some randomly sampled sessions is compared to the LTL free approach in Table 3.4. The improved performance again validates our approach.

Session	**1**	**2**	**5**	**10**	**20**
noLTL	58.7	58.7	58.7	58.7	58.7
Ours	58.7	64.5	70.8	74.1	76.6

Table 3.4: Average percentage precision in absence of iterative relevance feedback.

We next conducted experiments using the second more complex dataset. We used the iterative learning approach to demonstrate the gains. We have used the unique performance measure, Rank Convergence (Section 3.6), for evaluating this experiment.

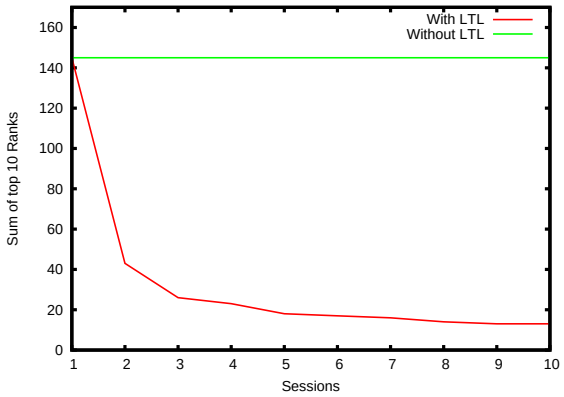


Figure 3.4: The plot shows the convergence of the sum of the ranks of the top 10 relevant images retrieved in the first iteration of each session. for the query. The convergence of the curve shows the improvement with inclusion of the long term learning into CBIR. The straight line appears for the case where there is no inter-query learning.

We have also included some visual results for a few queries from the database as shown in Figure 3.3. We show the top 9 results for 3 sessions for each query. The marked image on the left is also the query. The improving results in the subsequent rows show the gain with our approach long term learning approach.

All of these results validate the ability of our proposed long term learning method to improve retrieval accuracy.

3.9 Content Extraction

In most of the learning solutions in CBIR, validation of the idea is done using the performance improvement typically measured as precision. There is no explicit method for validating whether the semantic gap is really getting bridged. In case of user feedback based learning, right content is in agreement with the feedback of a majority of users.

Long term learning provides us an estimate of the importance of specific features to the content

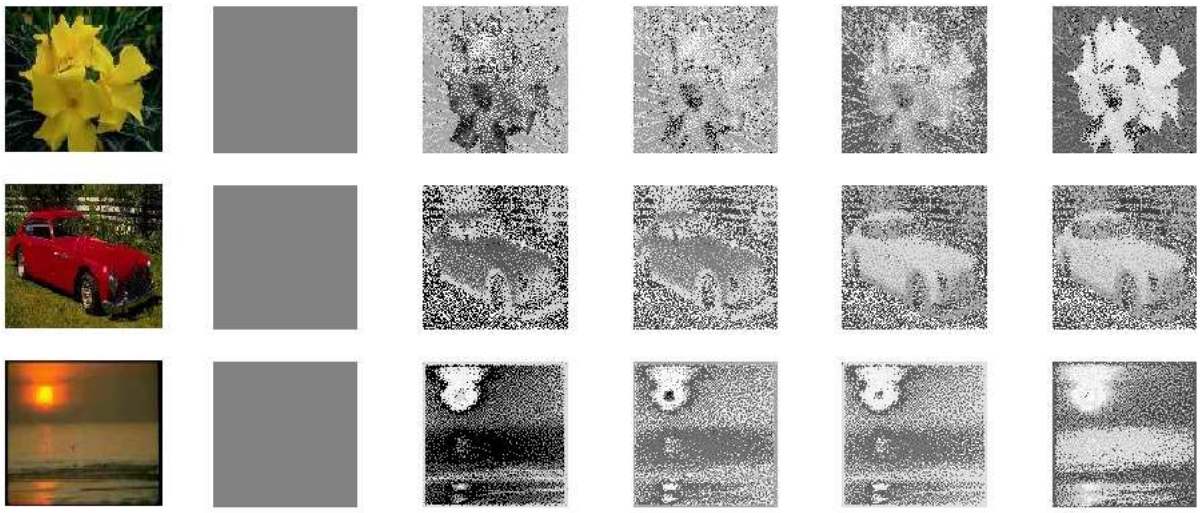


Figure 3.5: Content regions emerge over sessions(from l to r)

in the image using $\mathbf{c_i}$. Every pixel in the image has contributed to the feature vector description of the image. Now, in a class of situations, the relative importance of a feature c_{ij} could be mapped back to the image. i.e., the relevance of a pixel (or region) to the semantic content of the image can be computed. To simplify lets assume that the feature is primarily a color histogram. Then the estimated content $C_{mn} = c_{ij}\ iff\ I_{mn}$ is j, where I_{mn} is the color of the pixel (m, n). This allows distribution of the estimate of the content to the constituent pixels which have contributed to the specific feature. All pixels are similarly assigned a relevance based gray value. This image shows the most relevant as the brightest regions, performing a naive *region of interest* extraction. Figure 3.5 shows how the learning based content identification improves with sessions for a few sample images (left-most), from no information (in second image) towards *region of interest* extraction (in the right-most).

Next we present some images with their corresponding content images to validate our approach over some varied concepts in Figure 3.6. Our approach for content presentation also allows visual evaluation of the performance of long term learning.

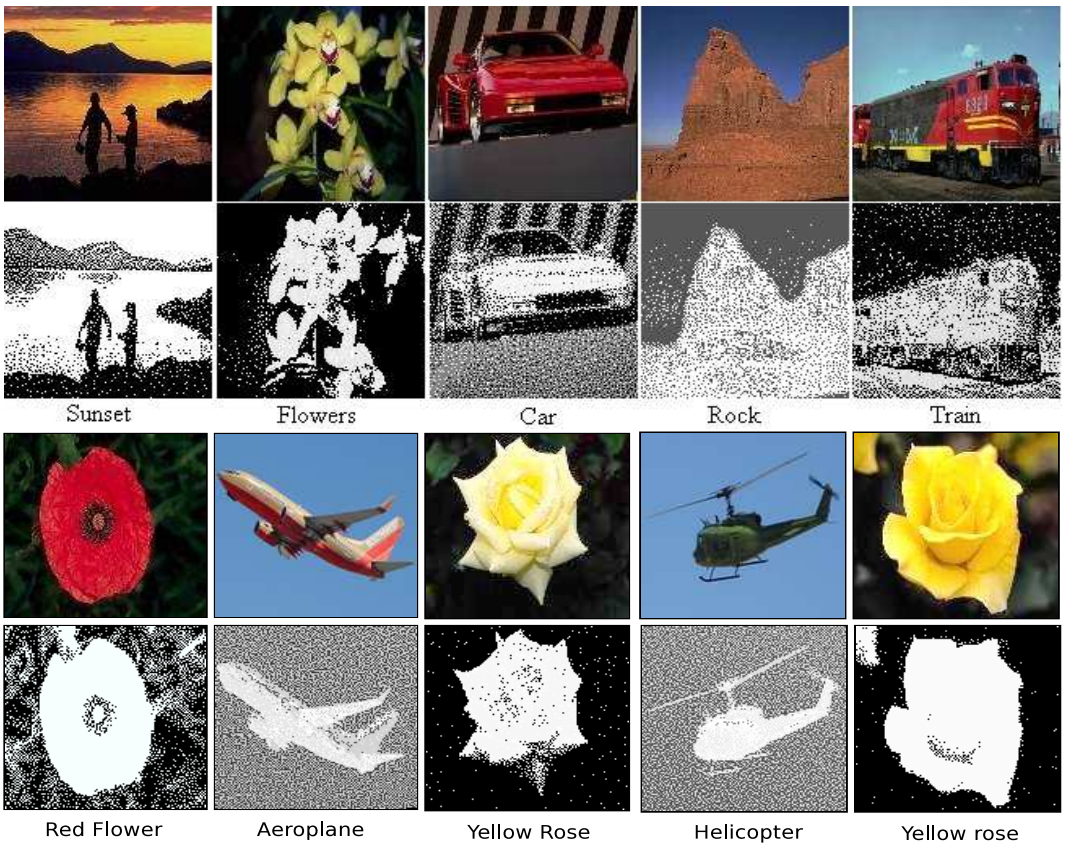


Figure 3.6: Content extracted from images using our long term content learning algorithm.

3.10 Summary

We have derived inexpensive incrementally learning solutions for long term learning motivated by pioneering ideas in text processing. We have proposed methods which work independent of the retrieval approach, making them highly portable. Our experiments show the effectiveness of our proposed approach on real datasets. We also propose a unique technique for extracting popular visual content from the images in the database using long term memory. In future we would like to extend our approach to also handle multiple concepts in images.

Chapter 4

Image Relevance Learning

4.1 Introduction

Content based image retrieval primarily relies on low level feature representations. Learning the relative importance of the features using relevance feedback has been extensively explored [49] for improving retrieval. But these still remain constrained by the capability of the low level features to represent the high level semantic information in the images. Relevance feedback is a user's judgment of the relevance of images to a concept. The co-relevance pattern in feedback is a semantic relationship between the images. This has motivated researchers to explore the logs of user feedback from systems as an orthogonal approach to semantic image retrieval [12].

Behavior patterns present in the system history *logs* have been explored as an orthogonal approach to semantic retrieval. Logs describe the relationships across images — which images are similar to which others. Here similarity is captured from the co-occurrence of the user acceptance. Performance of this feature independent approach, is heavily dependent on the quality and quantity of logs. With an ever increasing Internet user base, this seems to be an attractive proposition for image retrieval. The idea is to use the feedback pattern of the active user and predict his next set based on patterns in the logs which it closely resembles. This *content free* approach is based on collaborative filtering of logs as in [153]. Its success is deeply rooted in the log history. This method succumbs to sparse feedback. It is also unable to cope up with new or previously unseen query concepts.

To be applicable in real-life situations, the present day CBIR system needs to be scalable to Millions of images. Annotation of images (say with surrounding text) has been the popular method to build scalable image retrieval systems. However, noise-free annotations are hard to obtain. Feature-based direct retrieval is feasible with inverter file systems [154, 155] for large scale image retrieval. However, when the similarity metric is dynamic (i.e., when the user provides the feedback online), such rigid indexing schemes will fail to scale to large collections. This is an area which has not received much attention in the literature. This needs efficient methods for (approximate) nearest neighbor computations with a dynamic similarity metric.

In this chapter, we combine the advantages of both Content Based (CBIR) and Content Free (CFIR) Image Retrieval techniques in a Bayesian setting. The objective is to overcome the known drawbacks associated with these individual methods. We fuse the feature based *evidence* and log based *a priori* information in a Bayesian framework to design a posterior based retrieval. We have designed the system to support *interactive response* over large databases using an efficient multidimensional indexing scheme. This supports relevance feedback while similarity computation is done as a Bayesian inference. Retrieval time is in millisecond for a million images.

4.2 Image Retrieval

We shall first elucidate the two orthogonal paradigms of retrieval namely, Content Based(CBIR) and Content Free(CFIR) image retrieval before discussing our approach for optimally integrating them.

Content Based Image Retrieval The basic idea behind CBIR is to automatically extract visual characteristics from the images, and use them for indexing, comparisons and retrieval. These characteristics are generally some low level features like colors, textures or shapes. These features are often too primitive compared to the human perception, and as a result, there exists a semantic gap between the representation and the perception of content in images. This leads to poor performance accuracies. Extensive work has been done in literature on utilizing the invaluable user input for improving retrieval. Most researchers have tried to refine either the comparison metric based on the feedback or the features themselves. The general goal has been to estimate the relative importance vector of the features. These weights are then used for tuning the comparison metric to the query concept [49, 88]. Though the performance improves commendably, the overall performance is still restricted by descriptive power of the low level features. Researchers have also talked about methods to learn across sessions other than feature weighing. These generally include methods from information processing literature like SVD and LSI [77]. They work at the image level and use the patterns in the feedback for learning from them. These methods although good, generally do not scale up well with size in terms of computation. The accuracy also suffers due to the sparsity of feedback. This makes them infeasible for online deployment.

Most of the CBIR literature talks about the user feedback for improving retrieval for a specific user. However, only limited attempts are reported for extending the learning from one user to all subsequent users. A typical CBIR scheme retrieves results ranked on visual similarity. In general, only a very small fraction of the retrieved set is marked (accept/reject) by the user. This makes the marked set negligibly small as compared to the database size, especially in large scale setups. However pure feature based similarity computation is scalable to large databases, possibly at the cost of higher computational load.

Content Free Image Retrieval User behavior across different sessions and queries are relatively reliable. This assumption implies that different users will respond with similar feedback when presented with the same retrieved set for the same query. This allows prediction of the next iteration's relevant set based on the partial relevance vector of the present user from a history of similar queries or patterns [153]. This technique is known as collaborative filtering of logs. Especially in online web based systems, where collecting large feedback logs is possible, collaborative prediction based approaches are the apt choice. Such approaches are dependent on large clean logs for acceptable prediction.

CFIR-based approaches employ only the relative relevance of images, and are completely independent of the image representation [156]. This makes them better suited to perform semantically driven retrieval in comparison to content based methods. However, this is true only when the query is previously seen; else the performance could be very poor. This class of approaches effectively combines varied responses from many different users over the same or similar set of queries. Sparseness of the marked set of images is a major bottleneck in CFIR approaches. This happens because the users mark only a very small fraction of the samples as relevant or irrelevant. The sparsity effects the prediction accuracy and thus the results. Some of the general methods which address this sparsity issue are computationally infeasible for online use [157].

In summary both content based and content free approaches have their advantages and disadvantages. There exist some efforts in literature which have tried to design combined approaches which shares some of the advantages and shadows some shortcomings of each of the two [158, 159]. Most of these have tried to use one of them to boost the performance of the other.

There have been attempts on designing image retrieval systems throughout the last decade or so. The systems designed generally have the domain, scale, features, text, tags etc. as distinguishing parameters. Some notable ones are Airliners.net [31],ALIPR [34], MARS [82], Viper [89] etc. Most these operate on databases of tens of thousands using varied approaches from tags to general visual features. Our system supports relevance feedback for intra-session learning. We also do intersession learning with the help of retrieval logs and do the similarity computation in a Bayesian setting. In the next section, we demonstrate how the CBIR and CFIR can be integrated in a Bayesian framework.

4.3 Bayesian Image Retrieval

We employ a Bayesian framework for fusion of CBIR and CFIR. There have been works in literature which have tried to model the decision as a Bayes inference. Some groups used Bayes classifier for identification of relevant and irrelevant images [63], while others used Bayes in the content free framework, where they used Bayes with the conditional probabilities of images being relevant to a query to retrieve [156]. PicHunter [62] uses a probabilistic model for relevance feedback where they explicitly model the user's behavior in terms of his relevant image at every iteration till he stops. They then use the cumulative conditional relevance probability to scale down the feature based similarity of the target and the image.

Unlike the above methods, we use log based priors which gets built up over sessions of multiple users. Thus we use feedback patterns for inter-user and inter-session learning. We use the visual similarity of the database images with the query as evidence. This enables the feature relevance learning based on feedback across iterations for the active query. We use a Bayesian inference approach to estimate the posterior of relevance of the image to the query based on the visual evidence and the log based prior. The top N images are then selected based on the probabilities of relevance to the query.

We thus utilize the input from both content free and content based perspectives for enhancing the performance. Traditional direction for enhancing the performance has been by exploring a more descriptive feature set for a given database. Our approach formulates the image retrieval problem as one of estimating the probability of retrieving an image, as a posterior estimation problem. The approach uses the feedback logs and feature similarity for the priori and evidence input, respectively. We now discuss how two orthogonal approaches, namely content free and content based, are seamlessly integrated to give a hybrid Bayesian Inference based retrieval method.

4.3.1 Bayes Theory

Essentially, Bayes theorem provides a method to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself. In other words, Bayes theorem uses the history of use as the *a priori* knowledge, along with the evidence of the concept given the history exists, to provide an estimate of the concept's recurrence. We use the Bayes theory in the context of image retrieval by using both the logs of feedback from earlier users and visual comparison between the query and database samples to estimate an image's relevance to the concept and a specific query. We learn iteratively improving semantic relationships among images based on continuous feedback. We also use the

same feedback for learning the relevant image content in terms of feature weights for content based retrieval. We use the two in a Bayesian decision framework to estimate the posterior which serves as the probability of retrieval of a particular sample to the query concept. We can then retrieve the best few samples based on this posterior relevance probability distribution.

4.3.2 *a* priori from Feedback Logs

The history of user feedback logs is a collection of opinions which different users had on different queries. The belief is that the user opinion or feedback on the same query remains similar across different users and multiple sessions. The inconsistencies in the user feedback, results in the 'noise' in the probability distributions. As a result we can consider the feedback patterns in the logs as semantic relationships relating images together. We consider these iteratively improving relationships between images as a dynamic form of *a priori* knowledge of their semantic similarities. This relationship can be best modeled based on the pattern of co-occurrence of the two images in feedback logs. For the sake of simplicity, let us assume that the query image $\mathbf{q}$ is also present in the database. If the query is not present in the database (as in practical situations), the probabilities are calculated by weighted averaging over the concepts present in the database. For this purpose, concepts are extracted by clustering the logs (see next section for some details on how this is done in our system).

Let $R(\mathbf{q}, \mathbf{a})$ denote the event of retrieving the image $\mathbf{a}$ given $\mathbf{q}$ as the query. The *a priori* probability of this event $P(R)$ is the prior knowledge about the semantic similarity of these two images computed from the co-occurrence patterns in the logs. We compute it as

$$P(R) = \frac{n(\mathbf{a}, \mathbf{q})}{n(\mathbf{a})} \tag{4.1}$$

where $n(\mathbf{a})$ is the count or number of sessions/attempts where the image $\mathbf{a}$ is found to be relevant to a query by a user. And $n(\mathbf{a}, \mathbf{q})$ is the number of times when both $\mathbf{q}$ and $\mathbf{a}$ were acceptable together to a user. Note that $P(R) \leq 1$ It is 1 when $\mathbf{a}$ and $\mathbf{q}$ were also retrieved together. It is zero, when they never co-occur as acceptable images together.

This relationship models the relative relevant occurrence of images. It serves as a probabilistic estimate of the relevance of one image to other co-relevant images. The prior therefore provides us only a relationship and not a definitive content-level similarity measure among images. As the number of sessions increase, the reliability of this apriori knowledge increases. While implementing we start with $c(i, j) = 1$ for all images i and j to avoid any absolute zero priors.

4.3.3 Evidence from Visual Similarity

When the system is queried with an example image, a comparison at content level needs to take place. For this purpose, we represent each image $\mathbf{x}$ using a set of d numeric descriptors called features, $[x_1, \ldots, x_d]^T$. Then similarity of images is computed by comparing the corresponding feature descriptors. Relevance feedback is used to learn the relative importance of features. The basic idea is to learn the characteristics which relates the relevant images to the query. This is generally estimated in terms of weights for the features. We use both relevant and irrelevant images for learning the refined weights, $\mathbf{w}$, for the features. Note that this weight is specific to a query or concept of interest.

Let $S(\mathbf{q}, \mathbf{a})$ be the feature-level similarity of the images $\mathbf{q}$ and $\mathbf{a}$. Then the probability that the images are visually similar, given that $\mathbf{a}$ is retrieved for a query $\mathbf{q}$, is given by

$$p(S|R) = f(\mathbf{w}, \mathbf{q}, \mathbf{a}) \tag{4.2}$$

We make sure that the function is normalized to be used as probability. We compute the visual similarity between the query and each image in the database using a weighted similarity metric. We explain how these weights are computed in the next section (Equation (4.5)).

4.3.4 Retrieval as Bayesian Inference

Now, we explain how the retrieval is done given the similarity measure between images and the *a priori* known image-to-image relationships. This is done by Bayes rule:

$$p(R|S) = p(S|R)P(R) \tag{4.3}$$

We do not consider the denominator of the Bayes rule, since it does not modify the relative ranking of the database images, given the query. In practice, one could use alternate 'definitions' of the probabilities, as long as they satisfy the basic axioms of probability. The essence of our formulation is only a method to convert the *a priori* known image-to-image relationship into a posterior probability with the help of a feature-based similarity measure computed with the weight vector (corresponding to a session). At the end, we retrieve N images with highest posterior probability.

The patterns in the relevance feedback provide us an incrementally improving (or converging) *a priori* estimate of the semantic relationship among the images in the database. The visual similarity on the other hand is the evidence of the query-to-image feature level similarity. Together they provide a probabilistic estimate of the relevance of each database image to the user query.

When the query is not present in the database, the concept which is closer to the query image acts as a latent query. For this purpose the prior information is clustered and concept discovery is carried out. Note that this is an offline process, and does not directly affect the performance of the online system we present in the next section.

4.3.5 Comparison with Bayesian approaches

Bayes theory and inference has been explored in different ways by researchers in both multimedia and text search domains. People have basically used Bayes inference in trying to estimate probabilities of relevance for objects using available relations between documents or content free relations among images. Some have also combined the relations with content information such as in [158] where they use Bayes inference to estimate the probability of relevance of documents of text based on the patterns in words.

In image retrieval community Bayes theory was first explored by PicHunter researchers [62]. They used Bayes inference for probabilistically selecting target images for the user query based on his actions and those recorded in history. They also used feature similarity in the inference to support relationship based probabilities. They avoided any use of query refinement techniques at the image-to-image comparison level. The other notable work which uses Bayes theory effectively was done by Zhong et. al [63]. They use positive relevance feedback for learning the distribution of the positive samples and used negative feedback to correct ranking. The retrieval was essentially based on the posterior of the query belonging to a particular class denoted by a Gaussian.

Though commendable, none of these works identified and used the relationships and the visual similarity extensively and as independent complimentary sources of similarity. We have integrated the two sources of information into a Bayesian inference approach. We treat the relationships learned from logs as *a priori* and the visual similarity as the evidence. The two together provide a semantic similarity measure for the query-sample pair which is used for ranking. We use feedback for refining the query over iterations thereby improving retrieval further. We are thus able to bridge the semantic gap and at the same time avoid the classic cold start problem.

4.4 The Retrieval System

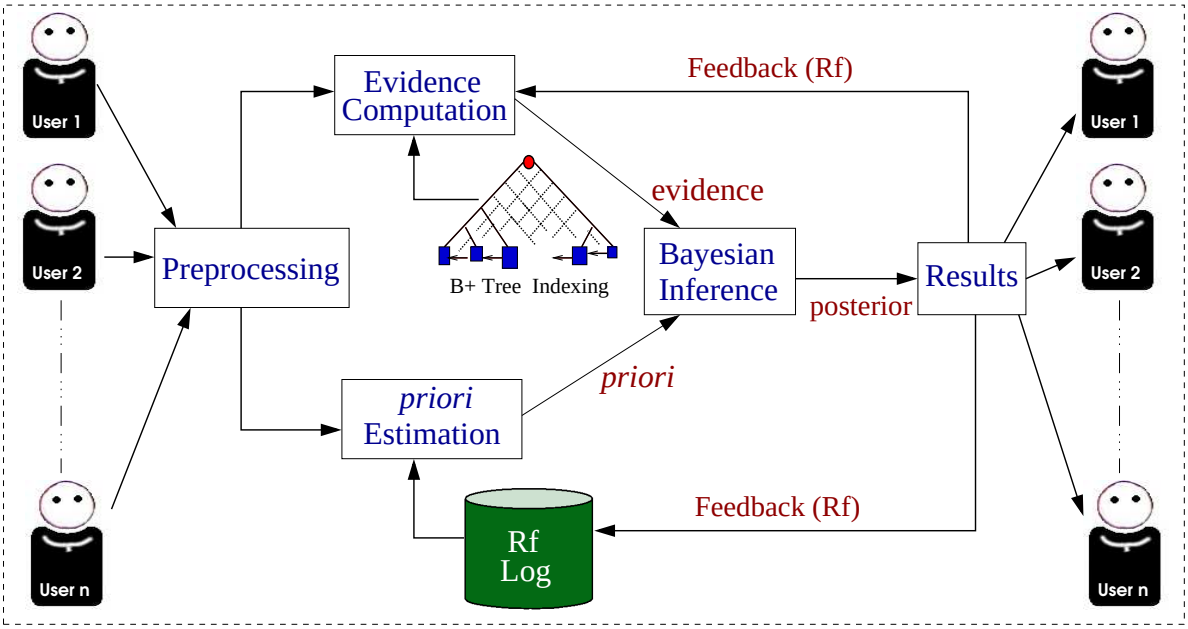


Figure 4.1: Architecture of our scalable Bayesian Image Retrieval system

Our system, as shown in Figure 4.1, supports Query-by-Example. Multiple users can access the system at a time. The query is processed online to extract a set of features. The query is then compared for similarity with a subset of the dataset, pruned on visual dissimilarity. The comparison with the query takes into account the visual evidence as well as the log based *a priori* knowledge on the semantic relationships it has with other images. The N most probable images are displayed to the user for his opinion. The user feedback is then used for refining the estimate of the visual concept in the query which is used for refining the retrieval in the next iteration. It is also used for updating the semantic relationship among the relevant images. All these happen in interactive speed with sub-second retrieval time for close to a Million images.

4.4.1 Representation

Each image in the system is represented as a vector of numeric feature values $[X_1, \ldots, X_d]^T$, constituting a multi-dimensional space where each image is a point. We chose a subset of a generally accepted standard set of descriptors, independent of the image collections we use. A detailed discussion on the features can be found in Section 2.6. The dataset of images is indexed offline on these features. The query image processed online to extract these features.

4.4.2 Indexing

Exhaustive k nearest computation is an infeasible approach for interactive systems. We propose to perform an approximate k-NN retrieval. We use a B+ tree based index as most other existing indexing schemes require a fixed similarity metric on which the index is built [4, 5, 6]. Such schemes do not suit our environment since the similarity metric depends on the weights of features, which continually change with fresh queries and user-feedback. A few indexing schemes are available for changing similarity metrics [79, 80]. However, these approaches enumerate a large number of

candidate images from the dataset before determining the most similar ones. The reason being that they treat all features (dimensions) uniformly, while real-life datasets have inherent clusters in subspaces.

Our index structure utilizes a B+ tree for each feature/dimension to store the corresponding feature values. We assume a query feature vector $\mathbf{x_q}$ is given and we are required to compute its k nearest neighbors of it with a given feature-weight vector $\mathbf{w}$. For retrieval, each B+ tree is used to find the position where $\mathbf{x_q}$ would be inserted in it. The data points in the neighborhood of $\mathbf{x_q}$ are then retrieved from all the B+ trees. They are then merged into a single set and the nearest p data points ($p \geq k$) from q are chosen based on the weighted visual dissimilarity metric. The B+ trees are enumerated in the order of decreasing relevance of the corresponding features or feature weights making it likely that the closest data points get enumerated early. The search space is reduced drastically with this simple approximation of p-NN samples without much loss of generality. Demonstration of the empirical error bounds of this technique is beyond the scope of this work. A more detailed discussion and evaluation of our indexing scheme can be found in Section 2.5.

4.4.3 Feature Learning from Relevance Feedback

Once the feedback is given, the system absorbs it by learning. The basic idea is to use the relevant images and learn the content which relates them to the query. This relation is generally learned in terms of weights for the features. We use relevant and irrelevant images to learn iteratively refined weights.

We use a *Discriminative Variance* based approach to estimate iteratively refined feature weights. Assume, $\mu_{j\mathcal{P}}$ and $\mu_{j\mathcal{N}}$ represent the means of the positive and the negative samples for the j^{th} feature and $\sigma_{j\mathcal{P}}$ and $\sigma_{j\mathcal{N}}$ their variances. And $\mathcal{P}$ and $\mathcal{N}$ denote the relevant and the irrelevant subsets of $\mathcal{R}$, the retrieved set of images.

$$s_j \;=\; \frac{\sigma_{j\mathcal{N}}}{\sigma_{j\mathcal{N}}} \tag{4.4}$$

We use equation (4.4) to estimate the score s_j for the j^{th} feature after every feedback iteration. These scores are then used for incrementally updating the weights of the features as

$$w_j^t \;=\; \gamma\, w_j^{t-1} \;+\; \beta\, s_j \tag{4.5}$$

where w_js represent the weights for the j^{th} feature after the $(t-1)^{th}$ and the t^{th} iterations. The parameters γ and β control the learning rate. Further discussion on the feature relevance learning can be read in Section 3.5.

4.4.4 Retrieval

Based on the query, $\mathbf{q}$ and the weight vector, $\mathbf{w}$ the system retrieves the k visually most similar images, x_i, from each B+ trees in non-increasing order of weights.

The weight vector, $\mathbf{w}$, is used to compute the visual similarity between q and x_i using a weighted Mahalanobis metric. The covariance matrix, $\mathbf{M}$ is estimated offline as a pre-processing step. This provides us as evidence of visual similarity between the query concept and the concepts in the database images. The visual similarity s_i between the i^{th} image with the query $\mathbf{q}$ is modeled as

$$S_i = \Big[\big(\mathbf{W}^T \left[\mathbf{x}_i \;-\; \mathbf{q} \right] \big)^T \mathbf{M} \, \big(\mathbf{W}^T \left[\mathbf{x}_i \;-\; \mathbf{q} \right] \big) \Big]^{\frac{1}{2}} \tag{4.6}$$

where $\mathbf{W}$ is the *diagonal* matrix made of $\mathbf{w}$.

The *a priori* probability, for an image x_i to be semantically related to image q is estimated using (4.1). Information from the matrix of these probabilistic relationships is summarized into a set of representative relations or concepts, $\mathbf{V}$, present in the logs, as discussed in Section 4.5. The retrieval in the first iteration is essentially evidence based with all images being equi-probable to be relevant to the query. Next iteration onwards, we use the active feedback vector to choose the most similar concept to that of the query and retrieve using it. The concepts or means in $\mathbf{V}$ are also expressed as relationships between images. Therefore, the probabilities of relevance of any image to the active query can now be used for performing improved retrieval from amongst the most probable images. We compute the probability of each image retrieved from the feature dimensions for their relevance to the query using the visual evidence and the prior probability. The most probable N images in a Bayesian inference framework are then returned to the user as retrieved results.

4.4.5 Updating the *a priori*

After every session the system absorbs the feedback pattern by updating the relationship estimates. In the process, every query on the system increments $n(\mathbf{i})$ for some $i \in \mathcal{P}$, where $\mathcal{P}$ represents the set of images from the database D, marked relevant by the user. A relevant occurrence of an image effects it's semantic relationship with all the other images it has ever been co-relevant with. This occurs based on the modification of their $n(\mathbf{i})$s in the above equation. We can re-compute the probabilistic relationships for all pairs of effected images in the database and update the $n(D) \times n(D)$ relationship matrix after every query or after every few queries. The update will also require knowledge of both the updated $n(\mathbf{i})$ and the updated $n(\mathbf{i}, \mathbf{i}')$, for every image i' co-relevant with image i. as a result they too will have to be archived in matrices making this approach expensive.

As an efficient solution we compute the *a priori* probabilities between pair of images, i and i', on the fly when needed during the retrieval process. As a result, we store only the $n(\mathbf{i}, \mathbf{i}')$ and the $n(\mathbf{i}, \mathbf{i})$ which make up a single data matrix sized, $n(D) \times n(D)$, unlike earlier. With every query session some of the values get incremented based on the i^{th} or the i'^{th} image getting marked as relevant by the user. Every relevant occurrence of an image i also increments its total relevance count $n(\mathbf{i})$. We use these incrementally updated values for estimating the semantic relationship between the pair of images under consideration at any point in time. We use MySQL to index the $n(\mathbf{i}, \mathbf{i}')$ and also the $n(\mathbf{i})$ values allowing efficient retrieval and update.

4.5 Concept Discovery

At the end of every feedback session some of the relationships in the relationship matrix are updated depending on the images which were judged similarly relevant by the user. The matrix can thus be considered as a loose collection of all possible relationships among the images in the database. These relationships can be summarized into some k concepts, $\mathbf{V}_i, ..., \mathbf{V}_k$. Each of these concepts represents, as a vector of relationships, a summary of original relationships which had similar sets of co-relevant images. When a new query is presented to the system there is essentially no way of assessing the concept it is associated with. So the initial retrieval happens based on visual evidence alone. Once feedback is received from the user for the next iteration the most similar concept known in the collection can be chosen. The co-relevance relationships among the images in the chosen concept can then be used as *priors* of their relevance to the active query in the Bayesian inference formulation.

We model the summary of relationships in the matrix as a matrix partitioning effort. The idea is to partition the matrix in a manner which brings all the similar relationships together to be represented as a concept, $\mathbf{V}_k$. We propose to use a hierarchical k means clustering to achieve the same. The hierarchical approach allows self discovery of the possible concepts present among this matrix of relationships. This truly reflects the dynamic adaptive nature of the concept descriptions which are free to get re-molded based on feedback patterns. So once the new feedback vector modifies the relationship matrix we can re-cluster the relations to re-estimate the concepts present in the log based matrix. These newly discovered concepts can then be employed for retrieval for subsequent queries.

This re-clustering requires a hierarchical similarity clustering of a dynamically changing set of a $n(D) \times n(D)$ matrix. So although it is a repetitive computation, but it does not effect the retrieval time as we have modeled it as an *offline* process. This *offline* update ensures commendable retrieval performance in terms of accuracy as well as time, thereby allowing for efficient unhindered scalability.

4.6 Experiments and Discussions

We have conducted extensive experiments to validate our claims on our Bayesian semantic image retrieval approach. We have used our **FISH** system (Section 2.4) for all our experiments. Through our experiments we compare the performance with a CBIR system only. We do not compare explicitly with a CFIR approach as in case of previously seen queries the accuracy will be 100% and *zero* for a previously unseen query. Accuracy estimates between these extremes are not a consequence of the choice of CFIR algorithm so they cannot be compared with.

4.6.1 Datasets

We have used two datasets for our experiments. The first, $\mathbf{D_1}$, is a set of around 12,000 real images collected from Flickr, COREL and cartoon videos. It consists of 58 categories with approximately 200 images from each. The set includes images of cars, ships, trucks, tiger, roses, trains, cycles, motorcycles, birds, scenes, sunsets etc. We have used a subset of features discussed earlier in Section 2.6. This set of images is fully manually annotated and is used for our accuracy experiments. We have also preprocessed queries on this set to generate a spare synthetic feedback log.

The second set, $\mathbf{D_2}$, comprises of the Caltech-256 dataset. We have used this set to show the improvement in precision with a complex set using the popular *bag of words* approach. We wanted to avoid any unnecessary dependency on interest point detection using one of the standard point detectors like MSER interest point detector. So we have used dense grid points as interest points on these images. The points were then described using SIFT [76] local region descriptors. The pool of points were clustered into a vocabulary of $2,000$ words. This was then used for generating the bags of words for the images for retrieval. This set too is completely manually annotated.

The two datasets represent two different approaches for feature based image retrieval and thus help us validate the universality of our proposed Bayesian Semantic Image Retrieval.

Apart from these two annotated sets we have also used the large unannotated set, $\mathbf{D_3}$ of approximately *1 Million*, 10 dimensional data points which we describe in Section 2.8.1. We used this in experiments to show efficient scalability. These experiments were conducted in a manner similar to those discussed in Section 2.8.2.

Approach	Category				
	1	2	3	4	5
BAYES	59.91%	75.69%	63.94%	72.94%	68.53%
CBIR	31.38%	39.38%	41.38%	32.54%	45.29%

Table 4.1: The tables reflects the performance of the proposed Bayes Inference method as compared to the CBIR using a human log on $\mathbf{D_1}$. We have included results for 5 of the categories. The gain varies based on visual complexity and availability of good *prior* information for the pairs of images which were used in the experiment.

4.6.2 Log Generation

For our experiments comparing the our approach with basic CBIR in terms of accuracy we used two different types of history logs. The synthetic logs are always doubtful for inherent patterns and otherwise non-reflective of the true human behavior with noise in feedback etc. In order to prove the effectiveness of our approach in real world environment we generated a log using relevance feedback from real, human subjects on dataset $\mathbf{D_1}$. This log was then used to show the comparison in accuracy achievable as discussed in Section 4.6.3. To generate the log we used a random selection technique which helped us rule out any bias in the log generation what-so-ever. We randomly selected a query and a retrieval set. Now the human user was asked to mark the images in the randomly retrieved set which he found were relevant to the random query. We presented sets of 50 images each in response to $20,000$ such queries. The feedback was recorded and then processed into the co-occurrence matrix.

In order to show the versatility of our approach we also ran a similar accuracy comparison experiment using the second dataset, $\mathbf{D_2}$. For this experiment we used a synthetic feedback log where relevant samples from the retrieved 50 were marked automatically based on class labels available. Here also we used around $20,000$ query sessions. The small number of samples evaluated for relevance (50) and limited number of queries conducted help mimic the lack of feedback encountered by operational retrieval system.

4.6.3 The Human Experiment

In this experiment we show how our approach using the integration of visual similarity and log based relationships in our Bayesian framework achieves better results compared to pure CBIR in a real world environment where the feedback is noisy and sparse. We used dataset $\mathbf{D_1}$ and logs from Section 4.6.2.

We used a set of 5 random queries from each of the 58 categories. The average precision was recorded for each category using our Bayesian approach with the logs above. The precision was also computed using our learning CBIR system in Section 2.3. We compare the gain in precision for a few of the categories in Table 4.1.

4.6.4 Precision gain with Bayesian

We conducted another experiment to prove the gain in accuracy using synthetic logs, with our Bayesian approach. We compared it to our learning CBIR 2.3. The average precision at the end of the experiment was compared category wise for the CBIR and our Bayesian approach. We compiled the results in Table 4.2.

As can be seen in Table 4.1 and Table 4.2 our Bayesian approach out performs the CBIR

Approach	Category				
	1	2	3	4	5
BAYES	72.08%	67.22%	78.75%	74.43%	59.35%
CBIR	42.00%	47.08%	46.52%	32.84%	28.84%

Table 4.2: The table compares the precision achieved using a pure CBIR and our Bayesian approach using synthetically generated feedback logs over dataset $\mathbf{D_2}$. We compare results for 5 of the categories.

commendably. This proves the effectiveness of our novel approach for integrating *a priori* logs and visual evidence for retrieving similar images.

4.6.5 Learning in BSIR

In Section 4.6.3 and 4.6.4 we showed the immediate advantage with using *priors* in our Bayesian framework. There we proved the case assuming a system in a state where it has considerable history logs. A Bayesian retrieval system like ours should be able to show improvement in performance at the start itself and mature with use. The result should reflect in the improving accuracy over time. We conducted an experiment to validate the same for our approach. We used the $\mathbf{D_1}$ dataset for this purpose. We initialized the system with *unit* co-occurrence values to avoid *zero* probability issues in computations. We ran the system for a randomly picked set of 20 queries for 10 sessions each continuously. The accuracy statistics were recorded and precision averaged across the 20 queries was plotted against the sessions. We also ran a similar experiment with the same queries on our CBIR system. The results are compared in the graph in Figure 4.2. As can be seen, initially the accuracy is similar to that of CBIR with learning but with time it becomes much better and converges to a commendable precision. The reason is that initially when logs are poor they do not contribute much to the retrieval accuracy so it is essentially CBIR based. But with time logs improve and retrieval too, as a consequence. In case of CBIR since there is no relation between sessions or queries, retrieval accuracy repeats across sessions. As is evident our approach converges

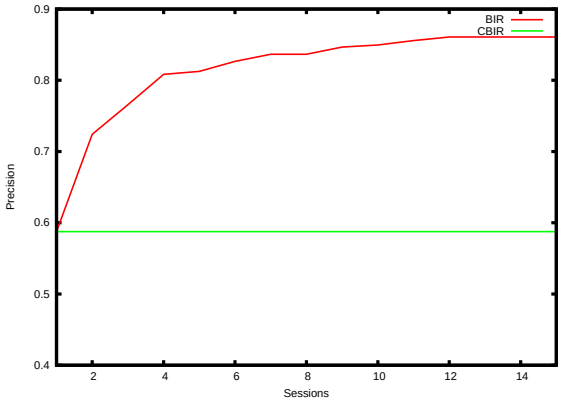


Figure 4.2: The graph on left shows how the average precision improves with sessions using our Bayesian approach while it remains stagnant for the CBIR. This provides improved accuracy to the user in far fewer sessions.

to a much closer to ideal situation quickly. The better accuracy achieved by our Bayesian approach is a result of effective integration of the *prior* knowledge and visual similarity.

4.6.6 Qualitative Comparison

We next present results of some of the queries we used with our approach. We have included the top 9 results for each of our queries as retrieved by our Bayesian approach in Figure 4.3. We also compare our results with those for the same queries from the CBIR approach. The first image on the left in each row is the query itself. For each pair of rows the first is from our approach while the second is the set from CBIR. Both approaches perform well for visually very similar images. But with complex queries where visual similarity is much weaker than conceptual similarity our Bayesian performs better than the CBIR with help from the *priors*. Next we present two example

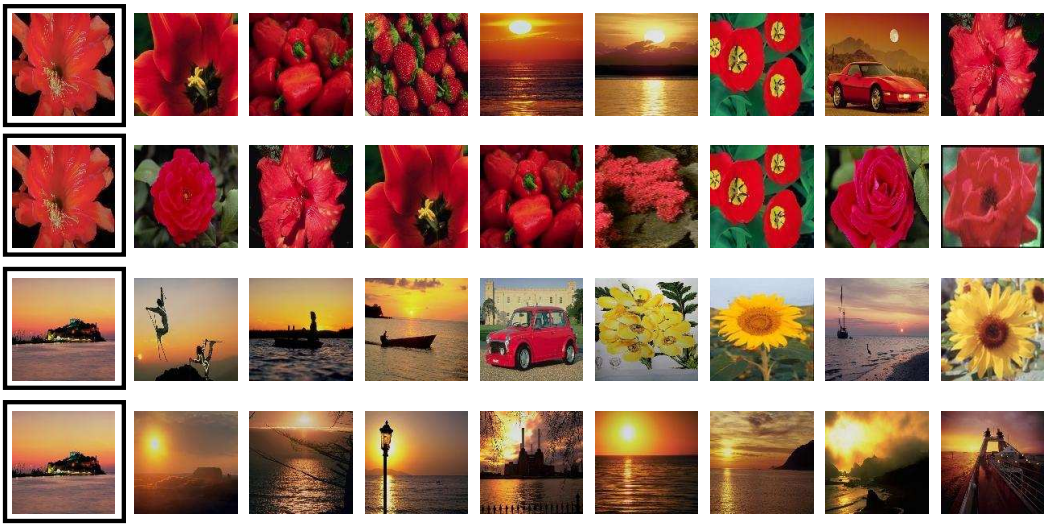


Figure 4.3: Top 9 ranked results returned by our Bayesian and the CBIR for a couple of queries (the left most image is also the query).

results where we show the new relevant images which got added to the retrieved set with Bayesian in comparison to pure CBIR in Figure 4.4 As can be seen, the results from Bayesian also includes visually dissimilar but conceptually similar ones unlike those with CBIR. The results, especially the visually complex ones, show how our Bayesian reduces the semantic gap by effectively integrating the *a priori* knowledge with the visual similarity.

4.6.7 Efficient Scalability

Efficient scalability is the key to the practical usefulness of any algorithm, especially for retrieval on the web. In this section we present results showing how our Bayesian approach retrieves in interactive times (sub-second) by effectively utilizing our index structure 4.4.2. We show how our approach scales seamlessly to large number of data points and also to large number of feature dimensions, while still maintaining interactive retrieval response times.

We use $\mathbf{D_3}$ for these experiments which are conducted in a manner similar to those in Section 2.8.2. We use average retrieval time over 5 randomly selected queries. We first observed the change in retrieval time with increase in database size. We plotted the average retrieval time at increasingly larger number of samples in Figure 4.5 The time was measured at intervals of $25,000$ samples increasing the number from $5,000$ up to *1 million*. As can be seen the time remains much less than a second throughout. The pattern in increase is discussed in detail in Section 2.5. Next we conducted an experiment to observe the effect of increasingly large number of dimensions on the retrieval time. We used a subset of $0.1\,million$ points from $\mathbf{D_3}$. We recorded the average response

(a)

(b)

Figure 4.4: Relevant images among the top 30 retrieved images. We show CBIR results in row one for each example followed by which other images got added to these when Bayesian was used. Notice the conceptual similarity among the new additions

time at 10, 20, 50 and 100 dimensions. The results plotted in Figure 4.5 show that the response times remains interactive even at large number of dimensions. A detailed analysis of the linear increase can be found in Section 2.5.

The pattern in response time remains governed by our index with minimal effect from our Bayesian scheme. This is so because we have designed the Bayesian update as an offline process. This ideally runs as a deferred update for the co-occurrence matrix. The retrieval of the occurrence pattern for the images for this query is also managed efficiently by using MySQL. We need just two MySQL calls for retrieving all the occurrence values required for the active query. This efficient method of storage avoids a lot of otherwise necessary operations and saves a lot on retrieval time.

4.7 Summary

In this chapter we proposed a novel Bayesian inference framework for integrating two complimentary sources of relevance information relating pairs of images. We integrated inference from the visual similarity based CBIR with the relations based CFIR in a manner so as to optimize the advantages and suppress some of the critical weaknesses of each of them, like the cold start and semantic gap. With helps of the experiments discussed above we have clearly shown the utility, and advantage of using our Bayesian framework for achieving more accurate retrieval efficiently in interactive

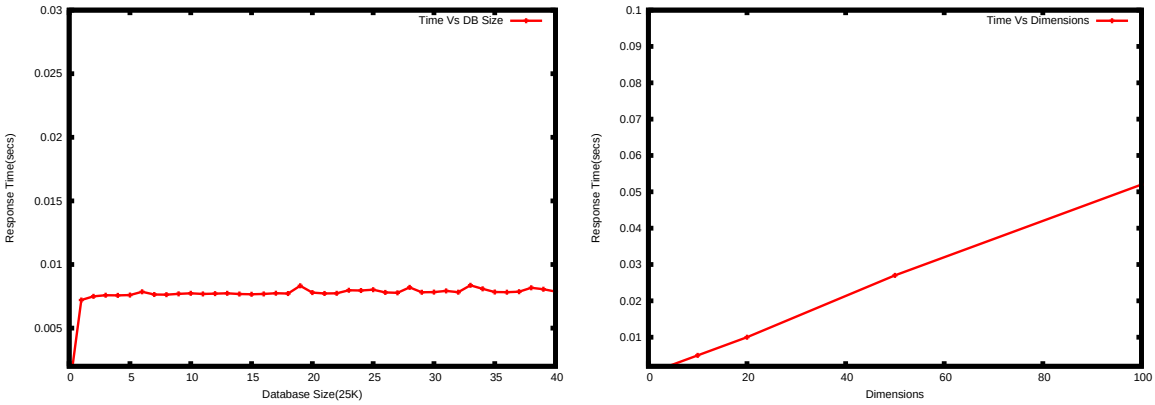


Figure 4.5: The graph shows the average retrieval times (in secs.) achieved by our approach at different DB Sizes over a 10 dimensional dataset. The sub-second response keeps the retrieval interactive. The graph shows the average retrieval times (in secs.) achieved by our approach at different number of dimensions using 0.1 *million* data points. The sub-second response ensures interactive retrieval.

time. Interactive response encourages user involvement and improves learning and accuracy. In future we would like to extend this approach to discover complex concepts embedded in multimedia databases.

Chapter 5

Diversity in Image Search with Skylines

5.1 Introduction

Interactive response within a few seconds is among the foremost criteria for judging the usefulness of a retrieval system. This is especially true for systems which serve online. In addition, the similarity computation carried out by the retrieval system will have to be semantically meaningful. In many scenarios the retrieval system is unaware of the user intent. More often than not the user too is only vaguely aware of his true intent. Even then, the system is expected to retrieve results which match the user's intent and that too in interactive time. The user effort in refining or presenting his query too should be minimized. The key is to present a diverse set or a variety of similar results to the user to allow him to quickly choose his intended concept. The gap between the representation and human perception leaves pure similarity based retrieval inadequate. These approaches are also inherently incapable of capturing the *variety* in concepts while retrieving similar results. This is especially true with audio-visual systems [12]. The requirements suggest the use of a method which can retrieve samples which are similar yet diverse by considering attributes independently. Another perspective could be a method which is capable of eliminating *conceptual redundancy* from the result set to be presented to the user. Skyline queries from database research merit as a natural solution to this problem.

5.1.1 Skyline Queries

Skyline queries are used in databases to handle data defined by metrically incomparable attributes, like cost of accommodation in a hotel and its distance from beach. So you can choose the best accommodation, either cost wise only or proximity wise only. If there are two hotels, first is closer to the beach and costlier and the other is farther but cheaper you cannot compare them without any information which external to the system like a preference, for example. Detailed discussions can be found in [160, 161].

Skylines operate by retaining only those samples which are not dominated by any others. In this context dominance is defined as being closer to the query in all the attributes, individually. Skyline computation has been noted to be an expensive procedure especially for interactive retrieval systems. This is primarily due to the exhaustive nature of the process that covers all attributes and samples in the database. Offline methods of pre-processing the skylines do not adapt to changing databases and is not applicable in learning based systems where the relationships between attributes

can change with the user.

Over the last few years skylines have attracted significant attention but it has remained confined to the database research community [161]. Their computational expense has motivated people to find better methods. Kossmann [162] first talked about skylines as a nearest neighbor problem and used a divide-and-conquer strategy on the R - tree indexed database. This suffers from major space requirements. In [160] we can find a comparative coverage of most of the core and peripheral ideas in skyline computation. A number of algorithms for computing skylines have been mentioned in [160] like the basic block nested loops where it is a nested loop structure, or divide-and-conquer approaches which try to partition the database to make main memory processing possible. It also discusses approaches like bit-mapping, indexing lists, also the general nearest neighbor approaches with R - trees and their variations like ranked, dynamic, dominating and constrained skyline queries and presents experimental evaluation of these. Later research has focused mainly on the computational feasibility and application benefits of skylines. In [163] a progressive skyline algorithm has been studied which produces the skyline incrementally. In [164] the idea for day-to-day applications by introducing spatial information into the skyline queries was adapted. Skylines have also been explored to retrieve the top-K Skylines points as in [165].

Dominance computation, being based on attributes can be addressed well when there is support for efficient retrieval from individual attributes. Efficient indexing is also the key for achieving interactive retrieval. The indexing scheme should support efficient similarity based retrieval. Therefore, the ideal scheme would be where the similar samples occur next to each other in the index. This can be achieved by using dimensional indexing schemes such as multi-dimensional tree based indexes which use similarity of data within an attribute or dimension like R trees [5] and SS trees [6].

To achieve this goal we propose our novel approach which uses an approximate k-NN approach for computing the skyline. It uses a B+ tree based dimensional indexing scheme proposed in [7]. It then employs a skyline extraction approach to retrieve the top k diversely similar objects for the user query. Our proposed approach gains on response time with a small cost on accuracy. Approximate k-NN retrieval reduces the search space drastically to a much smaller similarity space. Any efficient dimension wise skyline approach can now be used on this set to extract the skyline set. We conduct extensive experiments to validate the variety and accuracy of our proposed combination. We also present a case based analysis of how the diversity achieved with our novel approach benefits other online retrieval problems in the context of image retrieval.

5.2 Diversity in Similarity Retrieval

Ambiguity in user's query and its inexact interpretation by the system often results in poor user satisfaction. This happens as most retrieval systems are purely similarity based and try to compare data objects using some kind of scalar metric. This, though true for any general retrieval application, manifests itself boldly in image and video retrieval systems. Here it is a direct consequence of the wide gap between the subjectivity of human interpretation of visual information and the state of art methods for representing it automatically in machine efficient form. Especially in retrieval scenarios where the user tends to be vague in query specification and the systems find it hard to interpret the query in general, presenting a mixed set of results to the user is better. Consider a general situation in Figure 5.1. The user has presented an image of a house like building and a car in front of it. He hasn't in any manner specified whether he is interested in the car or the building. The system can end up presenting him results only with images of cars in front of buildings as in *Set 1* in Figure 5.1. The system can present a similar set of results as in *Set 2* in Figure 5.1 which comprises of images of *buildings*. It can also present a set of *car* images as in *Set 3* in Figure 5.1.

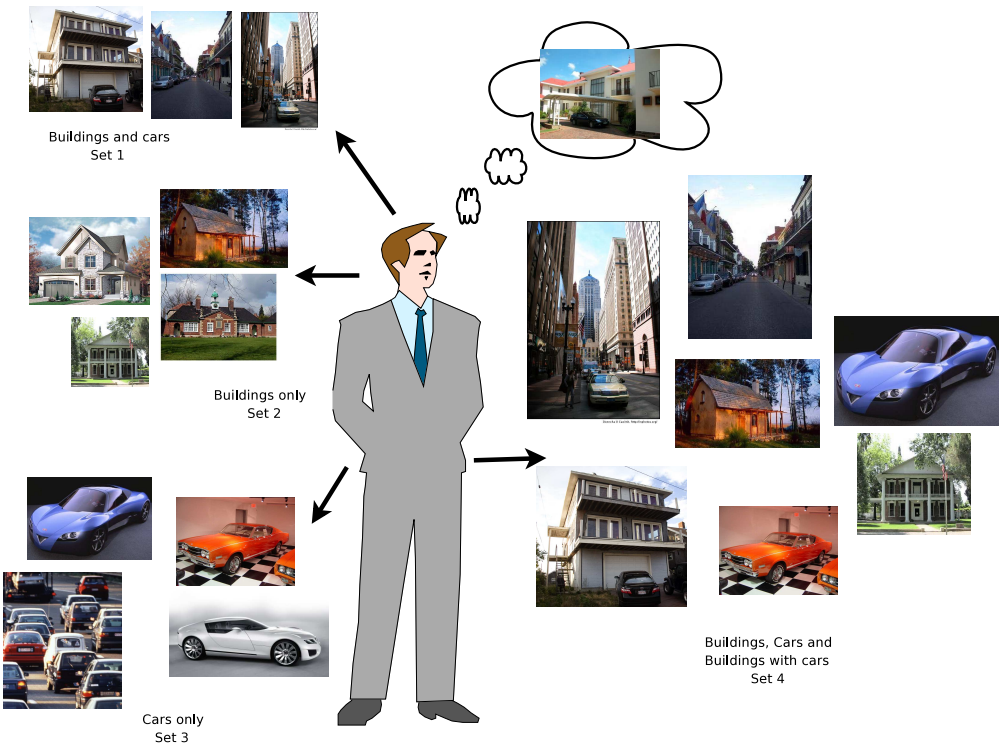


Figure 5.1: The user queried for an image with a car in front of a building. The ambiguity in the query would make a similarity approach like a traditional CBIR, to present results like one of those on his left (*Set 1* of cars in front of buildings, *Set 2* of buildings or *Set 3* of cars). A diversity enriched similarity retrieval approach like ours would present results like those on his right (*Set 4*). The variety in the results, helps the user quickly remove his ambiguity.

It can alternatively choose to present results like those in *Set 4*, where the user is shown images of buildings, cars and those having both buildings and cars. The user then has the chance of selecting the car, building or both depending on his intended query. His input can be interpreted by the system and he can be returned a set of results specific to his choice, quickly, in a couple of iterations only.

The mixed set can contain results which are similar to the query in different characteristics. But pure similarity based measures compare objects using scalar values estimated generally from dissimilarity metrics like the L-Norms. Such metrics occlude the comparative effect of individual attributes on the similarity. As a result these approaches are inherently incapable of retrieving diversely similar results. Skylines as we know eliminate redundancy from a set of data objects. This when applied to a set of samples, retains only the ones closest with respect to different attributes individually. This results in a diverse neighborhood. In terms of visual media this associates with results which are similar to the query in different characteristics or conceptual interpretations. Diversity in the retrieved results could be a unique advantage especially in multimedia retrieval systems where representations are still not very rich. Here retrieval has traditionally been a similarity problem and even the state of art in multimedia retrieval have focused on similarity metrics and approaches [12, 11]. The idea of engineering the retrieval set to present more information has been researched in multimedia retrieval as active learning [166]. But the major concern has been improving interaction for better retrieval accuracy. Little emphasis has been given to diversity based improved presentation of results to the user.

Figure 5.2: Figure shows the top 5 images retrieved in response to a 'cake' query image (highlighted image on the left is the query). First row shows results from a similarity retrieval CBIR and the second from our skyline based diverse similarity retrieval technique. The variety in similarity is evident among the images in the second row.

Motivated by the dominance based redundancy elimination property of skylines we propose an approximate k-NN based skyline algorithm. The problem can be posed as that of identifying a set of the N most similar objects to the query q and then eliminating conceptually redundant samples to present a diverse set S to the user. We approach it by first retrieving k nearest candidates from every concept or attribute into C and then eliminating redundancy from C to arrive at a diversely similar set, S, which can be presented to the user. We are able to achieve commendable improvements in response time with negligible loss, if any, in the accuracy of the skyline by using our dimensional B+-tree based indexing scheme. Such a diverse set of results is very useful for most of the similarity retrieval problems in general. Consider the example in Figure 5.2, here we can compare the results retrieved by a pure similarity image retrieval scheme like a traditional CBIR and those retrieved by a skyline assisted CBIR. As is evident from the variety in similar results from the skyline approach, it merits as a natural choice for diversity based improvement in retrieval performance.

5.3 Skylines with our Index

Retrieval systems compare database samples, one by one with the query for similarity. In principle this process can be quickened with the help of an index structure where relevant samples exist in the neighborhood of the query. Most of the similarity based retrieval schemes [5, 6, 80, 167] are able to efficiently retrieve similar samples with appreciable accuracy. These methods generally use comparison metrics which use all the attributes together to estimate the similarity between the query and the samples. As a result they tend to overlook the attribute to attribute relationships. Especially, when the attributes are numerical values, the results tend to get biased by the numerically dominant attributes. These also rely on expensive exhaustive comparisons for acceptable retrieval. Popular similarity indexing schemes for efficient retrieval like R - trees and SS - trees also depend on a fixed initial metric to build an efficient index. As a consequence these schemes work well only under the assumption of unambiguity in user queries. Schemes which can avoid exhaustive comparisons with minimal loss of accuracy should be preferred.

Skyline queries compare samples attribute-wise for dominance and eliminate redundancy. They are able to select diverse results out of a set of similar ones using attribute comparisons. They can thus be very interesting for answering ambiguous queries. But the attribute-wise computations are

prohibitive when performed exhaustively.

Motivated by these requirements, we adapt a multidimensional indexing scheme discussed in Section 2.5. Our scheme uses a B+ tree to index each individual attribute. The result of such an approach is that along every attribute the samples are arranged in a manner where the most similar ones are neighbors. This greatly enhances the chances of retrieving similar samples efficiently. We use our indexing optimally by retrieving t samples which are most similar to the query attribute from every tree. This approximation, as a function of t, saves a lot of computation with negligible loss in accuracy.

The t most similar samples suggested by each attribute then form the unique candidate list of skyline points. We have thus adapted the index structure which was designed for retrieving the approximate k nearest neighbors for reducing the skyline candidate samples to a much smaller number. We keep $t >> k$ to compensate for the approximation. The indexing scheme has been analyzed in detail for various characteristics in Section 2.5. The advantages of using this index structure are manifold. Its similarity based dimensional indexing allows for accurate approximate retrieval thereby improving the efficiency. The performance is also independent of the query and the metric as the index does not change.

5.4 CBIR using Skylines

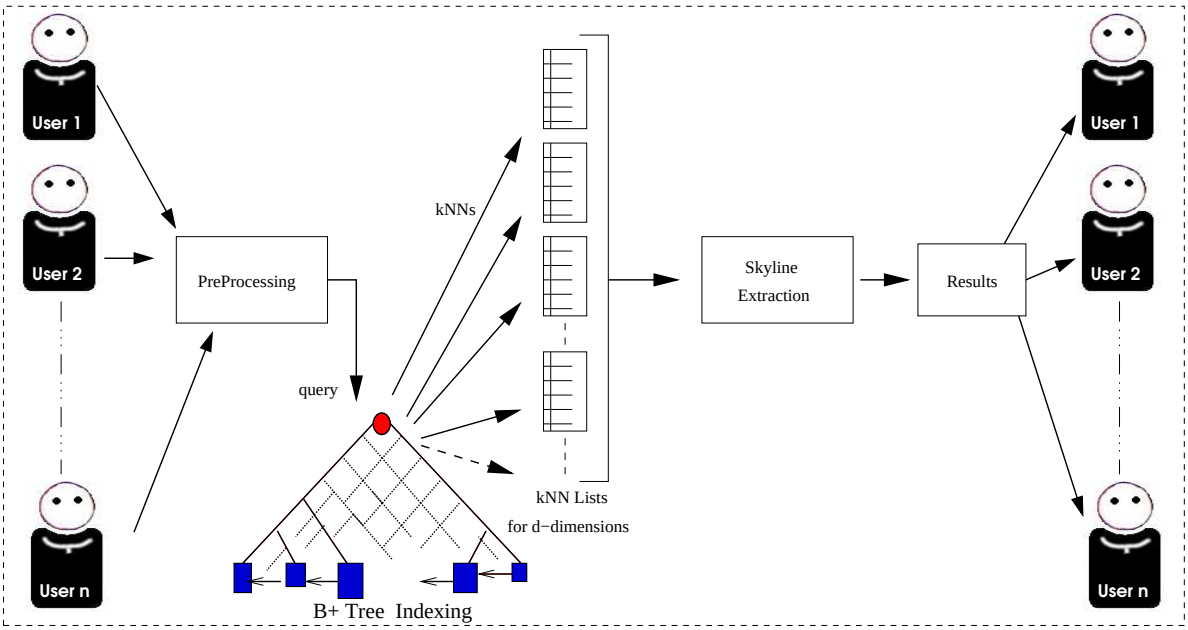


Figure 5.3: CBIR pipeline using Skylines

Briefly, our proposed approach for computing an approximate skyline consists of two important parts. First aspect is of selecting the set of samples on which the skyline will be computed. This set is required to consist of samples similar to the query as suggested by different attributes, independent of each other. Next, from this set of samples we compute the skyline in a single traversal of the list. The result is an efficiently computed skyline of the database with respect to the query, resulting in a diverse set of similar results.

Assume a query q is given then we are required to first compute its t nearest neighbors for each attribute or dimension. Each B+ tree is used to find the position where q would be inserted in

Require: Query as a feature vector q
Ensure: List **S** of images similar to the query in different concepts
  for all Dimensions, $d \in \mathbf{D}$ **do**
    Efficient Dimension-wise kNN Retrieval
    Merge them into the unique candidate list **C**
  end for
  Initialize **S** with the first candidate $c \in \mathbf{C}$
  for all Remaining $c \in \mathbf{C}$ **do**
    if c is not **dominated** by s, $\forall\, s \in \mathbf{S}$ **then**
      $\mathbf{S} = \mathbf{S} \cup c$
    end if
    if Any $s \exists \mathbf{S}$ is **dominated** by c **then**
      $\mathbf{S} = \mathbf{S} - \{s\}$
    end if
  end for

Algorithm 3: CBIR using Skylines

Require: Feature dimensions of data are indexed in **d** B+ trees
Ensure: List of **Skyline** candidates from the dimension d
  Get the k-NN to the query q in d

Algorithm 4: Efficient Dimension-wise kNN Retrieval

it. The data points k closest data points in the neighborhood of q are then retrieved. A detailed analysis of the way the index works can be found in Section 2.5. There can be overlap in the samples suggested by different attributes so we incrementally populate a unique list of dimension-wise most similar samples. This simple approach reduces the search space drastically by retrieving only the t most similar ones. Experiments show that our approach is an order of magnitude better in response time than the exhaustive approaches. Since we are retrieving the most similar t samples for that dimension, there is no approximation involved.

We prefer to call the list of unique samples accumulated by traversing over all attributes and selecting the closest t samples suggested by each, the *candidate* list. It consists of the attribute-wise closest samples and not necessarily the overall closest set. This set serves as our candidate set of similar samples from amongst which we will select the skyline and use it for populating the final retrieved set.

Now we have retrieved the candidate list by non-exhaustively traversing the dimensional trees. We then propose to use a simple dominant elimination approach for extracting the skyline from this candidate list. This method performs efficiently online rather than offline like the list generation approaches and space partitioning methods. But though, we present and validate our approach for skyline extraction which effectively utilizes our dimensional retrieval method, the skyline computation part of our framework is completely independent. This allows the use of any approach which can compute the skyline from a given set of such candidate data points.

In skyline computation the key idea is to eliminate the samples which are redundant in variety. Samples which are closer to the query in comparison to another sample over all attributes make the latter redundant or dominated. Non-dominance can similarly be explained as a condition where at least in one attribute the second sample is closer to the query than the first. Here dominance is a binary decision, i.e. either one dominates the other completely on all attributes or concepts or it

does not at all. Thus, if dominated the second or redundant sample should not be a part of the skyline. As a result the skyline approach results in a set of non-dominated samples. The elimination of the dominated samples retains samples which are similar to the query in varied characteristics. This results in a set of samples which are strictly non-redundant.

This strict criterion for dominance does not eliminate those samples which are practically the same as the other. For example, as an extreme case, consider two samples which are same along all but one attribute. The binary dominance decision considers them non-redundant and would retain them in the skyline. If the samples are very different, numerically far apart along this attribute, then they are non-redundant and show variety and both can be in the skyline. But if they are very similar, numerically close, then though they may not be redundant or dominated, they are not diverse either. Presence of such pairs of images leaves the skyline only will only be non-redundant but may not be completely diverse. This distinction between non-redundancy and diversity manifests itself more in dense neighborhoods especially where the attributes belong to a high precision continuous space, like real numbered visual features.

In order obtain a diverse set from this non-redundant set we propose a technique of neighborhood elimination. We argue that the samples which lie in the very close neighborhood of one another show visual and thus, conceptual redundancy, though they may not be completely redundant, as in skyline selection. We choose a representative from this dense group to remain in the skyline. as similarity is our guiding principle. Rest are considered as being dominated and are eliminated. We approximate the decision for a sample as lying inside or outside this neighborhood using a dissimilarity constraint. This radius is defined around the current point against which the other candidates are being compared. The radius of this perimeter,ϵ, can be computed based on the density in the hypersphere around the current point. It can also be heuristically estimated based on the conceptual and visual density of the dataset.

Thus, non-redundancy elimination serves our goal of selecting samples which are diversely similar. At the same time the effective use of the B+-tree based index allows us efficient sub-second retrieval making interactive response possible.

At the start of the skyline computation we assign the first candidate as a member of the skyline, by default. Over all the remaining samples the skyline is edited by adding and removing samples based on their satisfying the dominance requirements. Given the first candidate is assumed to be a part of the skyline to start with, the rest of the candidates are processed one by one. Every new candidate is checked with the samples which are already present in the skyline. If the new sample is not dominated by any of the existing members it is also included in the skyline, otherwise it is ignored. Also, if the new sample dominates any of the existing skyline members then the member is removed from the skyline. At the end we are left with a set of diversely similar samples. The approach requires a single traversal of the skyline candidate list for extracting the skyline.

Once the skyline list is ready, the subset which has to be presented to the user can be selected based on multiple factors. For example, the results could be presented ranked on similarity to the query if the goal is to retrieve similar samples and diversity is a novel advantage for enhancing user satisfaction. In a more real life example like one where a user is trying to reserve a hotel in a beach resort, the proximity to the beach and cost per day of the hotel both are important factors. And there will be some hotels which are better at cost while some are better in proximity. So the entire skyline set is interesting but the particular user may be more interested in beach view than cost of stay so the skyline could be presented sorted on proximity to the beach. The approximated skyline extraction can also be used for finding the representative sample set of the database showing the diversity of the data.

5.5 Learning the User Skyline

Skylines in literature have been considered to be objective or global in nature. This essentially means that for a given query on a said database there is only one set of skyline results. So when the user queried with the building and car image in Figure 5.1 he would probably get the set of only cars or buildings or both, always irrespective of his personal or subjective preferences. For example, if the user wishes to see cars only, or buildings only, it is not supported in the current approach for skyline extraction.

We present a novel solution to such subjectiveness in this thesis. Our proposed approach does not only allow improved retrieval for the current user but also allows the system to capture trends of preference in diverse results. It allows the system to learn, for example, that most people looking for buildings with a car in front are actually interested in the building and not the car. In the literature example in [160] of the hotel price and distance from beach, we can now suggest a refined list if the user prefers proximity to the beach over inexpensive accommodation.

We make effective use of our B+-tree based indexing scheme for efficiently incorporating preferences in skylines. User preferences are absorbed in a relevance feedback scheme where he marks the images he finds relevant to his query from among the diverse set presented to him. We use our feature relevance learning techniques discussed in Section 2.7 to learn the preference of the attributes for the user in terms of relative importance or weights for the attributes describing the images. The interpretation is that if the user prefers a certain attribute then he wants it to be predominantly present in the retrieval results or in other words he wants to see more images with that attribute. We propose a novel and simple approach for including this preference.

We argue that the weights of the attributes essentially modify each dimension to a user-centric one. We call this modified, shrunk or stretched dimension as a semantic dimension where the user's query is related by a Euclidean metric over attributes. As a consequence, in order to extract a user preferred skyline we should use this modified feature space. It will automatically result in a skyline where dominance is adapted according to the importance of the dimension. We keep the number of samples t from each dimension constant. As a consequence of this modification to the space, more similar samples from the preferred dimension or attribute appear at the top of the dissimilarity ranked set of results.

Weights are incrementally learned as discussed in Section 3.3. The learned preferences in weights are used for iteratively tuning the retrieval to the user's needs. The user query is known to be vague at the start and refines iteratively with results. Our approach is therefore able to provide iteratively focused results to the user query. In context of our 'car and building' example this means that over iterations more and more similar 'building' images are presented and others are discounted slowly. Such an inexpensive approach for learning user preferences, quickly in a few iterations improves user experience in general.

5.6 Results and Discussions

5.6.1 Implementation and Data Sets

To validate our claims on the capabilities of our proposed ideas with approximate skylines, we have conducted extensive experiments. We have used both synthetically generated databases and also those extracted from real images. Through our experiments we showcase how our approach saves significantly in response time at a small cost in accuracy.

For synthetic data we generated random attribute values for different database sizes and different number of attributes. They range from 1000 to 10000 samples and 5 to 50 attributes. For the real

database we used a set of real images collected from various sources including, Flickr Photos, COREL set, Caltech image sets and cartoon video splitting. This real database consists of 11901 images from 58 different categories like animals, sea images, ships, trains, fruits, airplanes, bicycles etc. This set is a highly mixed set of images which poses a perfect challenge for image retrieval algorithms and features. Then for one of our sets we extract the first three color moments namely, mean, variance and skew to form a 9 dimensional attribute space. For another experiment we extract the 12 dimensional MPEG-7 color layout descriptor from these 11901 images. We have conducted all our experiments on a machine with an AMD Dual Core 2GHz processor and 4GBs of RAM running on Fedora Core Linux platform.

Dominance is ideally discernible in discrete space, where it is easy to judge the sample as being close and far. In multimedia data the features are normally high precision values and the space is generally dense. As a result very similar samples tend to lie close together in tight clusters. These samples differ just enough along some of the many attributes to ensure all can exist in the skyline. These little differences along one of more attributes allows them to be non-dominated, though they are very similar. Skylines alone ensure non-redundancy, but diversity can still be absent. We use a neighborhood elimination technique for removing most of the very similar skyline members. We define ϵ as a small threshold distance from the current point. The only non-dominated member from this set, which includes the current point, is the one most similar to the query, overall. Rest are considered dominated and are not considered in the skyline. In our case we have empirically defined $\epsilon = 0.003746$. The value should be estimated based on the precision of the feature values and the average separation between any two images.

We have conducted experiments to evaluate the performance in terms of response time and how close is the approximate skyline to the one computed with the exhaustive approach. The exhaustive skyline works as our benchmark in all these experiments. We then vary t, the number of samples retrieved from every attribute and compute the skyline. We compute the number of skyline points from this approximation which *matched* the actual skyline. We also compute the number of points in the exhaustive skyline which are *missed* in this approximation and if there are any *extra points* which are getting listed in the present skyline but are not a part of the exhaustive skyline. We vary t in steps till the exhaustive retrieval condition is reached and we do this for different combinations of the database size and dimensions for both real and synthetic databases.

5.6.2 Response Time Vs Accuracy

We first present the results of our experiments highlighting the comparative gain achieved in terms of response time Vs the accuracy of the skyline. First in Figures 5.4, 5.5 and 5.6 we use synthetic databases of 10 attributes and 5000, 10000 and 15000 samples each respectively. We compute the skyline over increasing values of t, the number of samples from every attribute tree. The unique list generated after every round of attribute traversal is used for computing the skyline as explained in Section 5.4 above. In this experiment we go on incrementing t till the size of the unique candidate list becomes equal to the database size, thereby mimicking the exhaustive skyline computation. We note the response times and the error in the estimated skyline at every t step. Error is measured as the number of samples missed by the present instance in comparison to the exhaustive skyline set. We also add to it the number of extra skyline suggested by this step which do not occur in the exhaustive set. We regard both of these are overall erroneous judgments by the system and measure them as a *percentage error* compared to the exhaustive skyline. We also observe the response time for each step as a percentage of the time required to compute the exhaustive skyline. In Figures 5.4, 5.5 and 5.6 we plot the percentage error and time against t.

Next we present similar experiments which we conducted using our 11901 real image database.

We extracted popular visual features from the images and used them for experiments. We have included the graphs for the behavior of the system when presented with these real sets in Figure 5.7 and Figure 5.8.

As is evident from the graphs in Figures 5.4, 5.5, 5.6, 5.7 and 5.8 the error in skyline estimation reduces to negligible values at t values much lower than the exhaustive value in all cases. This convergence happens in a fraction of the time required for computing the exhaustive skyline. This gain in response time validates our claim on our proposed approximate skyline computation method. The results show the benefit the approximate k-NN method brings to the estimation of the skyline. Our simple single traversal approach for computing the skylines thereafter also integrates seamlessly with our indexing framework resulting in commendable gain in efficiency at negligible cost on accuracy, if any.

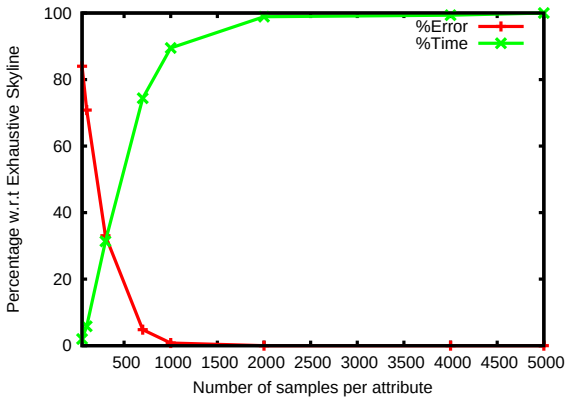


Figure 5.4: Percentage performance compared to an exhaustive skyline of 2872 samples over a synthetic database of 5000 samples with 10 attributes

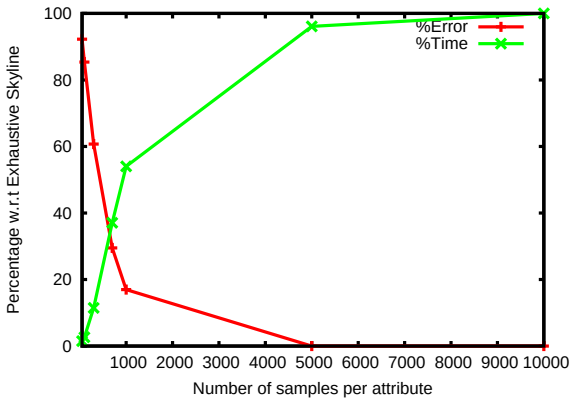


Figure 5.5: Percentage performance compared to an exhaustive skyline of 5095 samples over a synthetic database of 10000 samples with 10 attributes

5.6.3 Scalability

We claim that our approximate skyline approach is scalable with database size. To validate this claim we conducted accuracy experiments like the ones above with multiple database sizes. To contain the free parameters we kept the number of attributes constant at 10 while the database

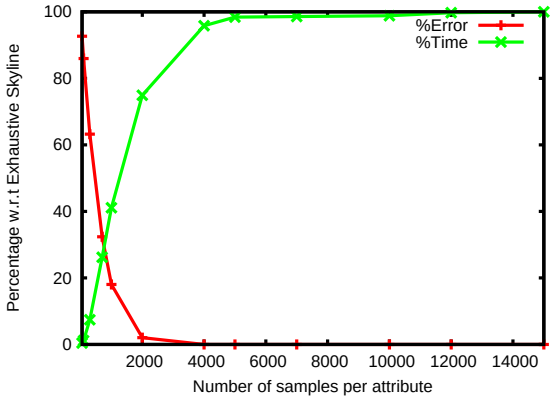


Figure 5.6: Percentage performance compared to an exhaustive skyline of 6571 samples over a synthetic database of 15000 samples with 10 attributes

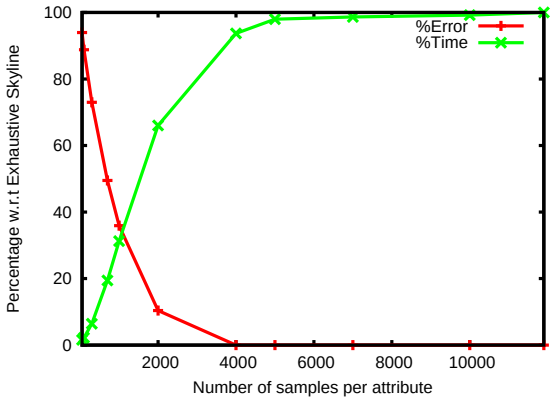


Figure 5.7: Percentage performance compared to an exhaustive skyline of 3832 samples over a real database of 11901 samples with 9 attributes

size varied from 5000 samples up to 20000 samples. To make it easier to appreciate the performance of our proposed algorithm we use a unique method of presentation. We assume that for all practical purposes 10% error in skyline computation over large databases is acceptable if the corresponding gain in time performance is appreciable. In Figure 5.9 we plot the percentage of corresponding exhaustive time used against the size of the database to perform with 10% error in skyline. As per our expectations the response time reduces dramatically from 70% for 5000 samples to 51% for 20000 samples. This validates our claim on the advantages of our proposed approximate skyline algorithm.

5.6.4 Diversity in Similarity

We next include results of some of the queries presented to our skyline based technique for diversely similar retrieval. As can be seen in Figure 5.2, Figure 5.10 our approach retrieves images with considerable diversity in similarity as compared to the pure similarity based CBIR approach. These visual results convincingly prove the ability of our approach to retrieve with diversity while still retaining similarity among results.

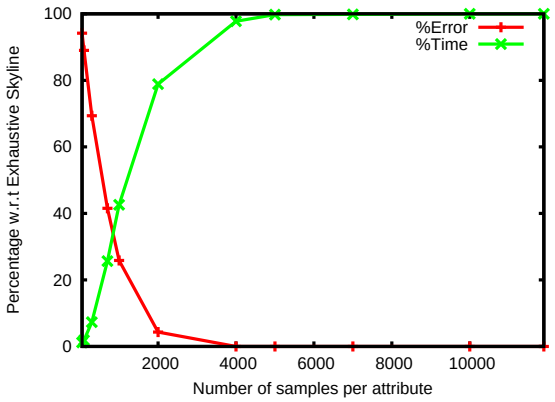


Figure 5.8: Percentage performance compared to an exhaustive skyline of 6723 samples over a real database of 11901 samples with 12 attributes

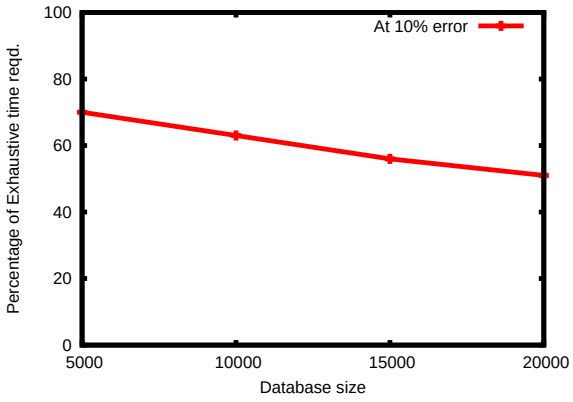


Figure 5.9: Percentage of time required. for maintaining 10% accuracy for increasing database sizes

5.6.5 Preferential Skylines

We conducted an interesting experiment to show the capabilities of our proposed approach to effectively handle preferences in skyline computation. We presented a query to the system, then we provided two different patterns of relevance feedback on the results, each favoring a subset of the diverse initial results. We have included the initial diverse set and the two subsequent result sets, one for each feedback pattern in Figure 5.11. We also repeated the same experiment with another query and the results are included in Figure 5.12. As can be seen in the figure the initial diversely similar result set is narrowed down to the two similar sets after the feedback rounds. These results clearly show the capability of our approach to retrieved preferred skylines based on user feedback.
We have introduced the concept of preferred skylines with some interesting results on a couple of real natural images. We believe extensive evaluation of learning in skylines would be an interesting direction for future research.

5.6.6 Limitations of Skylines

We have conducted numerous experiments on varied database sizes and number of attributes. During some of the experiments we have been able to observe scenarios where skylines may not be useful and advantageous. Out of the many experimental instances we include a few here and

(a)

(b)

(c)

(d)

Figure 5.10: Comparison of skyline based results with k-NN for different queries. Top few results are compared between CBIR(row one) and our skyline approach(row two). The highlighted images are the queries. Skylines can select and discard different number of similar results. As a result we see a variation in the number of results presented for different queries.

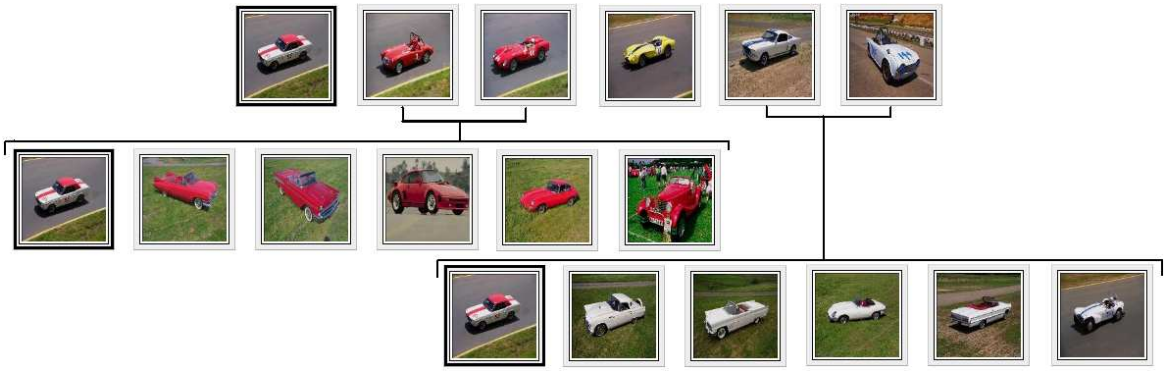


Figure 5.11: Results of a query 'car' with diversity in first iteration (row 1). Next (row 2) we show results of positive feedback to 'red cars' in the initial set. Next row (row 3) shows results of positive feedback to 'white cars' in the initial iterations. Due to space constraints we show only the top few images, feedback is provided to a larger set of retrieved results.

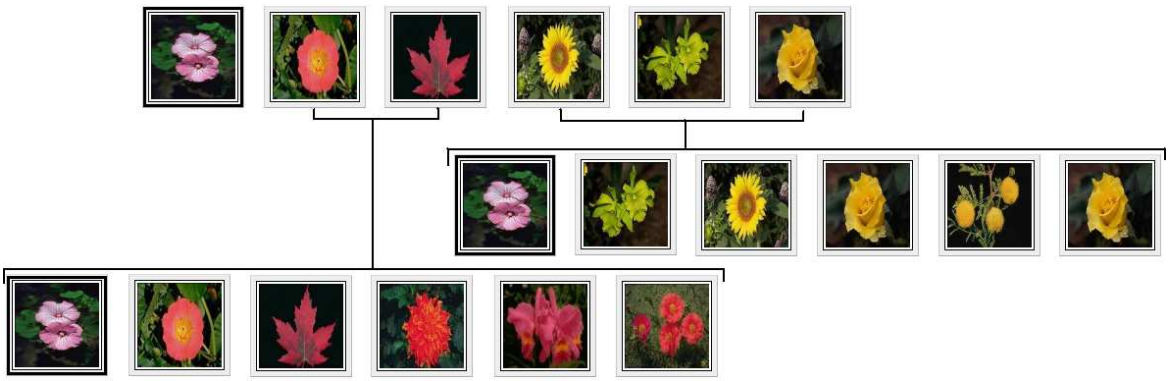


Figure 5.12: Results of the first iteration for a 'flower' query are in first row. Then (row 2) we show results of positive feedback to 'yellow flowers' in the set in the first iteration. Next row (row 3) shows results of positive feedback to 'pink flowers' in the first iteration. Only few images are shown while the retrieved and feedback set is larger.

analyze their behavior to validate our inference.

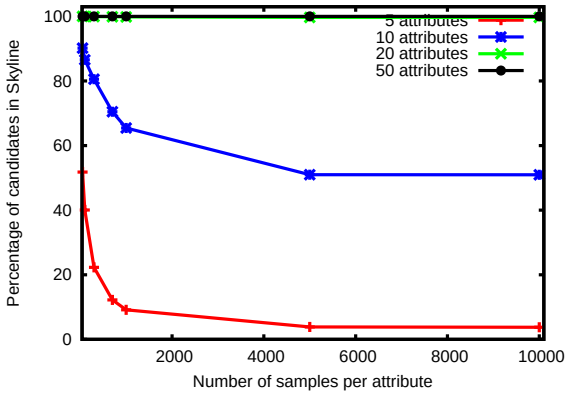


Figure 5.13: Graph shows the percentage overlap of the candidates with the skyline for a 10000 sample synthetic database

We conducted a set of experiments where we kept the database size constant at 10000 samples. We then varied the number of attributes from 5 to 50 and observed the behavior of skylines. This variation in the number of attributes mimics the "descriptiveness" of the database. For each variation in the number of attributes, we estimated the candidate lists and from them the skylines for increasing t values. We then observed the relationship between the sets for the different t values. In the case of 5 attributes we found that as per our expectations as t increases the size of the candidate list goes on increasing. The size of the skyline set increases but slowly lags behind it and over larger t values becomes constant. This is as expected and means that at any given t value only some of the candidates become members of the skyline. The overlap between the candidate list and the skyline is large for smaller t and decreases as t increases.

We observed a different behavior when the number of attributes was increased. With large number of attributes nearly all the members of the candidate list become members of the skyline, irrespective of the change in t. We have plotted this observation for a few of the attribute sizes in Figure 5.13. The figure shows the percentage overlap between the candidate and the skyline sets. We plot these for different number of attributes namely, 5, 10, 20 and 50. As can be observed from the figure for a lower number of attributes, 5 and 10, the skyline set is a visible fraction of the candidate set. But as the number of attributes increases to 20 and then 50 in our experiment the overlap reaches nearly 100%!.

5.7 Summary

We have proposed a novel method for efficiently retrieving diversely similar results for a given query. We use skyline queries to achieve the same. We have used our indexing scheme to compute skylines efficiently in interactive time. We have shown the efficacy and performance of our approach with extensive experiments using both real and synthetic dataset. We have also presented an analysis of when skylines are useful and when they are not. We have proved the idea of using skylines to efficiently achieve diversity in similarity image retrieval using visual and empirical results. We have also introduced the notion of preferred skylines, where the skyline is tuned to the user's intent.

Chapter 6

Conclusions

The volume of user generated content on the web, in terms of personal images and videos, is continuously increasing. Unreliable nature of the few tags assigned by users makes accurate tag based retrieval infeasible. These have led to a renewed interest in content based image retrieval, which had taken a backseat with the success of text search. Content based image retrieval systems rely on low level representations for efficiency and try to support complex semantics using learning.

Content based image retrieval essentially relies on visual characteristics extracted automatically from the images. These machine centric features allow efficient indexing and retrieval. However, their ability to capture human visual perception of semantics is limited. Relevance feedback based user interaction techniques have been found to be very useful to bridge the semantic gap. Sophisticated use of such techniques has been hindered primarily due to the extended user engagement required. Complementary paradigm of using only relationships based on co-relevance has been explored to bypass the dependency on features. Absorption of logs of similar feedback patterns in history of use can also be useful in reducing the number of user iterations. These techniques are not very popular in image retrieval either, primarily due to their critical dependence on large amount of logs. Learning techniques have also suffered because of the sparsity of feedback owing to the small number of samples ever marked by any user. The user is understood to be vague in his presentation of the query. This is aggravated with the *polysemy* related issues in image retrieval. This necessitates that the user be presented with a set of results which are learner-centric but are useful to the user also. Active learning studies methods of presenting such learner-centric set to the user. In this thesis, we have proposed our solutions to some of these important issues in image retrieval.

In Chapter 2, we have presented **FISH**, a system for interactive image retrieval from large collections. Our CBIR system, **FISH** has been designed with certain fundamental real-world utility perspectives in mind. These include efficiency, scalability, acceptable accuracy, ease of querying and interaction, seamlessly extensible modular design. It is able to effortlessly scale to millions and retrieve in less than a second. We use an efficient multi-dimensional indexing scheme for interactive retrieval. It also supports learning allowing it to adapt to the user needs for the current and subsequent users. We have also shown experimentally that the approximation error in accuracy is negligible and reduces further with learning. We have validated the optimization of using the approx. k-NN and also retrieving only from the most important few dimensions. Chapter 2 discusses various aspects of our system in depth.

In Chapter 3, we discuss our novel approach for optimally using expensive user feedback across users. This approach of long term learning allows us to incrementally learn the popular concept in the images in the database. It is then used for improved retrieval. We are also able to use it for

extracting visual content from the images. We also showed how we efficiently implement learning on top of our indexing scheme.

Realizing the weakness of over dependence on low level visual features and motivated by the ever increasing user interactions on the Internet, we proposed an integrated framework for image retrieval. Our Bayesian inference approach integrates user interaction logs based relationships as *a priori* knowledge with visual feature based similarity evidence to avoid the cold start and reduce the semantic gap in image retrieval. We efficiently maintain and update logs. We also discuss our idea of discovering semantic concepts using the logs. We present the usefulness of our approach both qualitatively and quantitatively in Chapter 4 using different datasets.

User interaction, especially in the form of implicit or explicit feedback given by him, is the key for successful image retrieval in future. This warrants improved interaction with the user in terms of both quantity and quality of feedback. Incorporation of more human-centric features can help improve the user experience, but the quality of results and their meaningful presentation is the key to user persistence. Knowing well that the user query is ambiguous, presenting a pure similarity solution may be to restrictive. In Chapter 5, we propose to present a diversified set of results to him. We propose to present images which are all similar to the query but in different ways. This would help the user quickly narrow down to his intended concept comfortably. This reduced burden on the user to ensure that he provides the feedback with utmost care encourages him to interact more with the system. This results in better learning, and hence retrieval accuracy.

In this thesis, we have been able to build an interactive image retrieval system which effortlessly scales to millions of samples while retrieving in less than a second. We have also incorporated efficient learning schemes, both using feature relevance and semantic relationships from logs. We also proposed a unique method for extracting an inherently diverse set of similar results to the user's query. These characteristics of our proposed solutions for an image retrieval system improve user experience and as a result, the system's utility as a real world system.

6.1 Scope for future work

CBIR is a diverse field of study. We have tried to provide our solutions to some of the most critical problems in CBIR in this thesis but much is left to be explored outside and beyond the purview of this work. Aspects in CBIR research like the development of better representation schemes was considered outside the focus of this work. While analyzing the many algorithms and results proposed in this thesis, some constraints and some future directions of interest were noticed. We close this compilation with some such interesting possibilities for the future.

Content based access to videos can be considered as the next step for most of the algorithms designed for images. Videos are more popular as they belong to a richer class of media. Videos are generally abstracted into set of representative frames. These are then used for content based management. This abstraction into a set of frames makes them very similar in algorithmic requirements to images. The difference is generally at the level of the matching scheme and not the index structure. The schemes need to impose constraints like chronological ordering, temporal separations etc. We therefore, look forward to adapting the **FISH** system for content based video retrieval.

Continuously evolving nature of user generated content in today's online collections, necessitates use of highly dynamic indexing schemes. Our index in **FISH** can handle additions to some extent. Migrating to thousands of concurrent additions (content uploads) and deletions (content removals) will require further development. All these updates will have to be supported in real time (online).

We would also like to explore further the ideas of concept discovery and extraction using learn-

ing. These if refined further could allow automatic interpretation of visual content, possibly even without search based efforts. We would also like to explore its direct consequence on content aware recommendation systems which mostly use behavior patterns only, as of now. This would require advancements in our proposed Bayesian inference approach by incorporating more complex relationships, possibly using Bayes nets or something similar. Most of the present work interprets using low level features but in future we would like to learn higher level semantics using low level features.

A shift from machine-centric to user-centric design of image retrieval systems is required. These systems would improve user experience by better meeting his expectations. Such ideas would include measures like diversity in results, ease of feedback, preferably implicit feedback, etc. We found a unique approach for diversity by using skylines with CBIR. But diversity again is a subjective decision so we would like to extend our initial discussions on preferential skylines with quantitative evaluations of performance and more robust formulations.

FISH is a CBIR system which uses query-by-example and operates on numeric features which represent visual content. Through this thesis, we have shown learning based methods for achieving better performance using features alone. We feel that the popularity of user generated content on the web is growing and more reliable tags are available from the numerous collaborative applications. Querying with text is more common, primarily because text search is used more, and secondly because a user may not always have an image available to query with. These factors suggest a convergence of retrieval paradigms of text and visuals. We strongly feel that in a not-so-distant-future, systems like **FISH** will be able to accept a query in text, interpret it into a visual example and respond with results useful for a wide class of users, casual surfers to serious searchers.

Related Publications

The work in my thesis has been disseminated to the following conferences.

- Pradhee Tandon, Piyush Nigam, Vikram Pudi, C. V. Jawahar, "**FISH: A Practical System for Fast Interactive Image Search in Huge Databases**", in *Proceedings of the 7th ACM International Conference on Image and Video Retrieval (CIVR '08)*, July 6-8, 2008, Niagara Falls, Canada.
- Pradhee Tandon, C. V. Jawahar, "**Long Term Learning for Content Extraction in Image Retrieval**", in *Proceedings of the 15th National Conference on Communications (NCC '09)*, January 16-18, 2009, Guwahati, India.
- Pradhee Tandon, C. V. Jawahar, "**Bayesian Image Retrieval**" under submission to *3rd International Conference on Pattern Recognition and Machine Intelligence (PReMI '09)*, December 16-20, 2009, New Delhi, India.
- Pradhee Tandon, Vikram Pudi, C. V. Jawahar, "**Diversity in Retrieval with Skyline Queries**" under submission to *26th IEEE International Conference on Data Engineering (ICDE 2010)*, March 1-6, 2010, Long Beach, California, USA.
- Pradhee Tandon, C. V. Jawahar, "**Bayesian Image Retrieval: integrating logs with similarity**" under submission to *3rd International Conference on Data Mining (ICDM '09)*, December 6-9, 2009, Miami, Florida, USA.

Bibliography

[1] *Flickr Photo Sharing,* `http://flickr.com`.

[2] *Picasa Web Albums,* `http://picasaweb.google.com`.

[3] *Facebook,* `http://www.facebook.com`.

[4] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[5] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Special Interest Group on Management Of Data Conf*, 1984, pp. 47–57.

[6] D. A. White and R. Jain, "Similarity indexing with the ss-tree," in *Proc of the Intl Conf on Data Engineering*, Washington, DC, USA, IEEE Computer Society, 1996, pp. 516–523.

[7] N. Jammalamadaka, V. Pudi, and C. V. Jawahar, "Efficient search with changing similarity measures on large multimedia datasets," in *Intl MultiMedia Modeling Conf*, 2007, pp. 206–215.

[8] R. C. Veltkamp and M. Tanase, "Content-based image retrieval systems: A survey," 2000.

[9] N. Vasconcelos, "Content-based retrieval from image databases: current solutions and future directions," in *In International Conference in Image Processing (ICIP01)*, 2001, pp. 6–9.

[10] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, 2000.

[11] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 1, pp. 1–19, 2006.

[12] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Survey*, vol. 40, no. 2, pp. 1–60, 2008.

[13] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The qbic project: Querying images by content, using color, texture, and shape," in *Storage and Retrieval for Image and Video Databases*, 1993, pp. 173–187.

[14] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu, "Virage image search engine: An open framework for image management," in *Storage and Retrieval for Image and Video Databases (SPIE)*, 1996, pp. 76–87.

[15] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233–254, 1996.

[16] V. E. Ogle and M. Stonebraker, "Chabot: Retrieval from a relational database of images," *IEEE Computer*, vol. 28, no. 9, pp. 40–48, 1995.

[17] J. R. Smith and S.-F. Chang, "Visualseek: A fully automated content-based image query system," in *ACM Multimedia*, 1996, pp. 87–98.

[18] J. R. Smith and S.-F. Chang, "Visually searching the web for content," *IEEE MultiMedia*, vol. 4, no. 3, pp. 12–20, 1997.

[19] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang, "Supporting similarity queries in mars," in *ACM Multimedia*, 1997, pp. 403–413.

[20] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, no. 5, pp. 644–655, Sep 1998.

[21] C. Nastar, M. Mitschke, C. Meilhac, and N. Boujemaa, "Surfimage: A flexible content-based image retrieval system," in *ACM Multimedia*, 1998, pp. 339–344.

[22] W.-Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in *ICIP (1)*, 1997, pp. 568–571.

[23] S. Ravela and R. Manmatha, "Retrieving images by appearance," in *ICCV*, 1998, pp. 608–613.

[24] *Google Image Search,* `http://images.google.com`.

[25] *Yahoo Image Search,* `http://images.search.yahoo.com`.

[26] L. Zhang, L. Chen, M. Li, and H. Zhang, "Automated annotation of human faces in family albums," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, New York, NY, USA, ACM, 2003, pp. 355–358.

[27] Z. C. Z. Wang and D. Feng, "Fuzzy integral for leaf image retrieval," *IEEE International Conference on Fuzzy Systems*, 2002.

[28] A. Csillaghy, H. Hinterberger, and A. O. Benz, "Content-based image retrieval in astronomy," *Inf. Retr.*, vol. 3, no. 3, pp. 229–241, 2000.

[29] J. Dozier, D. A. Roberts, R. E. Davis, T. H. Painter, and R. O. Green, "Retrieval of subpixel snow-covered area and grain size from imaging spectrometer data," *Remote Sens. Env*, vol. 85, no. 1, pp. 64–77, 2003.

[30] M. Schrö, H. Rehrauer, K. Seidel, and M. Datcu, "Interactive learning and probabilistic retrieval in remote sensing image archives," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 38, pp. 2288–2298, 2000.

[31] *Airliners.net Aviation Photo Search Engine,* `http://www.airliners.net/search/`.

[32] *Global Memory Net,* `http://www.memorynet.org`.

[33] *Terragalleria,* `http://www.terragalleria.com`.

[34] J. Li and J. Z. Wang, "Real-time computerized annotation of pictures," in *Proceedings of the ACM International Conference on Multimedia*, New York, NY, USA, ACM, 2006, pp. 911–920.

[35] J. Li and J. Z. Wang, "Real-time computerized annotation of pictures," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 985–1002, 2008.

[36] R. Datta, Z. Zhuang, W. P. Weiss, M. Friedenberg, J. Li, D. Joshi, and J. Z. Wang, "Paragrab: A comprehensive architecture for web image management and multimodal querying," in *VLDB*, 2006, pp. 1163–1166.

[37] A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-based image matching using angular partitioning," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 35, no. 1, pp. 28–41, 2005.

[38] J. Assfalg, A. D. Bimbo, and P. Pala, "Three-dimensional interfaces for querying by example in content-based image retrieval," *IEEE Transactions on Vis. Comput. Graph.*, vol. 8, no. 4, pp. 305–318, 2002.

[39] T. Käster, M. Pfeiffer, and C. Bauckhage, "Combining speech and haptics for intuitive and efficient navigation through image databases," in *ICMI*, 2003, pp. 180–187.

[40] Y. Fang, D. Geman, and N. Boujemaa, "An interactive system for mental face retrieval," in *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, New York, NY, USA, ACM, 2005, pp. 193–200.

[41] A. Jaimes, K. Omura, T. Nagamine, and K. Hirata, "Memory cues for meeting video retrieval," in *CARPE'04: Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences*, New York, NY, USA, ACM, 2004, pp. 74–85.

[42] T. Nagamine, A. Jaimes, K. Omura, and K. Hirata, "A visuospatial memory cue system for meeting video retrieval," in *ACM Multimedia*, 2004, pp. 752–753.

[43] Q. Zhang, S. A. Goldman, W. Yu, and J. Fritts, "Content-based image retrieval using multiple-instance learning," in *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 2002, pp. 682–689.

[44] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statictical Learning.* Springer-Verlag, 2001.

[45] N. Panda and E. Y. Chang, "Efficient top-k hyperplane query processing for multimedia information retrieval," in *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, New York, NY, USA, ACM, 2006, pp. 317–326.

[46] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *J. Mach. Learn. Res.*, vol. 5, pp. 913–939, 2004.

[47] N. Sebe, M. S. Lew, and D. P. Huijsmans, "Toward improved ranking metrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1132–1143, 2000.

[48] G. Wu, E. Y. Chang, and N. Panda, "Formulating context-dependent similarity functions," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, New York, NY, USA, ACM, 2005, pp. 725–734.

[49] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Systems*, vol. 8, pp. 536–544, April 2003.

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

[51] Y. Wu, Q. Tian, and T. S. Huang, "Discriminant-em algorithm with application to image retrieval," in *CVPR*, 2000, pp. 1222–1227.

[52] Y. Rui and T. S. Huang, "Optimizing learning in image retrieval," in *CVPR*, 2000, pp. 1236–1238.

[53] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, New York, NY, USA, ACM, 2001, pp. 107–118.

[54] K.-S. Goh, E. Y. Chang, and W.-C. Lai, "Multimodal concept-dependent active learning for image retrieval," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, New York, NY, USA, ACM, 2004, pp. 564–571.

[55] J. He, H. Tong, M. Li, H.-J. Zhang, and C. Zhang, "Mean version space: a new active learning method for content-based image retrieval," in *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, New York, NY, USA, ACM, 2004, pp. 15–22.

[56] C. Yang, M. Dong, and F. Fotouhi, "Semantic feedback for interactive image retrieval," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, New York, NY, USA, ACM, 2005, pp. 415–418.

[57] C.-H. Hoi and M. R. Lyu, "A novel log-based relevance feedback technique in content-based image retrieval," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, New York, NY, USA, ACM, 2004, pp. 24–31.

[58] D.-H. Kim and C.-W. Chung, "Qcluster: relevance feedback using adaptive clustering for content-based image retrieval," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, New York, NY, USA, ACM, 2003, pp. 599–610.

[59] X. S. Zhou and T. S. Huang, "Small sample learning during multimedia retrieval using biasmap," in *CVPR (1)*, 2001, pp. 11–17.

[60] M. Nakazato, C. Dagli, and T. Huang, "Evaluating group-based relevance feedback for content-based image retrieval," 2003, pp. II: 599–602.

[61] H. Wu, H. Lu, and S. Ma, "Willhunter: Interactive image retrieval with multilevel relevance measurement," in *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, Washington, DC, USA, IEEE Computer Society, 2004, pp. 1009–1012.

[62] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, "The bayesian image retrieval system, pichunter: Theory, implementation and psychophysical experiments," *IEEE transactions on image processing*, vol. 9, pp. 20–37, 2000.

[63] Z. Su, H. Zhang, S. Z. Li, and S. Ma, "Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 924–937, 2003.

[64] N. Vasconcelos and A. Lippman, "Learning from user feedback in image retrieval systems," in *in Proc. of NIPS'99*, MIT press, 1999.

[65] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang, "A unified framework for semantics and feature based relevance feedback in image retrieval systems," in *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, New York, NY, USA, ACM, 2000, pp. 31–37.

[66] X. S. Zhou and T. S. Huang, "Unifying keywords and visual contents in image retrieval," *IEEE MultiMedia*, vol. 9, no. 2, pp. 23–33, 2002.

[67] J. Amores, N. Sebe, P. Radeva, T. Gevers, and A. Smeulders, "Boosting contextual information in content-based image retrieval," in *MIR '04: Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, New York, NY, USA, ACM, 2004, pp. 31–38.

[68] F. Jing, M. Li, H. Zhang, and B. Zhang, "A unified framework for image retrieval using keyword and visual features,"

[69] J. Laaksonen, M. Koskela, and E. Oja, "Picsom-self-organizing image retrieval with mpeg-7 content descriptors," *Neural Networks, IEEE Transactions on*, vol. 13, no. 4, pp. 841–853, 2002.

[70] P. Wu and B. S. Manjunath, "Adaptive nearest neighbor search for relevance feedback in large image databases," in *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, New York, NY, USA, ACM, 2001, pp. 89–97.

[71] J. Han, K. N. Ngan, M. Li, and H. Zhang, "A memory learning framework for effective image retrieval," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 511–524, 2005.

[72] J. Zhang, X. Zhou, W. Wang, B. Shi, and J. Pei, "Using high dimensional indexes to support relevance feedback based interactive images retrieval," in *Proc of the Intl Conf on Very Large Data Bases*, VLDB Endowment, 2006, pp. 1211–1214.

[73] E. Louupias and S. Bres, "Key points-based indexing for pre-attentive similarities: The kiwi system," *PAA*, vol. 4, no. 2/3 2001, pp. 200–214, 2001.

[74] W.-Y. Ma and B. S. Manjunath, "Netra: a toolbox for navigating large image databases," Secaucus, NJ, USA, 1999.

[75] J. M. Martínez, "Mpeg-7: Overview of mpeg-7 description tools, part 2," *IEEE MultiMedia*, vol. 9, no. 3, pp. 83–93, 2002.

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

[77] D. Heisterkamp, "Building a latent semantic index of an image database from patterns of relevance feedback," *Pattern Recognition, 2002. Proc. 16th Intl Conf on*, vol. 4, pp. 134–137 vol.4, 2002.

[78] M. G. P. Cord, "Image retrieval using long-term semantic learning," *Intl Conf on Image Processing*, pp. 2909–2912, 8-11 Oct. 2006.

[79] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *J. Intell. Inf. Syst.*, vol. 3, no. 3-4, pp. 231–262, 1994.

[80] R. Kurniawati, J. S. Jin, and J. Shepherd, "Efficient nearest-neighbour searches using weighted euclidean metric," in *Proc of the British National Conf on Databases*, London, UK, Springer-Verlag, 1998, pp. 64–76.

[81] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Proc of the Intl Conf on Visual Information and Information Systems*, 1999, pp. 509–516.

[82] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. S. Huang, "Supporting similarity queries in MARS," in *ACM Multimedia*, 1997, pp. 403–413.

[83] E. D. Sciascio, G. Mingolla, and M. Mongiello, "Content-based image retrieval over the web using query by sketch and relevance feedback," in *Proc of the Intl Conf on Visual Information and Information Systems*, London, UK, Springer-Verlag, 1999, pp. 123–130.

[84] E. Kasutani and A. Yamada, "The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval," in *Intl Conf on Image Processing*, 2001, pp. I: 674–677.

[85] D. Messing, P. van Beek, and J. Errico, "The mpeg-7 colour structure descriptor: Image description using colour and local spatial information," in *Intl Conf on Image Processing*, 2001, pp. I: 670–673.

[86] D. K. Park, Y. S. Jeon, and C. S. Won, "Efficient use of local edge histogram descriptor," in *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, New York, NY, USA, ACM, 2000, pp. 51–54.

[87] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence," in *ECCV (3)*, 2002, pp. 255–271.

[88] T. Huang and X. S. Zhou, "Image retrieval with relevance feedback: from heuristic weight adjustment to optimal learning methods," *Proc. of the Intl Conf on Image Proc.*, vol. 3, pp. 2–5, 2001.

[89] H. Muller, W. Muller, D. Squire, S. Marchand-Maillet, and T. Pun, "Long-term learning from user behavior in content-based image retrieval," *Tech. Rep. 00.04*, March 2000.

[90] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "Picsom - content-based image retrieval with self-organizing maps," *Pattern Recogn. Lett.*, vol. 21, no. 13-14, pp. 1199–1207, 2000.

[91] A. Shah-Hosseini and G. M. Knapp, "Learning image semantics from users relevance feedback," in *Proceedings of the ACM International Conference on Multimedia*, New York, USA, 2004, pp. 452–455.

[92] P.-Y. Yin, K.-C. Chang, and A. Dong, "Integrating relevance feedback techniques for image retrieval using reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1536–1551, 2005. Fellow-Bhanu, Bir.

[93] A. Celentano and E. D. Sciascio, "Feature integration and relevance feedback analysis in image similarity evaluation," *Journal of Electronic Imaging*, vol. 7, pp. 308–317, 1998.

[94] H. Müller, W. Müller, S. Marchand-Maillet, S. March, T. Pun, and D. M. Squire, "Strategies for positive and negative relevance feedback in image retrieval," in *In Proceedings of the 15th International Conference on Pattern Recognition (ICPR 2000)*, IEEE, 2000, pp. 1043–1046.

[95] J. Huang, S. R. Kumar, and M. Mitra, "Combining supervised learning with color correlograms for content-based image retrieval," in *MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia*, New York, NY, USA, ACM, 1997, pp. 325–334.

[96] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in mars," in *In Proc. IEEE Int. Conf. on Image Proc*, 1997, pp. 815–818.

[97] D. Squire, W. Muller, H. Muller, and T. Pun, "Content-based query of image databases: Inspirations from text retrieval,"

[98] J. Fournier, M. Cord, and S. Philipp Foliguet, "Back-propagation algorithm for relevance feedback in image retrieval," 2001, pp. I: 686–689.

[99] S. Aksoy, R. M. Haralick, F. A. Cheikh, and M. Gabbouj, "A weighted distance approach to relevance feedback," in *International Conference on Pattern Recognition*, 2000, pp. 4812–4815.

[100] R. Brunelli and O. Mich, "Image retrieval by examples," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 164–171, 2000.

[101] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "Mindreader: Querying databases through multiple examples," in *VLDB '98: Proceedings of the 24rd International Conference on Very Large Data Bases*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1998, pp. 218–227.

[102] A. D. Doulamis and N. D. Doulamis, "A recursive optimal relevance feedback scheme for content based image retrieval," in *ICIP (2)*, 2001, pp. 741–744.

[103] R. Schettini, G. Ciocca, and I. Gagliardi, "Content-based color image retrieval with relevance feedback," in *ICIP (3)*, 1999, pp. 75–79.

[104] Y. Wu and A. Zhang, "A feature re-weighting approach for relevance feedback in image retrieval," in *ICIP (2)*, 2002, pp. 581–584.

[105] R. Picard, T. Minka, and M. Szummer, "Modeling user subjectivity in image libraries," 1996, pp. II: 777–780.

[106] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[107] K. robert Mller, S. Mika, G. Rtsch, K. Tsuda, and B. Schlkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, pp. 181–201, 2001.

[108] P. Hong, Q. Tian, and T. S. Huang, "Incorporate support vector machines to content-based image retrieval with relevant feedback," in *ICIP*, 2000.

[109] L. Zhang, F. Lin, and B. Zhang, "Support vector machine learning for image retrieval," in *ICIP (2)*, 2001, pp. 721–724.

[110] D. Tao and X. Tang, "Random sampling based svm for relevance feedback image retrieval," in *CVPR (2)*, 2004, pp. 647–652.

[111] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class svm for learning in image retrieval," in *ICIP (1)*, 2001, pp. 34–37.

[112] H. Xie and A. Ortega, "An user preference information based kernel for svm active learning in content-based image retrieval," in *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, New York, NY, USA, ACM, 2004, pp. 1–6.

[113] S. M. D., R. P. K., and V. R., *Introduction to Statistical Signal Processing with Applications*. Prentice Hall, 1996.

[114] Y. Wu, T. Huang, and K. Toyama, "Self-supervised learning for object recognition based on kernel discriminant-em algorithm," 2001, pp. I: 275–280.

[115] D. Tao and X. Tang, "Nonparametric discriminant analysis in relevance feedback for content-based image retrieval," in *ICPR (2)*, 2004, pp. 1013–1016.

[116] S. MacArthur, C. Brodley, and C. Shyu, "Relevance feedback decision trees in content-based image retrieval," in *CBAIVL '00: Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, Washington, DC, USA, IEEE Computer Society, 2000, p. 68.

[117] G. Guo, A. Jain, W. Ma, and H. Zhang, "Learning similarity measure for natural image retrieval with relevance feedback," 2001, pp. I:731–736.

[118] K. Tieu and P. A. Viola, "Boosting image retrieval," *International Journal of Computer Vision*, vol. 56, no. 1-2, pp. 17–36, 2004.

[119] F. Qian, B. Zhang, and F. Lin, "Constructive learning algorithm-based rbf network for relevance feedback in image retrieval," in *CIVR*, 2003, pp. 352–361.

[120] C. Nastar, M. Mitschke, and C. Meilhac, "Efficient query refinement for image retrieval," in *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, IEEE Computer Society, 1998, p. 547.

[121] C. Meilhac and C. Nastar, "Relevance feedback and category search in image databases," in *ICMCS '99: Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Washington, DC, USA, IEEE Computer Society, 1999, p. 9512.

[122] A. Dong and B. Bhanu, "A new semi-supervised em algorithm for image retrieval," 2003, pp. II: 662–667.

[123] M. Najjar, C. Ambroise, and J. P. Cocquerez, "Image retrieval using mixture models and em algorithm," in *SCIA*, 2003, pp. 1114–1121.

[124] J. Yoon and N. Jayant, "Relevance feedback for semantics based image retrieval," in *ICIP (1)*, 2001, pp. 42–45.

[125] F. Qian, M. Li, L. Zhang, H.-J. Zhang, and B. Zhang, "Gaussian mixture model for relevance feedback in image retrieval," in *Proc. IEEE ICME*, 2002, pp. 26–29.

[126] F. Jing, B. Zhang, F. Lin, W.-Y. Ma, and H.-J. Zhang, "A novel region-based image retrieval method using relevance feedback," in *MULTIMEDIA '01: Proceedings of the 2001 ACM workshops on Multimedia*, New York, NY, USA, ACM, 2001, pp. 28–31.

[127] F. Jing, M. Li, H. jiang Zhang, and B. Zhang, "Region-based relevance feedback in image retrieval," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2002, pp. 145–148.

[128] F. Jing, M. Li, H.-J. Zhang, and B. Zhang, "Support vector machines for region-based image retrieval," in *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, Washington, DC, USA, IEEE Computer Society, 2003, pp. 21–24.

[129] F. Jing, M. Li, H.-J. Zhang, and B. Zhang, "An effective region-based image retrieval framework," in *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, New York, NY, USA, ACM, 2002, pp. 456–465.

[130] "Probabilistic neural networks supporting multi-class relevance feedback in region-based image retrieval," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4*, Washington, DC, USA, IEEE Computer Society, 2002, p. 40138.

[131] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schlkopf, and B. S. Olkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*, MIT Press, 2003, pp. 321–328.

[132] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Manifold-ranking based image retrieval," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, New York, NY, USA, ACM, 2004, pp. 9–16.

[133] X. Jin, Y. Zhou, and B. Mobasher, "Web usage mining based on probabilistic latent semantic analysis," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, ACM, 2004, pp. 197–205.

[134] J. Fournier and M. Cord, "Long-term similarity learning in content-based image retrieval," in *In Proceedings of the International Conference on Image Processing*, IEEE Press, 2002, pp. 441–444.

[135] H. D. R., "Building a latent semantic index of an image database from patterns of relevance feedback," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4*, Washington, DC, USA, IEEE Computer Society, 2002, p. 40134.

[136] M. Koskela and J. Laaksonen, "Using long-term learning to improve efficiency of content-based image retrieval," in *PRIS*, 2003, pp. 72–79.

[137] X. He, O. King, W. Ma, M. Li, and H. Zhang, "Learning a semantic space from user's relevance feedback for image retrieval," *Circuit Systems and Video Technology*, vol. 13, pp. 39–48, January 2003.

[138] X. Chen, C. Zhang, S.-C. Chen, and M. Chen, "A latent semantic indexing based method for solving multiple instance learning problem in region-based image retrieval," in *ISM '05: Proceedings of the Seventh IEEE International Symposium on Multimedia*, Washington, DC, USA, IEEE Computer Society, 2005, pp. 37–45.

[139] M. Li, Z. Chen, L. Wenyin, and H.-J. Zhang, "A statistical correlation model for image retrieval," in *MULTIMEDIA '01: Proceedings of the 2001 ACM workshops on Multimedia*, New York, NY, USA, ACM, 2001, pp. 42–45.

[140] C.-H. Hoi and M. R. Lyu, "A novel log-based relevance feedback technique in content-based image retrieval," in *Proc of the ACM Intl Conf on Multimedia*, New York, NY, USA, ACM, 2004, pp. 24–31.

[141] T. Yoshizawa and H. Schweitzer, "Long-term learning of semantic grouping from relevance-feedback," in *Proc of the ACM SIGMM Intl workshop on Multimedia Information Retrieval*, New York, NY, USA, ACM, 2004, pp. 165–172.

[142] P. H. Gosselin and M. Cord, "Semantic kernel updating for content-based image retrieval," in *ISMSE '04: Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering*, Washington, DC, USA, IEEE Computer Society, 2004, pp. 537–544.

[143] P. Gosselin and M. Cord, "Semantic kernel learning for interactive image retrieval," 2005, pp. I: 1177–1180.

[144] J. Han, M. Li, H. Zhang, and L. Guo, "A memorization learning model for image retrieval," 2003, pp. III: 605–608.

[145] P.-Y. Yin, B. Bhanu, K.-C. Chang, and A. Dong, "Improving retrieval performance by long-term relevance information," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*, Washington, DC, USA, IEEE Computer Society, 2002, p. 30533.

[146] W. Jiang, G. Er, and Q. Dai, "Multi-layer semantic representation learning for image retrieval," 2004, pp. IV: 2215–2218.

[147] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "Self-organising maps as a relevance feedback technique in content-based image retrieval," *Pattern Anal. Appl.*, vol. 4, no. 2-3, pp. 140–152, 2001.

[148] C.-H. Chan and I. King, "Using biased support vector machine to improve retrieval result in image retrieval with self-organizing map," in *ICONIP*, 2004, pp. 714–719.

[149] I. Gondra and D. R. Heisterkamp, "Summarizing inter-query learning in content-based image retrieval via incremental semantic clustering," in *ITCC (2)*, 2004, pp. 18–22.

[150] I. Gondra, D. R. Heisterkamp, and J. Peng, "Improving the initial image retrieval set by inter-query learning with one-class svms," in *In Proceedings of the 3 rd International Conference on Intelligent Systems Design and Applications*, 2004, pp. 1–10.

[151] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *Information Processing and Management*, 1988, pp. 513–523.

[152] V. S. N. Prasad, A. G. Faheema, and S. Rakshit, "Feature selection in example-based image retrieval systems," in *Intl Conf on Computer Vision Graphics and Image Processing*, 2002.

[153] X. Zhou, Q. Zhang, L. Zhang, L. Liu, and B. Shi, "An image retrieval method based on collaborative filtering," *Intelligent Data Engineering and Automated Learning*, 2003.

[154] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of Conf. on Comp. Vision and Pattern Recog.*, 2006, pp. II: 2161–2168.

[155] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. of the International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1470–1477.

[156] D. Liu and T. Chen, "Content-free image retrieval using bayesian product rule," in *ICME*, 2006, pp. 89–92.

[157] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," Tech. Rep. UCB/ERL M00/41, EECS Department, University of California, Berkeley, 2000.

[158] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *18th National Conference on Artificial Intelligence*, 2002, pp. 187–192.

[159] K. Yu, A. Schwaighofer, V. Tresp, W.-Y. Ma, and H. Zhang, "Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical bayes," in *UAI*, 2003, pp. 616–623.

[160] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "An optimal and progressive algorithm for skyline queries," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, New York, NY, USA, ACM, 2003, pp. 467–478.

[161] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *Proceedings of the 17th International Conference on Data Engineering*, Washington, DC, USA, IEEE Computer Society, 2001, pp. 421–430.

[162] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: An online algorithm for skyline queries," in *VLDB*, 2002, pp. 275–286.

[163] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive skyline computation in database systems," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 41–82, 2005.

[164] M. Sharifzadeh and C. Shahabi, "The spatial skyline queries," in *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, VLDB Endowment, 2006, pp. 751–762.

[165] C. Brando, M. Goncalves, and V. González, "Evaluating top-k skyline queries over relational databases," Master's thesis, 2007.

[166] T. S. Huang, C. K. Dagli, S. Rajaram, E. Y. Chang, M. I. Mandel, G. E. Poliner, and D. P. W. Ellis, "Active Learning for Interactive Multimedia Retrieval," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 648–667, 2008.

[167] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Special Interest Group on Management Of Data Conf*, 1994, pp. 419–429.