

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**IMAGE AUTO-ANNOTATION BASED ON
COMBINATION OF TEXT AND VISUAL
CLUSTERING**

by
Erbuğ ÇELEBİ

February, 2006
İZMİR

IMAGE AUTO-ANNOTATION BASED ON COMBINATION OF TEXT AND VISUAL CLUSTERING

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfilment of the Requirements for the Degree of Doctor of
Philosophy in Computer Engineering, Computer Engineering Program.**

**by
Erbuğ ÇELEBİ**

**February, 2006
İZMİR**

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**IMAGE AUTO-ANNOTATION BASED ON COMBINATION OF TEXT AND VISUAL CLUSTERING**” completed by **Erbuğ ÇELEBİ** under supervision of **Asst. Prof. Dr. Adil ALPKOÇAK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

.....
Asst. Prof. Dr. Adil ALPKOÇAK

Supervisor

.....
Prof. Dr. Tatyana YAKNO

(Committee Member)

.....
Asst. Prof. Dr. Haldun SARNEL

(Committee Member)

.....
Prof. Dr. Alp KUT

(Jury Member)

.....
Assoc. Prof. Dr. M. Ertuğrul ÇELEBİ

(Jury Member)

.....
Prof.Dr. Cahit HELVACI

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Dr. Adil Alpkoçak for his patience, guidance and encouragement. I appreciate his vast knowledge and skill in many areas (i.e., vision, ethics, and work quality) and his assistance in reviewing and guiding me on my reports, papers and thesis. It was a privilege to work with him.

I thank the members of thesis committee for their comments and suggestions.

I appreciate to my friend Dr. Oktay Ünal who helped me on grammar correction for chapter three.

Finally, I would like to thank to my parents Kemal Çelebi and Gülen Çelebi. Without their support, encouragement and love completing this thesis will be much more difficult and long.

Erbuğ ÇELEBİ

IMAGE AUTO-ANNOTATION BASED ON COMBINATION OF TEXT AND VISUAL CLUSTERING

ABSTRACT

The emergence of multimedia technology and the rapidly expanding multimedia collections on the Internet have attracted significant research efforts in providing tools for effective retrieval and management of multimedia data. Traditional image retrieval systems were based on manual annotations of images. This is not powerful enough for proper image retrieval, because of manual annotations. Researchers are focused on extracting image features and annotating images automatically by considering their content that are color, texture and shape. Generally, user needs are high level features and retrieval systems consider/process the low-level features to accomplish the retrieval task. This difference between human interpretation and extracted/processed information is known as the semantic gap of such systems. In this thesis, our aim is to find a linkage between low-level features and high level features to bridge the semantic gap.

In this thesis, we propose a novel strategy at an abstract level by combining textual and visual clustering results to retrieve images using semantic keywords and auto-annotate images based on similarity with existing keywords. Our main hypothesis is that images that fall in to the same text-cluster can be described with common visual features of those images. In order to implement this hypothesis, we set out to estimate the common visual features in the textually clustered images. When an un-annotated image is given we find the best image match in the different textual clusters by processing their low-level features. Experiments have demonstrated that good accuracy of proposal and its high potential of use in annotation of images and for improvement of content based image retrieval.

Keywords: Content Based Image Retrieval, Semantic Information Retrieval, Semantic Gap, Automatic Image Annotation.

RESİMLERİN METİN VE GÖRSEL KÜMELEMESİNE DAYALI OLARAK OTOMATİK ETİKETLENMESİ

ÖZ

Gelişen çoklu ortam teknolojileri ve İnternet üzerinde hızlı bir şekilde çoğalan çoklu ortam ürünleri, araştırmacıları çoklu ortam verilerini sorgulamak ve yönetmek üzere yeni araçlar üretme yönündeki araştırmalara yönelmelerine sebep olmuştur. Geleneksel resim sorgulama sistemleri, elle etiketlenen resimler üzerinden yapılmaktadır. Fakat bu yöntem resim sorgulaması için yeterince güçlü bir yöntem değildir. Daha etkin çözümler bulmak için araştırmacılar, resim içindeki renk, desen ve şekil özelliklerini kullanarak resimleri otomatik etiketlendirme yönünde odaklanmışlardır. Genel olarak, kullanıcı istekleri yüksek seviyeli özellikler üzerinde olup, buna karşın, sorgulama sistemleri düşük seviyeli özellikler üzerinden bilgi üretmektedirler. Kullanıcı istekleri ve sistemlerin elde ettikleri bilgiler arasındaki fark “anlamsal eksiklik” olarak bilinir. Bu tezdeki temel amacımız düşük seviyeli özellikler ile yüksek seviyeli özellikler arasında bir bağlantı kurarak bu boşluğu doldurmaktır.

Bu tezde, metin ve görsel kümelemeleri birleştirerek, resimlerin otomatik etiketlenmesi ve sorgulanması için yeni bir yöntem önerilmiştir. Temel hipotezimiz, aynı metin kümesinde olan resimlerin ortak görsel özellikleri ile ifade edilebileceğidir. Hipotezi gerçeklemek için, metin kümelerindeki ortak görsel öğeleri belirleyecek bir yapı oluşturduk. Etiketlenmemiş bir resim sorgulandığı zaman, metin sınıflarındaki resimlerin düşük seviyeli özelliklerini işleyerek benzer resimleri bulduk ve verilen resmin otomatik etiketlenmesini sağladık. Sistemin doğruluğu ve resimleri otomatik etiketleyen bir sistem olarak kullanılabileceği deneyimlerle gösterilmiştir.

Anahtar sözcükler: İçeriğe dayalı resim sorgulama, anlamsal bilgi sorgulama, anlamsal eksiklik, otomatik resim etiketleme.

CONTENTS	Page
THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT.....	iv
ÖZ	v
CHAPTER ONE – INTRODUCTION	1
1.1 General.....	1
1.2 Image Databases	3
1.2.1 Low-level features	3
1.2.2 Forming Query for Image Retrieval	5
1.3 Bridging the Semantic Gap.....	6
1.4 Objective	9
1.4.1 Purpose of Research.....	9
1.4.2 Contribution of Research	9
1.5 Thesis Organization	10
CHAPTER TWO – LITERATURE SURVEY ON CONTENT BASED IMAGE RETRIEVAL.....	11
2.1 Introduction.....	11
2.2 Latent Semantic Indexing	11
2.2.1 Previous Works on Semantic Indexing.....	13
2.3 Applications of Content Based Image Retrieval.....	15
2.3.1 QBIC	15
2.3.2 Virage.....	15
2.3.3 Excalibur	16
2.3.4 VisualSeek	16
2.3.5 WebSeek	17
2.3.6 Shoebox.....	17

2.4	Content Based Semantic Image Retrieval.....	18
2.4.1	Blobworld	18
2.4.2	SIMPLIcity	19
2.4.3	Co-occurrence model	20
2.4.4	SemQuery	20
2.4.5	Cross Media Relevance Model	21
2.4.6	Translation Model	21
2.4.7	Multiple Bernoulli Relevance Model.....	22
2.4.8	Mix-Hier	22

CHAPTER THREE – IMAGE AUTO-ANNOTATION BASED ON COMBINATION OF TEXT AND VISUAL CLUSTERING..... 24

3.1	Introduction.....	24
3.2	Combination of Textual and Visual Clustering	24
3.2.1	Cover Coefficient based Clustering (C3M)	28
3.2.1.1	C3M Algorithm.....	28
3.2.1.2	C Matrix.....	29
3.2.1.3	C' Matrix.....	33
3.2.1.4	Number of Cluster Hypothesis.....	34
3.2.1.5	Cluster Seed Selection	34
3.2.1.6	The C3M Algorithm	35
3.2.2	Training.....	35
3.2.2.1	Multiple Low-Level Feature as Image Descriptor.....	38
3.2.3	Auto-Annotation and Image Retrieval.....	40
3.2.3.1	Constructing Query Vector	40
3.2.3.2	Query Processing	41
3.3	Numerical Example for Proposed Method	43
3.4	Maintenance of Training Set.....	48
3.5	Conclusion	49

CHAPTER FOUR – EXPERIMENTS	50
4.1 Introduction.....	50
4.2 Dataset.....	50
4.2.1 Descriptions of Dataset-files.....	52
4.3 Method for Evaluating Performance of Retrieval Effectiveness	54
4.4 Experimental results.....	57
4.4.1 Traditional Image Retrieval and annotation	58
4.4.2 Blob Based Image Retrieval and Auto-annotation.	61
4.4.3 Image Retrieval and Auto-annotation by Translating Text Space to Image Space	63
4.5 Evaluating results.....	66
 CHAPTER FIVE – CONCLUSION	 70
5.1 Conclusion	70
5.2 Future Directions	72
 REFERENCES	 74
 APPENDIX A.....	 79
A.1 Traditional Image Retrieval and annotation.....	79
A.2 Blob Based Image Retrieval and Auto-annotation	87
A.3 Image Retrieval and Auto-annotation by Combining High-level and Low Level Features.....	89
APPENDIX B	93
B.1 Two level Thesaurus List for Corel Data Set.....	93
APPENDIX C	98
APPENDIX D.....	110
D.1 Singular Value Decomposition	110

CHAPTER ONE

INTRODUCTION

1.1 General

The emergence of multimedia technology and the rapidly expanding text, image and video collections on the Internet have attracted significant research efforts in providing tools for effective retrieval and management of visual data. Image databases are becoming increasingly popular due to advances in mobile phones, digital cameras and various applications where they produce large amount of images. Advances in computation power, storage devices, scanning, networking, image compression, and desktop publishing makes image databases a part of our digital life. The application areas of such systems may be on personal photo album, medical image database, clip art images, textile archive, photojournalism, world-wide-web, museum archives and etc.

Multimedia databases have become more important since the simplicity of creating multimedia information (such as text, audio, image and video) is cheaper, faster and easier than few years ago. Multimedia records are collections of multimedia documents. Each document in the collection can contain multiple media and be a mixture of text, images, video and audio. However, the term multimedia is sometimes used to refer to a single medium, provided that it is not text. We adopt this convention and use the term multimedia document to refer to any document containing at least an image, a piece of video material, or an audio fragment.

For each kind of multimedia collection there is a need for ad-hoc retrieval system. In other words, retrieval systems are designed separately for each kind of multimedia collection. However, we cannot access to or make use of the information unless it is organized to allow efficient browsing, searching and retrieval. Image retrieval has been a very active research area since the 1970's.

Often, the phrase “content based retrieval” is used to denote the retrieval, browsing and searching tasks of multimedia objects when their low-level features are considered. There are four major difficulties in content-based retrieval of multimedia data. *First*, the content of multimedia data needs a powerful set of search facilities for keywords, sounds, color, texture, spatial information, motion and features. An instance for a feature in video may be the goal scenes in a football game. *Second*, if a method or processing technique is designed feature extraction of one type of data feature, it's usually not appropriate for others. For instance, a technique designed for indexing audio archives may not be usable for image archives; or, a technique developed for extracting texture feature may not be useful for extracting shape feature in image and video data. *Third*, the usual huge size of multimedia data requires an exhaustive search. A similarity search is desirable for a multimedia database to find the similar objects to query objects; since there is not any exact matching method exists. For example, if a picture of a car submitted as a query to an image database, we expect to retrieve pictures that contain similar cars in them. Shape, color and texture information are used as individually and sometimes the combination of those features are considered for the comparison. There exists a different distance metric for each kind of information in the images. Evaluated distances are ranked and most similar images are retrieved as similar objects. Depending on the similarity metric used; sometimes results may be ranked as ascending or descending order. As the *fourth*, there is always a gap between low-level and high level definitions of multimedia objects. That means, users always issue query by considering high-level features (i.e. keywords), but the processing is performed on low-level features (color, texture, shape and etc.). For example a user who is trying to retrieve images that contains car submits the keyword “car” as the query string that it is high level definition. However, retrieval system considers and processes the low-level features of images in the archive to accomplish the retrieval task. This difference between human interpretation and extracted/processed information is known as the semantic gap of such systems. In this thesis we try to fill this gap by linking low-level and high level features of images in order to make more meaningful retrieval results.

In the following; section 1.2 describes the traditional text based image retrieval systems and their limitations. Low-level features and query forming fashions are also discussed in section 1.2. The details about the semantic gap in content-based retrieval systems are explained in section 1.3. The objective of the thesis is given in section 1.4, and the chapter is concluded with last section that describes the thesis organization.

1.2 Image Databases

Image databases differ from traditional databases and they are a subset of multimedia databases. Conventional information retrieval systems were based solely on text. What most system developers do is, they simply give keyword(s) to each image, and allow the users to make query on these keywords for accessing the images or multimedia object. Associating each image with keywords is not useful with several reasons. First of all, it is very time consuming and labor intensive task. Additionally, there may be given different annotations even for the same image because of the annotation operators. Instead of giving keywords for each image and making query on these keywords, researchers focused on low-level features of the images. Low-level features of images are extracted automatically and processed by the image databases to obtain information about the image content. Generally these features are based on colors, shapes and textures of images where they are explained in the following sub-section.

1.2.1 Low-level features

The base of content-based image retrieval is visual feature extraction. Color, texture, shape and spatial relationships are among the widely used features. Because of perceptual subjectivity and the complex composition of visual data there does not exist a single best representation for any given visual feature. Multiple approaches have been introduced for each of these visual features and each of them

characterizes the feature from a different perspective. It is a separate research topic for each kind feature extraction methodology for images.

Color is one of the most widely used visual features in content-based image retrieval. It is relatively robust and simple to represent. Various studies of color perception and color spaces have been proposed (Monay, et. al., 2003 and Smith, et. al., 1996). The color histogram is the most commonly used representation technique, statistically describing the combined probabilistic property of the three-color channels. RGB, Lab and HSV color models and their histograms are used widely to describe the images.

Texture is an attribute representing the spatial arrangement of gray levels of pixels in a region. Smith (Smith, & Chang, 1996) said that texture refers to visual pattern that has properties of homogeneity that do not result from presence of only a single color or intensity. It is a powerful discriminating feature, present almost everywhere in nature. However, it is almost impossible to describe texture in words, because it is virtually a statistical and structural property. There are three major categories of texture-based techniques (Celebi, & Alpkocak, 2000) namely, *statistical*, *spectral*, and *structural* approaches.

Shape representation is normally required to be invariant to *translation*, *rotation*, and *scaling*. In general, shape representations can be categorized into either *boundary-based* or *region-based*. The former uses only the outer boundary characteristics of the entities while the latter uses the entire region (Rui, Huang, & Chang, 1998). Well known methods include Fourier descriptors and moment invariants.

In general these visual features are not extracted from the whole image. The widely used approach is; first segment the image into meaningful regions and then extract features from those segmented regions as in study of Smith et. al. (Smith, & Chang, 1996). Also, segmenting an image in to meaningful regions is also a separate research topic, as it is for feature extraction process. Deb & Zhang (Deb, & Zhang,

2004) claims that despite large number of indexing and retrieval techniques have been developed, there is still no universally accepted feature extraction, indexing and retrieval technique available.

Image retrieval systems that process the low-level features of images are called content-based image retrieval (CBIR) systems in the literature. "Content-based" means that the search makes use of the content of the images themselves, rather than relying on manually associated meta-data that called annotations, captions or keywords.

Few authors claim that (Mojsilovic, & Jose, 1999) color, texture and shape features do not adequately model image semantics when we consider broad image domains. Existing CBIR systems depend on visual attributes as mentioned above to classify and search for similar images. This approach provides excellent results when constrained to a single application domain, however no matter how sophisticated they are, color, texture and shape features are alone do not adequately model image semantics and thus have many limitations when applied to broad content image databases.

Example scenarios to image retrieval systems may be; a designer needs images of fabrics with particular texture and color, an artist looks for sunshine over the sea, and movie producer needs a video clip of a man running from left to right on the football field. Other application areas can be remote sensing, geographic information system, trade and copyright database management, image mosaics, e-commerce, WWW image search engines, and picture archiving.

1.2.2 Forming Query for Image Retrieval

In a typical usage of content based image retrieval system, user should be able to search an image database for images that express the desired information or (s)he may process an image and (s)he is interested in and wants to find images from the

database that are similar to the query image (or example image). Different implementations of image retrieval make use of different types of user queries.

- *Query by example*, the user searches with a query image (supplied by the user or chosen from a random set), and the software finds images similar to it based on various low-level criteria.
- *Query by keyword*, the user submits a keyword and software locates images that are related with that keyword. Traditional systems use this approach and retrieve the results by exact match with annotations. For content based image retrieval systems, query operation does not performed on manual annotations instead system makes search on annotations that are estimated automatically (auto-annotation).
- *Query by sketch*, user draws a rough approximation of the image they need and for example with colored regions, and the system locates images whose layout matches the sketch.
- Other methods include specifying the proportions of colors desired (e.g. "80% red, 20% blue") and searching for images that contain an object given in a query image.

1.3 Bridging the Semantic Gap

Querying on text documents such as Google does is somehow simple; because there exist an alphabet to express user desires. In addition to expressing the query in a simple way, it is possible to make exact match on keywords without any pre-processing on keywords and their relations. However, there is no such an alphabet for image archives. Further more there is a need to pre-process the multimedia objects to make them available for querying. Also, exact match is not possible for multimedia objects.

One of the challenges for multimedia information retrieval is to make user to form their query in a simple and effective way. In content-based image retrieval, query-by-example has been used as a method to search image databases. However, in many real world applications, it is hard to find an example to describe the user's information need. Because it is more intuitive to use keyword (text) to describe information need, most of current commercial image search engines are keyword based. Keyword based CBIR systems automatically annotate images using text (or pre-defined vocabulary) so that users use text to search multimedia. On the other hand, text-based traditional image retrieval systems are not sufficient for retrieving visual data because they depend on file tags, keywords, or annotations with the images. They do not allow queries based directly on the visual properties of the images. They depend on the particular vocabulary used and they do not provide queries for images "similar" to given image. In traditional text based databases, retrieval is based on an exact match of the attribute values so they do not have the ability to rank-order results by the degree of similarity with the query image. Unfortunately, it is impossible to represent the content of an image with a few words. For example, an image annotated as containing "woman" and "children" cannot be retrieved by a query searching for the keyword "people".

We do not suggest, "Do not to use keywords or text, and just to use image features instead." Instead, both keywords and image features can be used for image properties. Retrieval performance can be increased, by using keywords together with image features.

It has been widely recognized that the family of image retrieval techniques should become an integration of both *low-level* visual features addressing the more detailed perceptual aspects and *high-level* semantic features underlying the more general conceptual aspects of visual data. Neither of these two types of features is sufficient to retrieve or manage visual data in an effective or efficient way (Smeulders, Worring, & Santini, 2000). Although efforts have been devoted to combining these two aspects of visual data, the semantic gap between them is still a huge barrier in

front of researchers. Figure 1.1 shows the layers between low-level and high level features that we need to overcome to bridge the gap.

Intuitive and heuristic approaches do not provide us with satisfactory performance. Therefore, there is an urgent need of finding the latent correlation between low-level features and high-level concepts and merging them from a different perspective. How to find this new perspective and bridge the gap between visual features and semantic features has been a major challenge in this research field.

Methods used for textual information retrieval have been transplanted into image retrieval in a variety of ways, at the very beginning of CBIR studies. Some researches (Feng, Manmatha, & Lavrenk, 2004 and Duygulu, P., 2003) used machine-learning approaches for content based image retrieval. Because the semantic gap problem is not likely to be solved in a near term, researchers have been trying to develop new solutions for this issue as in this study.

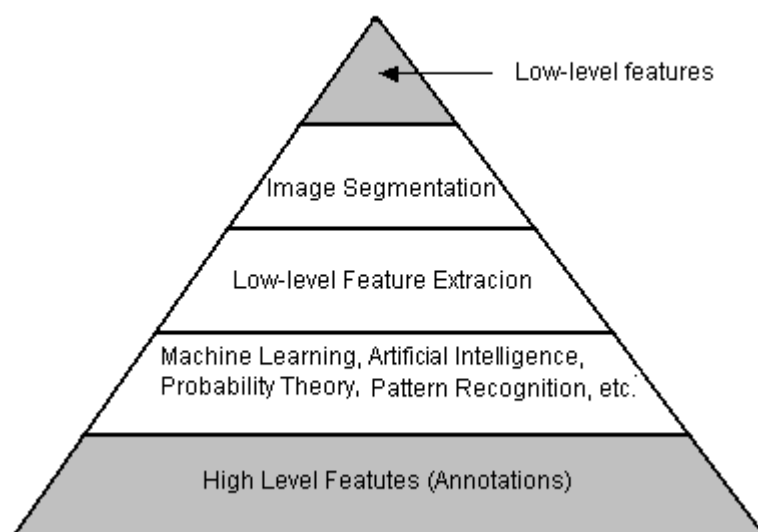


Figure 1.1 Layers (open problems) between high level and low level features that show the semantic gap.

1.4 Objective

1.4.1 Purpose of Research

The main difficulty of Content Based Image Retrieval Systems is to describe or to understand images. It is obvious that, human beings are much better than computers at extracting and making use of semantic information from images. Image understanding should start from explaining the meaning of image objects and their relationships in automatic way. Unfortunately, this goal is still beyond the reach of state-of-the-art in computer vision.

In this thesis, we are trying to assign keywords to images by considering their content, in an automatic way. Annotating images will help users making queries with keywords on image archives. In addition to *query by keyword*, content-based image retrieval systems try to answer *query by example* where users submits a sample image to retrieve semantically similar ones. Mostly the answers of those two problems are the same.

The goal of this research is to find a methodology that finds the correlation between low-level visual features and high-level semantics of images thus fills the semantic gap. To be more specific, the significance of this approach is to design and implement an effective and efficient framework for image retrieval by using the relation between visual features and annotations. *C3M*, which is a two level probabilistic clustering technique, is used for content-based image retrieval. By using this technique, the aim is to find most common visual features in annotation clusters and retrieve the images by comparing the query image with those common features.

1.4.2 Contribution of Research

The main contribution of this research is to propose a new strategy (1) to retrieve images using semantic keywords and (2) auto-annotate images based on similarity

with existing keywords for bridging the gap between low-level visual features and lack of semantic knowledge in multimedia information retrieval. Our solution works on an abstract level and combines both textual and visual clustering algorithms performance. The main idea behind this strategy is that the images within the same text cluster should also have common visual features and could be stored in the same visual cluster.

1.5 Thesis Organization

In this chapter we have given information about multimedia information retrieval systems, their challenges, query models and low-level features. Also, in addition to this information, we have stated what we are trying to accomplish, what is our goal and what is our contribution to the field. In chapter two, an overview on content-based image retrieval is given, to show the aspects of variety number of methods and to explore similar studies in the literature. In chapter three, we introduce the proposed method to content based image retrieval and an example is presented to make the topic clearer. In chapter four, several experiments are performed on the dataset to show that our proposed method is powerful enough for CBIR. A comparison of similar studies in the literature is also provided in chapter four. The last chapter concludes the thesis by providing the results we obtained and offers a look at potential future works on this topic.

CHAPTER TWO

LITERATURE SURVEY ON CONTENT BASED IMAGE RETRIEVAL

2.1 Introduction

The task of content based image retrieval system is to identify relevant image to query image and retrieve the most similar images. As we discussed in chapter one; text based image archives have advantage of retrieving the exact keyword matches. However, the absence of vocabulary for multimedia object makes exact matching not possible. In other words, multimedia information retrieval systems cannot access information and document content directly, but have to rely on descriptors of them.

In essence, the main functionality of CBIR systems is about representing documents and queries, and about comparing the descriptors to determine if a relevance relation exists. Hence, an image retrieval model needs to specify a query representation method, an image description method and a function to compute the retrieval status value based on the query representation and image descriptors.

Following subsections describe the most common models on content based retrieval systems. Early studies of CBIR start with using semantic text retrieval techniques that are the applications of latent semantic indexing and cross language retrieval. So, brief review on latent semantic indexing is given, in section 2.1. We have explained application of CBIR in Section 2.2. In the last few years, researchers are focused on filling the semantic gap for CBIR as in our study where these similar studies are discussed in section 2.3.

2.2 Latent Semantic Indexing

In today's search engines, most approaches used to retrieving data from document databases depend on keyword match between user's input and those assign to

documents. However, sometimes these methods are incomplete and imprecise because of the description styles used in the documents. By saying description styles we mean; one word may be used for different meanings or same object may be described by different keywords from different authors in the documents. As a more advanced search method to traditional ones LSI (Latent Semantic Indexing) is used to overcome these issues. LSI tries to find solution to polysemy and synonym problems.

Latent Semantic Indexing is an information retrieval method, which attempts to capture the hidden structure we mentioned above by using techniques from linear algebra. LSI is completely automatic and does not need any user interaction to extract hidden semantics such as polysemy and synonyms. LSI is based on producing a reduced-rank approximation of documents by keyword matrix and users query vector. To produce a reduced-rank approximation of $m \times n$ term by document matrix A , one must first be able to identify the dependence between the columns or rows of the matrix. The most popular methods used for that purpose are SVD (Singular value decomposition) and QR Factorisation. Principal component analysis which reduces the redundancy contained in the data also creates low-dimensioned feature matrix, however it is not popular method for LSI.

In Latent Semantic Space, query vector and document vector can have small distance (i.e. Cosine distance) even if they do not share any terms as long as their terms are semantically similar. LSI can be considered as a similarity metric alternative to keyword match technique used in text retrieval.

The latent semantic space has fewer dimensions than the original space (which has as many dimensions as terms). LSI is thus a method for dimensionality reduction. A dimensionality reduction technique takes a set of objects that exist in a high-dimensional space and represents them in a low-dimensional space, often in a two-dimensional or three-dimensional space for the purpose of visualization. On the LSI case, matrixes are reduced to 200-300 but in general it is experimental.

As we mentioned above, latent semantic indexing is the application of a particular mathematical technique, called Singular Value Decomposition or SVD. SVD (and hence LSI) is a least-squares method. The projection into the latent semantic space is chosen such that the representations in the original space are changed as little as possible when measured by the sum of the squares of the differences. The description of SVD is given in Appendix E.

2.2.1 Previous Works on Semantic Indexing

The application of Singular Value Decomposition in information retrieval was originally proposed by a group of researchers at Bellcore (Deerwester, et. al., 1990) and called Latent Semantic Indexing in this context. Regarding the performances, (Berry et al., 1995) reports that for several information science test collections, the average precision using LSI ranged from comparable to 30% better than that obtained using standard keyword vector methods. The LSI method performs best relative to standard vector methods when the queries and relevant documents do not share many words, and at high levels of recall.

Some of multimedia retrieval applications use relevance feedback where it is an approach to take an interactive help from users to define relevant and non-relevant documents. In relevance feedback applications attribute (keyword, image and etc.) weights are increased and decreased depending on the user's feedback. Most of the tests of Relevance Feedback using LSI have involved a method in which the initial query is replaced with the vector sum of the documents the users has selected as relevant. The use of negative information has not yet been exploited in LSI; for example, by moving the query away from documents which the user has indicated are irrelevant. Replacing the users' query with the first relevant document improves performance by an average of 33% and replacing it with the average of the first three relevant documents improves performance by an average of 67%. Relevance feedback provides sizable and consistent retrieval advantages. One way of thinking about the success of these methods is that many words (those from relevant documents) augment the initial query that is usually quite impoverished. LSI does

some of this kind of query expansion or enhancement even without relevance information, but can be augmented with relevance information (Berry, et al., 1995).

It is important to note that the LSI analysis does not base on English syntax or semantics. This means that LSI is applicable to any language. In addition, it can be used for cross-language retrieval - documents are in several languages and user queries can match documents in any language. What is required for cross-language applications is a common space in which words from many languages are represented for the same content.

Landauer et. al (Landauer, & Littman, 1990) described a method for creating such an LSI space. The original term-document matrix is formed using a collection of abstracts that have versions in more than one language. Each abstract is treated as the combination of its say, French English versions. The truncated SVD is computed for this term by combined-abstract matrix A. The resulting space consists of combined-language abstracts, English words and French words. English words and French words that occur in similar combined abstracts will be near each other in the reduced-dimension LSI space. After this analysis, monolingual abstracts can be folded-in: a French abstract will simply be located at the vector sum of its constituent words that are already in the LSI space. Queries in either French or English can be matched to French or English abstracts. There is no difficult translation involved in retrieval from the multilingual LSI space. Experiments showed that the completely automatic multilingual space was more effective than single-language spaces. The retrieval of French documents in response to English queries (and vice versa) was as effective as first translating the queries into French and searching a French-only database. The method has shown almost as good results for retrieving English abstracts and Japanese Kanji ideographs, and for multilingual translations (English and Greek) of the Bible (Young, 1994)

2.3 Applications of Content Based Image Retrieval

There are varieties (more than 50) of multimedia retrieval systems for text, image and video archives. Commercial applications are available for specific corpuses in the market. In this section some of commercial applications of MIR will be illustrated.

2.3.1 *QBIC*

IBM's QBIC (Query by Image Content) system (Niblack, et. al., 1993) is probably the best known of all image content retrieval systems. It is available commercially either in standalone form, or as part of other IBM products such as the DB2 Digital Library. 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, specifying an example query image, or sketching a desired shape on the screen. The system extracts and stores colour, shape and texture features from each image added to the database. At search time, the system matches appropriate features from query and stored images, calculates a similarity score between the query and each stored image examined, and displays the most similar images on the screen as thumbnails. The latest version of the system incorporates more efficient indexing techniques, an improved user interface, the ability to search grey-level images, and a video storyboarding facility (Niblack, et. al., 1998). An online demonstration, together with information on how to download an evaluation copy of the software, is available on the World-Wide Web at <http://wwwqbic.almaden.ibm.com/>.

2.3.2 *Virage*.

Another well-known commercial system is the VIR Image Engine from Virage, Inc (Bach, et. al., 1996). This is available as a series of independent modules, which

systems developers can build in to their own programs. This makes it easy to extend the system by building in new types of query interface, or additional customized modules to process specialized collections of images such as trademarks. Alternatively, the system is available as an add-on to existing database management systems such as Oracle or Informix. An on-line demonstration of the VIR Image Engine can be found at <http://www.virage.com/online/>. A high-profile application of Virage technology is AltaVista's *AV Photo Finder* (<http://image.altavista.com/cgi-bin/avncgi>), allowing Web surfers to search for images by content similarity. Virage technology has also been extended to the management of video data details of their commercial Videologger product can be found on the Web at <http://www.virage.com/market/cataloger.html>.

2.3.3 *Excalibur*

A similar philosophy has been adopted by Excalibur Technologies, a company with a long history of successful database applications, for their Visual RetrievalWare product (Feder, 1996). This product offers a variety of image indexing and matching techniques based on the company's own proprietary pattern recognition technology. It is marketed principally as an applications development tool rather than as a standalone retrieval package. Its best-known application is probably the Yahoo! Image Surfer, allowing content-based retrieval of images from the World-wide Web. Further information on Visual RetrievalWare can be found at <http://www.excalib.com/>, and a demonstration of the Yahoo Image Surfer at <http://isurf.yahoo.com/>. Excalibur's product range also includes the video data management system Screening Room.

2.3.4 *VisualSeek*

In the first step of VisualSeek (Smith, et. al., 1996), each image is automatically decomposed into regions of equally dominant colours. For each region, feature properties and spatial properties are retained for the subsequent queries. A query

consists of finding the images that contain the most similar arrangements of similar regions. The color region extraction uses the back-projection technique. To start a query, the user sketches a number of regions, positions and dimensions them on the grid and selects a color for each region. Also, the user can indicate boundaries for location and size and/or spatial relationships between regions. After the system returns the thumbnail images of the best matches, the user is allowed to search by example using the returned images. VisualSeek system is available on the Internet at: <http://www.ctr.columbia.edu/~jrsmith/html/pubs/acmmm96/acm.html>

2.3.5 *WebSeek*

WebSeek (Smith, 1997) collects its content by a collection processes through Web robots, though it has the advantage of video search and collection as well. It was developed at Columbia University. WebSeek makes text-based and color based queries through a catalogue of images and videos. Color is represented by means of a normalized 166-bin histogram in the HSV color space. For the query, user initiates a query by choosing a subject from the available catalogue or entering a topic. The results of the query may be used for a color query in the whole catalogue or for sorting the result list by decreasing color similarity to the selected item. Also, the user has the possibility of manually modifying an image/video color histogram before reiterating the search. Visual Seek system is available on the Internet at <http://persia.ee.columbia.edu:8008/>.

2.3.6 *Shoebox*

In ShoeBox (Mills, et. al., 2000) system, the images are annotated by keywords using speech recognition. The automatic features extracted are the average color in HSV color space, and the variances in each of the color channels of regions. The regions are either resulting from color segmentation, or a fixed portioning of the image, or the overlay of both. Querying can be done with the spoken annotations.

Another way is to select a region from an image in the database to find similar images.

2.4 Content Based Semantic Image Retrieval

Content based semantic image retrieval techniques differs from the applications described in section 2.3. When we say “semantic image retrieval” we mean that the relation between low level and high level features are considered. In the following, we have explained few of those methods where their aims are similar to our study. In chapter four, we have compared the performance results of few similar researches with the proposed method, where they are Co-occurrence, Translation Model, CMRM, MBRM and Mix-Hier. The details of those similar are also discussed in this section.

2.4.1 *Blobworld*

Blobworld (Carson, et. al., 1999) is a CBIR system developed at University of California, Berkeley. The system automatically extracts the regions of an image, which roughly correspond to object or parts of objects. It allows users to query for images based on the objects they contain. The user first selects a category, which already limits the search space. In an initial image, the user selects a region (blob), and indicates the importance of the blob. Next, the user indicates the importance of the blob’s color, texture, location, and shape. More than one region can be used for querying.

Their approach is useful in finding specific objects and not, as they put it, “stuff” as most systems which concentrate only on “low level” features with little regard for the spatial organization of those features. It allows for both textual and content-based searching. This system is also useful in its feedback to the user, in that it shows the internal representation of the submitted image and the query results. Thus, unlike some of the other systems, which allow for color histogram similarity metrics, which

can be adjusted, this can help the user understand why they are getting certain results.

Example of query that finds the similar images of a flower can found in Figure 2.1 and other examples can be reached from <http://elib.cs.berkeley.edu/blobworld/>

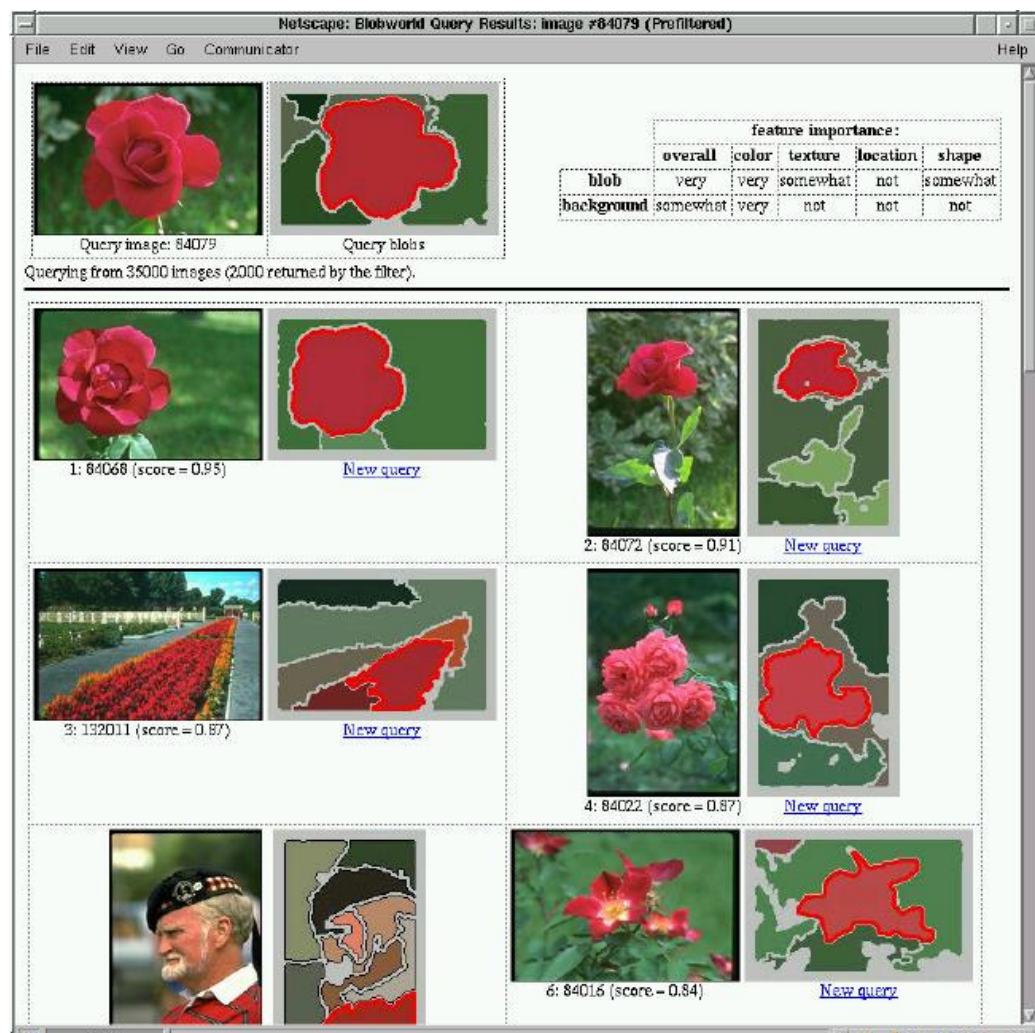


Figure 2.1: Blobworld (Carson, et. al., 1999), example query results.

2.4.2 SIMPLIcity

James Z. Wang et al. (2001) present an image retrieval system, SIMPLIcity (Semantics sensitive Integrated Matching for Picture Libraries), an image retrieval

system, which uses a wavelet-based approach for feature extraction, semantics classification methods, and integrated region matching based upon image segmentation. Their system classifies images into semantic categories such as textured-nontextured, graph photograph. Potentially, the categorization enhances retrieval by permitting semantically adaptive searching methods and narrowing down the searching range in a database. A measure for the overall similarity between images is developed using a region-matching scheme that integrates properties of all the regions in the images. For the purpose of searching images, they have developed a series of statistical image classification methods.

2.4.3 *Co-occurrence model*

Sometimes, extracting semantics of images can be considered as auto-annotation of images with keywords. One approach to automatically annotate images is to look at the probability of associating words with image regions. Mori et. al. (1999) used a co-occurrence model, which they look at the co-occurrence of words with image regions (sub images) created using regular grid. Each sub image is described with the color histogram and 45° rotated Sobel filters for measuring gradient intensity. The sub images are reduced to prototypes by using vector quantization. As each block inherits all words of the annotation of the complete image, the prototypes have accumulated all words of the clustered blocks they stand for. This aggregation of words is used to calculate the most probable words for a prototype. Experiments indicate an increase over random word assignment of 9%.

2.4.4 *SemQuery*

The integration of the retrieval on heterogeneous features is not a trivial task since the features in texture, shape and color are generated using different computation methods. SemQuery (Sheikholeslami, et. al, 2002) makes semantic based clustering to support visual queries on heterogeneous features of images. In that study database images are clustered based on their heterogeneous features that they assume that they

will be semantic clusters. Each semantic image cluster contains a set of sub clusters that are represented by the heterogeneous features that the images contain. An image is included into a feature sub cluster only if the image contains all the features under the same cluster. In their experiments they used 29,400 color and texture feature vectors of images and they divided the entire set into five categories of cloud, floral, leaves, mountain and water. Their average recall on the semantic clustered database is 0.45.

2.4.5 Cross Media Relevance Model

In their study Jeon et. al. (Jeon et. al., 2003) assume that image regions could be described via small vocabulary of blobs. Given a training set of images with annotations, they claim that probabilistic models allow predicting the probability of generating a word given the blobs in an image as in our study. This may be used to automatically annotate and retrieve images given a word as a query. Using a training set of annotated images, they learn the joint distribution of blobs and words, which they call a cross-media relevance model (CMRM) for images. They used Corel database to evaluate their results as in our study. They have evaluated their system same as our evaluation. Their system's mean precision is 0.10 and mean recall is 0.09, where the comparison is presented in chapter four.

2.4.6 Translation Model

Few other researches (Duygulu, et. al., 2002) have also examined the problem by using machine learning approaches. In particular Duygulu, et. al. proposed to describe images using a vocabulary of blobs as in other similar studies. Each image is generated by using a certain number of blobs. Their Translation Model (a substantial improvement on the Co-occurrence Model) assumes that image annotation can be viewed as the task of translating from a vocabulary of blobs to a vocabulary of words. Given a set of annotated training images, they show how one can use one of the classical machine translation models suggested by Brown et al. (Brown et. al,

1993) to annotate a test set of images. The method proposed by Brown et. al. is developed for translating English sentences to French sentences by using Expectation maximization algorithm.

2.4.7 Multiple Bernoulli Relevance Model

In study, Multiple Bernoulli relevance models (MBRM) for image and video annotation system (Feng et. al. 2004) they show how multiple Bernoulli relevance model can be used for both auto-annotation and retrieval from images and videos. They choose to partition each image into a set of rectangular regions and to compute a feature vector over these regions. The relevance model is a joint probability of distribution of the word annotations and the image feature vectors. The word probabilities are estimated using a multiple Bernoulli model and the image feature probabilities using a non-parametric kernel density estimate. They have used Corel dataset and a set of video key frames from NIST's Video Trec for evaluation. Their evaluation results show that their model outperforms similar reported results on the task of image and video retrieval. However, it should be noted that they have created image features from rectangular grids in image instead of region features obtained from Duygulu et. al. study.

2.4.8 Mix-Hier

In their study, Carneiro et. al. (Carneiro et. al., 2005) also proposed a method to automatically annotate and retrieve images using a vocabulary of images semantics, where they denote the method as Mix-Hier. Their novelty includes a discriminant formulation of the problem, a multiple instance learning solution that enables the estimation of concept probability distributions without prior image segmentation, and a hierarchical description of the density of each image class that enables very efficient training. As in MBRM, they divide the image in to rectangular grids and extract the features. For the region features they convert RGB color space to YBR color space and compute 8x8 Discrete Cosine Transform (DCT). Their evaluations

show that Mix-Hier outperforms all other methods that are used the same dataset in the literature.

CHAPTER THREE

IMAGE AUTO-ANNOTATION BASED ON COMBINATION OF TEXT AND VISUAL CLUSTERING

3.1 Introduction

In this chapter, we present our novel methodology that is developed for auto-annotation and image retrieval problems. We call the process “combination of text and visual clustering” because we are training the system with text annotation clusters and then each image is described with image features in those clusters. The main hypothesis is that the images falling into the same text-cluster can be expressed with common visual features of those images. In order to prove the hypothesis, we set our estimation procedure for finding out the similar visual features in the textually clustered images. When an un-annotated image is given, we find the best image matching in different textual clusters by processing their low-level features and decoupling values in the annotation clusters. Experiments have demonstrated that our proposal has good accuracy and it has high potential for using it for annotation of images and it can also be used for improving the content-based image retrieval quality.

In this chapter, the proposed system will be explained in detail in section 3.2. *C3M* algorithm, training and querying steps of the method will be described in the subsections of section 3.2. In section 3.2 an example will be demonstrated to make the method more understandable. The methodology at the case of data set modification will be explained in section 3.4. The chapter will be concluded with section 3.5.

3.2 Combination of Textual and Visual Clustering

In this section we will describe our approach in detail. The main idea of the methodology to auto-annotate and retrieve images is simple. We presume that the

images with similar annotations must share at least few similar low-level features. If this is true; may this correlation be used to associate some *low-level* visual features addressing the more detailed perceptual aspects and *high-level* semantic features? More formally, images falling into the same text cluster can be described with their common visual features and could be stored in the same cluster that contains low-level features. One can easily think that it is possible to find images with annotations as counter examples not obeying the underlying hypothesis. However it is not possible to say that the images with similar annotations never share similar low-level features. Moreover, our main hypothesis relies on to intersection of textual and low-level visual features. We present few examples of annotation clusters and their corresponding images in section Appendix D. In these examples you can see that images in an annotation cluster have common visual features. It is clear that our approach highly depends on training set descriptors and the images must be annotated with care.

Hypothesis: Images falling into the same text-cluster should also have some common visual features of those images.

Proof: Let us assume the statement,

p = “images falling in to the same text-cluster can be described with common visual features of those images”

Is false.

So the statement,

$\neg p$ = “images falling in to the same text-cluster can not be described with common visual features of those images”

Will be true.

It is obvious that the $\neg p$ is false. Because, if we cluster image annotations, it means that the images having the same or similar objects will fall in to the same text-cluster. Because of the annotations are tagged to images by considering the objects in the images by annotators, similar images will be annotated with similar keywords.

Images having common objects will be annotated with the common keywords, so they will fall into the same text cluster. Common objects also mean common low-level features; because low-level features are the numerical descriptors of object features (shape, colour, texture, size and location) of the image. This does not mean that all images in the annotation cluster have common low-level features, however all images in a cluster can be represented with most common features. For example let us consider images having annotations of “tiger, tree and grass”. It is obvious that, these images will fall into the same text cluster and they also have common low-level features. Based on this discussion we can easily say that $\neg p$ is false.

If $\neg p$ is false $\Rightarrow p$ is true. \square

Figure 3.1 presents the overview of proposed system. There are two distinct image sets for training and testing the system. At the first step, training image annotations are clustered to obtain text clusters and then low-level image region descriptors are clustered for generating image blobs to describe the images. For a given query image a new feature vector is created by the means of blobs created in training step. Images that are most similar to query image, retrieved and their high frequent annotations are used as auto-annotation.

In our approach, images are firstly clustered by considering their text annotations as we said above. The images are also segmented into regions and then those regions are clustered based on their low-level visual features. In literature the word *blob* (Duygulu, P., et. al., 2002) is used for naming those clusters. The feature vector of each image in training set is then changed to a dimension equal to the number of visual clusters (number of blobs) where each entry of the new feature vector signifies the contribution of the image to the corresponding blob. Then a matrix is created for each textual cluster, and descriptors of images in the clusters are replaced with the blob feature vectors.

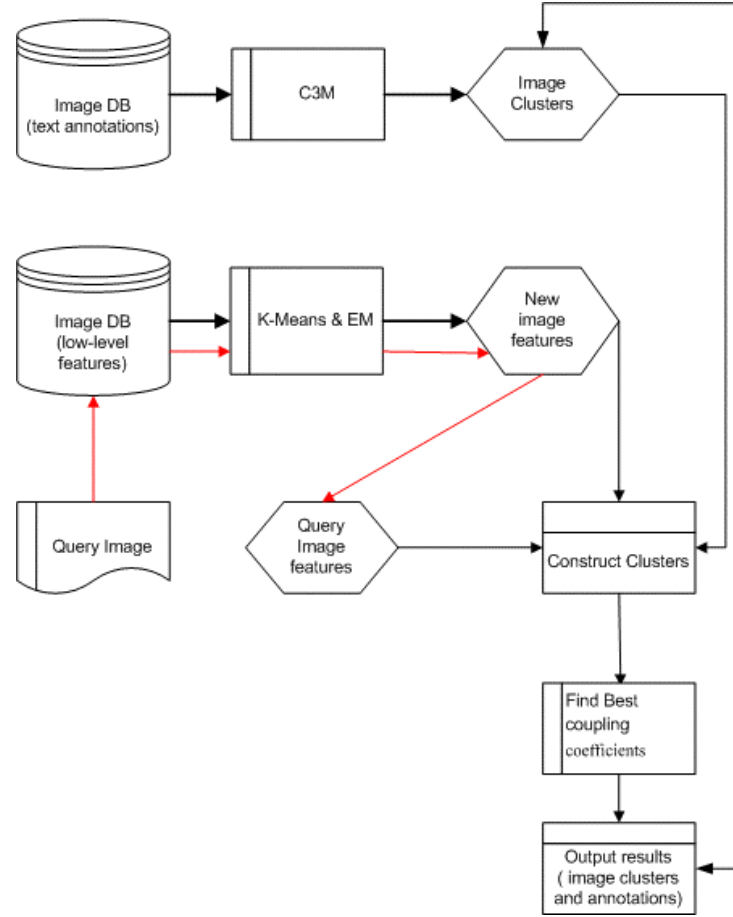


Figure 3.1 Diagram of image retrieval and auto-annotation process.

A feature vector is also created for the query image and it is then appended to the matrix of each cluster. Images in the clusters giving the highest coupling coefficient are considered for retrieval and retrieved image annotations are considered as candidate annotations for query image.

The proposed method consists of two steps that are training and querying steps as defined in following sections. Both in training and querying steps *C3M* algorithm has been used. For that reason, in the following subsection we will describe the *C3M* algorithm firstly. After the explanation of *C3M*, detailed description of training and querying steps will be illustrated in the following sections.

3.2.1 Cover Coefficient based Clustering (C3M)

In this section Cover Coefficient-based Clustering Methodology that employs the idea of text document clusters will be explained. *C3M* originally proposed by Can, F., & Ozkaran E.A. (1990) The base concept of the algorithm, the Cover Coefficient (CC) concept, provides a means of estimating the number of clusters within a document database and relates indexing and clustering analytically. The CC concept is used also to identify the cluster seeds and to form clusters around these seeds.

Cover Coefficient-based Clustering Methodology envisions document clusters as cluster seeds and member documents. Cluster seeds are selected by employing the seed power concept and the documents with the highest seed power are selected as the seed documents. In their paper Can, F. and Ozkaran E.A., showed that the complexity of *C3M* is less than most other clustering algorithms, whose complexities range from $O(m^2)$ to $O(m^3)$. Also their experiments show that *C3M* is time efficient and suitable for very large databases. Its low complexity is experimentally validated. *C3M* has all the desirable properties of a good clustering algorithm. The retrieval experiments show that the information-retrieval effectiveness of the algorithm is compatible with a very demanding complete linkage clustering method that is known to have good retrieval performance.

3.2.1.1 C3M Algorithm

C3M algorithm is partitioning clustering type (clusters cannot have common documents). A generally accepted strategy to generate a partition is to choose a set of documents as the seeds and to assign the ordinary (non-seed) documents to the clusters initiated by seed documents to form clusters. This is the strategy used by *C3M*. Cover coefficient, CC, is the base concept of *C3M* clustering. The CC concept serves to;

- i. Identify relationships among documents of a database by use of the CC matrix,

- ii. Determine the number of clusters that will result in a document database;
- iii. Select cluster seeds using a new concept, cluster seed power;
- iv. Form clusters with respect to $C3M$, using concepts (i)-(iii);
- v. Correlate the relationships between clustering and indexing.

$C3M$ is a seed-based partitioning type clustering scheme. Basically, it consists of two different steps:

- i. Cluster seed selection
- ii. Cluster construction.

D matrix is the input for $C3M$, which represents documents and their terms. We assume that each document contains n terms and database consists of m documents. We need to construct C matrix, in order to employ cluster seeds for $C3M$. C , is a document-by-document matrix whose entries c_{ij} ($1 < i, j < m$) indicate the probability of selecting any term of d_i from d_j . In other words, the C matrix indicates the relationship between documents based on a two-stage probability experiment. The experiment randomly selects terms from documents in two stages. The first stage randomly chooses a term t_k of document d_i ; then the second stage chooses the selected term t_k from document d_j .

3.2.1.2 C Matrix

For the calculation of C matrix, c_{ij} , one must first select an arbitrary term of d_i , say, t_k , and use this term to try to select document d_j from this term, that is, to check if d_j contains t_k . In other words, we have a two-stage experiment. Each row of the C matrix summarizes the results of this two-stage experiment.

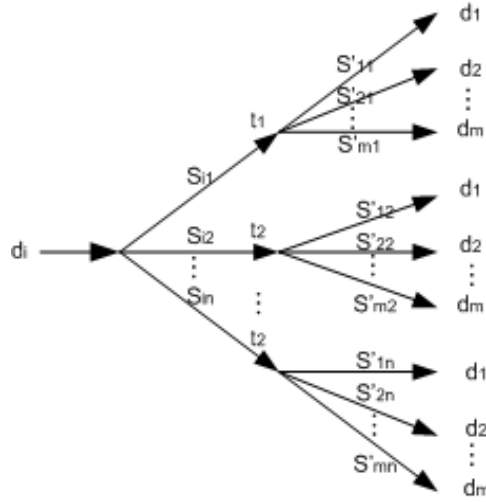


Figure 3.2 Hierarchical representation of two-stage probability model for d_i of D Matrix.

Let s_{ik} indicate the event of selecting t_k from d_i at the first stage, and let s'_{jk} indicate the event of selecting d_j , from t_k at the second stage. In this experiment, the probability of the simple event “ s_{ik} and s'_{jk} ” that is, $P(s_{ik}, s'_{jk})$ can be represented as $P(s_{ik}) \times P(s'_{jk})$. To simplify the notation, we use s_{ik} and s'_{jk} respectively as in equation 3.1.a and 3.1.b, for $P(s_{ik})$ and $P(s'_{jk})$.

$$s_{ik} = \frac{d_{ik}}{\sum_{h=1}^n d_{ih}} \quad 1 \leq i, j \leq m, 1 \leq k \leq n \quad (eq:3.1.a)$$

$$s'_{jk} = \frac{d_{jk}}{\sum_{h=1}^m d_{hk}} \quad 1 \leq i, j \leq m, 1 \leq k \leq n \quad (eq:3.1.b)$$

By considering document d_i , we can represent the D matrix with respect to the two-stage probability model, as shown in Figure 3.2. Each element of C matrix, c_{ij} , (the probability of selecting a term of d_i from d_j) can be founded by summing the probabilities of individual path from d_i to d_j . c_{ij} can be calculated as in eq. 3.2.a or, as defined in eq. 3.2.b.

$$c_{ij} = \sum_{k=1}^n s_{ik} s'_{jk} \quad (eq:3.2.a)$$

$$c_{ij} = \sum_{k=1}^n (\text{prob. of selecting } t_k \text{ from } d_i) \times (\text{prob of selecting } d_j \text{ from } t_k) \quad (eq:3.2.b)$$

To decrease the complexity of calculating c_{ij} , this can be rewritten as in eq. 3.3.

$$c_{ij} = \alpha_i \sum_{k=1}^n d_{ik} \beta_k d_{jk}, \text{ where } 1 \leq i, j \leq m \quad (eq:3.3)$$

In eq. 3.3, α_i and β_k are reciprocals of the i^{th} row sum and k^{th} column sum, respectively, as shown in eq. 3.4.a and eq. 3.4.b.

$$\alpha_i = \frac{1}{\sum_{j=1}^n d_{ij}}, \text{ where } 1 \leq i \leq m \quad (eq:3.4.a)$$

$$\beta_k = \frac{1}{\sum_{j=1}^m d_{jk}}, \text{ Where } 1 \leq k \leq n \quad (eq:3.4.b)$$

Properties of C Matrix:

The following properties hold for the C matrix:

- For $i \neq j$, $0 \leq c_{ij} \leq c_{ii}$ and $c_{ii} > 0$ $c_{i1} + c_{i2} + c_{i3} + \dots + c_{im} = 1$
- If none of the terms of d_i is used by the other documents, then $c_{ii}=1$ otherwise, $c_{ii} < 1$.
- If $c_{ij} = 0$, then $c_{ji} = 0$, and similarly, if $c_{ij} > 0$, then $c_{ji} > 0$; but in general, $C_{ij} \neq C_{ji}$.
- $c_{ii} = c_{jj} = c_{ij} = c_{ji}$ iff d_i and d_j are identical.

From these properties of the C matrix and from the CC relationships between two document vectors, c_{ij} can be seen to have the meaning in eq. 3.5.

$$c_{ij} = \begin{cases} \text{extent to which } d_i \text{ is covered by } d_j \text{ for } i \neq j \\ \text{(coupling of } d_i \text{ with } d_j), \\ \text{extent to which } d_i \text{ is covered by itself for } i=j \\ \text{(decoupling of } d_i \text{ from the rest of the documents)} \end{cases} \quad (eq:3.5)$$

To obtain a better understanding of the meaning of the C matrix, consider two-document vectors d_i and d_j . For these document vectors, four possible relationships can be defined in terms of the C matrix entries (i.e., in terms of c_{ii} , c_{ij} , c_{jj} , and c_{ji}):

- *Identical documents:* Coupling and decoupling of any two such documents are equivalent. Furthermore, the extent to which these two documents are covered by other documents is also identical (i.e., $c_{ik} = c_{jk}$, where $1 \leq k \leq m$). Similarly, the extent to which these two documents cover other documents is the same (i.e., $c_{ki} = c_{kj}$, where $1 \leq k \leq m$).
- *Overlapping documents:* Each document will cover itself more than any other ($c_{ii} > c_{ij}$, $c_{jj} > c_{ji}$). However, this does not provide enough information to compare c_{ii} with c_{jj} and c_{ij} with c_{ji} . This is because these values are also affected by the couplings with the other documents of the database.
- *A document is a subset of another document:* Let d_i be a subset of d_j . Since d_i is a subset of d_j the extent to which d_i is covered by itself (c_{ii}) will be identical to the extent to which d_i is covered by d_j (c_{ij}). Furthermore, since d_j contains all of the terms of d_i as well as some additional terms, then the extent to which d_j covers itself will be higher than the extent to which d_i covers itself (i.e., $c_{jj} > c_{ii}$). By similar reasoning, the extent to which d_j covers d_i is higher than the extent to which d_i covers d_j ($c_{ij} > c_{ji}$).
- *Disjoint documents:* Since d_i and d_j do not have any common terms, then they will not cover each other ($c_{ij} = c_{ji} = 0$). Obviously, the documents will cover themselves. However, because these documents may also be coupled with the others, c_{ii} and c_{jj} may be less than 1.

As can be seen from the foregoing discussions, in a D matrix, if d_i ($1 \leq i \leq m$) is relatively more distinct (i.e., if d_i contains fewer terms that are common with other documents), then c_{ii} will take higher values. Because of this, c_{ii} is called the decoupling coefficient, δ_i , of d_i . (Notice that δ_i is a “measure” of how much the document is not related to the other documents, and this is why the word coefficient is used.)

The sum of the off-diagonal entries of the i^{th} row indicates the extent of coupling of d_i with the other documents of the database and is referred to as the coupling coefficient, ψ_i , of d_i . From the properties of the C matrix,

$\delta_i = c_{ii}$: decoupling coefficient of d_i

$\psi_i = 1 - \delta_i$: coupling coefficient of d_i

$$\delta = \sum_{i=1}^m \frac{\delta_i}{m}, \text{ where } 0 < \delta < 1 \quad (eq:3.6.a)$$

$$\psi = \sum_{i=1}^m \frac{\psi_i}{m} \quad \text{where } 0 \leq \psi \leq 1 \quad (eq:3.6.b)$$

3.2.1.3 C' Matrix

By following a methodology similar to the construction of the C matrix, we can construct a term-by-term C' matrix of size n by n for index terms. In our studies we use C' matrix to construct a thesaurus specific to the data set used.

Its elements are defined in eq. 3.7.a and it is also can be defined as in eq. 3.7.b.

$$c'_{ij} = \sum_{k=1}^n s'_{ki} s_{kj} \quad (eq:3.7.a)$$

$$c'_{ij} = \sum_{k=1}^n (\text{prob. of selecting } d_k \text{ from } t_i) \times (\text{prob of selecting } t_j \text{ from } d_k) \quad (eq:3.7.b)$$

By using the definitions of the S and S' matrices, the entries of the C' matrix can be defined as in eq. 3.8.

$$c'_{ij} = \beta_i \sum_{k=1}^m d_{ki} \alpha_k d_{kj} \quad (eq:3.8)$$

Similar to C matrix, $\delta' = c_{jj}$ and $\psi'_j = 1 - \delta'$

3.2.1.4 Number of Cluster Hypothesis

The prediction of number of clusters is one of the major issues of cluster analysis methodologies. It is obvious that, the number of clusters within a database should be high if individual documents are dissimilar, and low otherwise. However, the similarity of documents is not much help in obtaining the number of clusters. This is because; it is difficult to predict a similarity threshold (stopping rule) that will yield the desired number of clusters. In an effort to provide a solution to this problem, $C3M$ use the CC concept to predict the number of clusters. The diagonal entries of the C matrix yield the number of clusters (n_c), as in eq. 3.9.

$$n_c = \sum_{i=1}^m \delta_i m \quad (eq:3.9)$$

This equation is consistent because a database with similar documents will have a low δ (decoupling) and few clusters. On the other hand, a database with dissimilar documents will have a high δ and many clusters.

3.2.1.5 Cluster Seed Selection

Clusters seeds are the representative document of clusters in cluster analysis. The $C3M$ is a seed-oriented document-clustering methodology; that is, n_c documents are selected as cluster seeds, and non-seed documents are grouped around the seeds to form clusters. Seed documents must be well separated from each other and at the

same time must be able to pull non-seed documents to themselves. *C3M* defines the cluster seed power P_i of d_i for binary matrixes as in eq. 3.10.a and for weighted matrixes P_i of d_i is defined in eq. 3.10.b.

$$P_i = \delta_i \psi_i \sum_{j=1}^n d_{ij} \quad (\text{eq:3.10.a})$$

$$P_i = \delta_i \psi_i \sum_{j=1}^n (d_{ij} \delta'_j \psi'_j) \quad (\text{eq:3.10.b})$$

3.2.1.6 The C3M Algorithm

C3M is a partitioning-type clustering algorithm that operates in a single pass. A brief description of the algorithm is as follows:

```
[1] Determine the cluster seeds of the database.
[2] i=1;
    repeat /* construction of clusters */
        if  $d_i$  is not a cluster seed then
            begin
                Find the cluster seed (if any) that maximally covers  $d_i$ ;
                if there is more than one cluster seed that meets this
                condition, assign  $d_i$  to the cluster whose seed power value
                is the greatest among the candidates;
            end
        i=i+1;
    until i > m.
[3] If there remain unclustered documents, group them into a ragbag
    cluster (some nonseed documents may not have any covering seed
    document).
```

3.2.2 Training

In section 3.2.1 we describe the details of *C3M* algorithm that is used in training and querying steps in our methodology. The training step, based on the combination of textual and visual clustering has three sub-steps: *textual clustering*, *visual*

clustering and *replacing*. In this section we will describe the training step by considering only the colour features, in the other section we will describe how multiple features can be combined.

The first sub-step, *textual clustering* occurs at training phase and all of training images, T , are clustered according to their textual annotations by using *C3M*.

At the second sub-step, all image regions are clustered according to visual similarities by *k*-means clustering algorithm where the number of clusters is $n_{c-color}$ for colour features. Each cluster will represent a blob, where number of blobs will be equal to number of low-level feature clusters. Let $K(t)$ is the *k*-means function and T_s is set of regions, clustering can be formally defined as in eq. 3.11.

$$K_c(t):T_s \rightarrow Mc, s(Mc)=n_{c-color} \quad (eq:3.11)$$

Where $T_s=\{t: t \text{ is the segment of image } I, \forall I \in T\}$ contains all the regions of all images. The dimension of image feature vectors after $K(t)$ transformation is equal to the number of elements (number of blobs) in Mc . Then each image, I_j , is represented as a vector in $n_{c-color}$ dimensional space and holds the corresponding blob of region t for color clusters.

$$I_j = \langle i_{j1}, i_{j2}, \dots, i_{jnc-color} \rangle$$

Each entry of new feature vector signifies the contribution of corresponding colour cluster (blob) to the image j . Formally, let i_{jk} indicates the k^{th} entry of vector I_j which is for j^{th} image in collection. More formally, an arbitrary entry of vector I_j can be defined as follows:

$$i_{jk} = \begin{cases} \sum w_t & \text{if } K(s_t)=m_k, K(s_p)=m_k \text{ for } \forall s_t \in I_j, \exists s_p \in I_j, p \neq t \\ w_t & \text{if } K(s_t)=m_k, K(s_p) \neq m_k \text{ for } \forall s_t \in I_j, \forall s_p \in I_j, p \neq t \end{cases} \quad (eq:3.12)$$

In other words, the $n_{c-color}$ dimensional I_j feature vector gives the information about its regions. I_j vector is weighted where each entity (represents blob) indicates

the percentage of corresponding image region covered in the image. If more than one region corresponds to the same blob, their weights are summed as in eq. 3.12. Also following condition holds for each image.

For each I_j vector, $0 \leq i_{jt} \leq 1$, where $1 \leq t \leq n_{c-color}$

The vector is normalized so that sum of the entries of vector I_j is equal to 1. In another words, in this step, each image is transformed into a dimension, called blob space. We have constructed new feature vectors for each image in the training set by using k -means clustering. The new features for each image are consisting of blobs and weights that represent the segments of images.

The process of assigning blobs to images is summarized in Figure 3.3. Once clustering image regions creates the blobs, each image blob is associated with image regions. Each image vector is created by the means of their assigned blobs and weighted as the regions' size.

At the end of first two steps of training phase, we then have two sets of clusters: First set contains the clusters of images based on text annotations and the second one contains clusters of images based on visual features of their regions.

The last step of training is replacing the image descriptors in textual clusters with visual features. More clearly, we use textual clustering of images to train the system. However, blob descriptors of images within the clusters are used for annotation and retrieval task. Each image described with “number of blobs”-dimensional feature vector, in the resulting matrixes. This concludes the training phase and forms a combination of textual and visual features of image collection. This is the most important phase of our approach, which is based on the hypothesis that images with similar annotations should also have similar low-level features, and images falling into the same text cluster should also have common visual features and could be stored in the same colour cluster. We call \overline{D} matrixes to these blob based matrixes for distinguishing them from other document clusters.

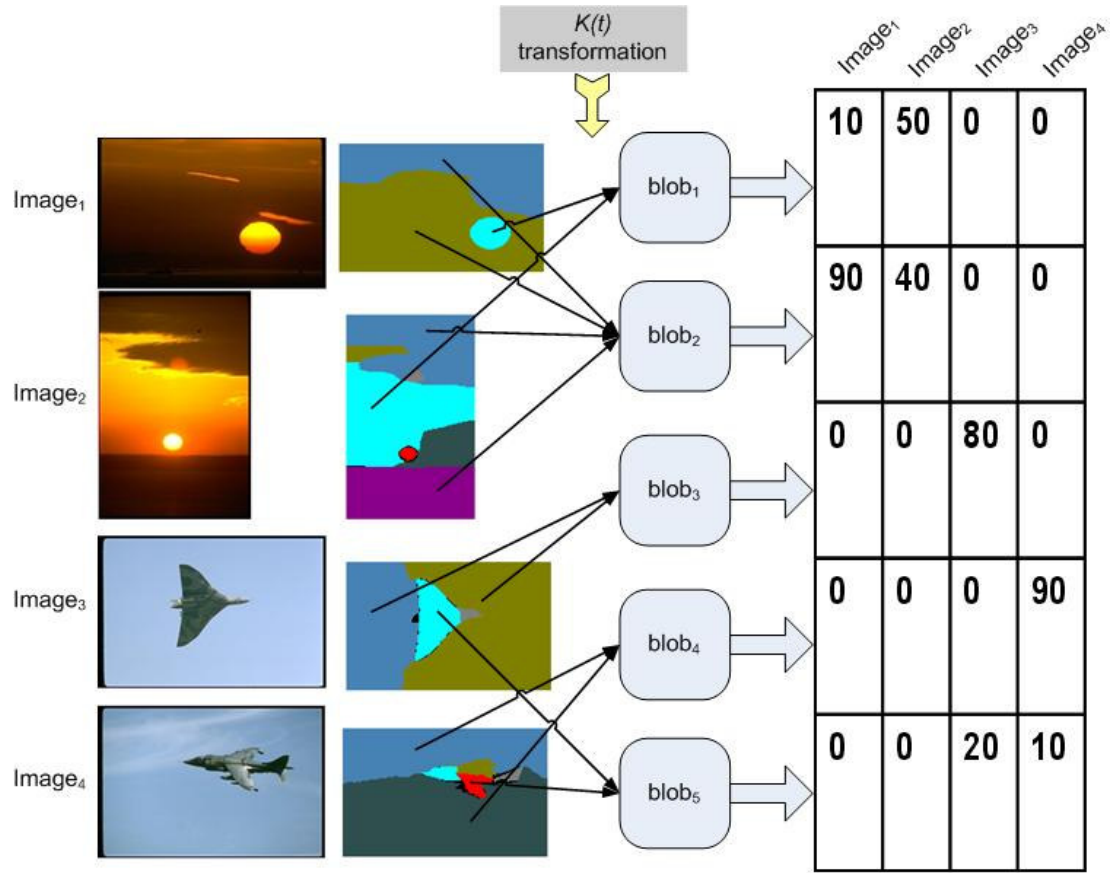


Figure 3.3 The process of assigning blobs to images and generating $blob \times image$ matrix.

At the third step, image descriptors in the annotation cluster are replaced with the vectors, which contain blob information. At the end of this step, the system becomes ready to query.

3.2.2.1 Multiple Low-Level Feature as Image Descriptor

In traditional image retrieval it is an issue to combine more than one low-level feature as we stated in the chapter one. The reason is the applied distance metric creates different measures for each type of feature, and when we consider all features together, this will be meaningless result.

We have described how blob based feature vector is created for each image in the previous section. Due to the proposed method based on blobs instead of low-level

features themselves, it will not be difficult to use (combine) more one low-level feature type as the image descriptor.

It is not a difficult process to combine multiple low-level features as an image descriptor for our methodology. As the first step, regions must be clustered by considering each low-level feature individually. So for each low-level feature type, a different blob set will be generated. For a given image a separate feature vector is organized for each feature type as described in the previous section. To combine more than one feature as a single image descriptor, blob vectors of each feature type must be concatenated as in the Figure 3.4. Now, the resulting feature vector contains the information about the selected low-level feature types.

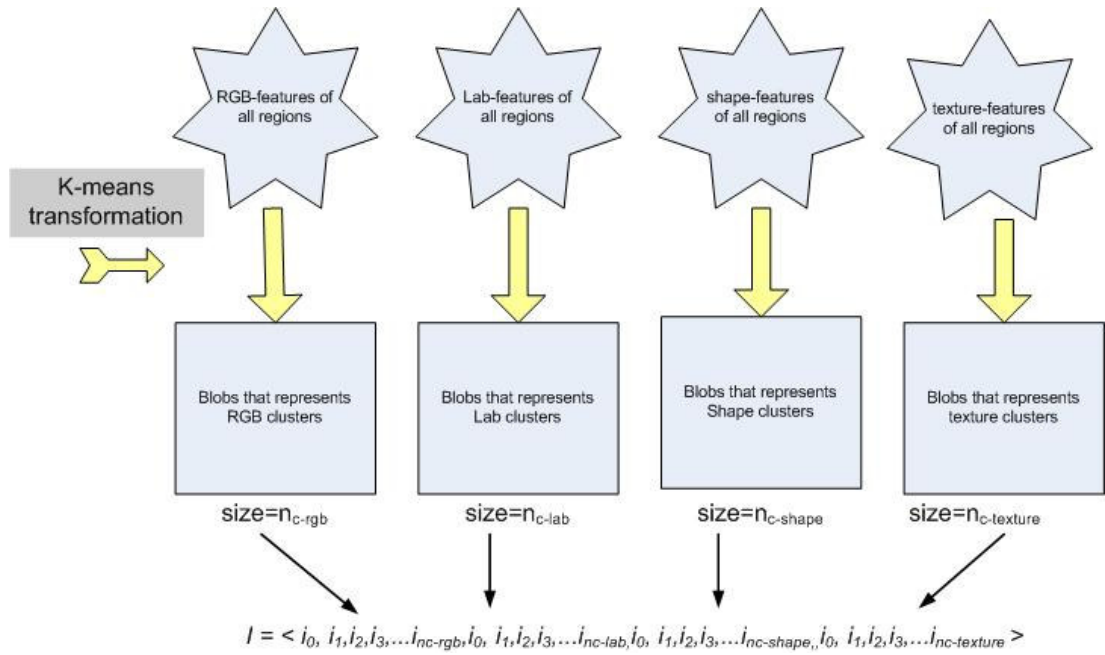


Figure 3.4 Combining multiple blobs (low-level features) to describe an image I .

This property should not be confused with combining annotations with low-level features. Because we don't use annotations as the image descriptors, instead we use them to cluster images.

3.2.3 Auto-Annotation and Image Retrieval

After training the system, we have a cluster of images (clustered by their annotations) where each image in the clusters is represented by blobs, which are visual features. In query phase, a feature vector based on blobs obtained at the training step is organized for the image to be annotated or retrieved for similarities, as explained in previous sections.

3.2.3.1 Constructing Query Vector

The aim of the proposed method is to auto-annotate the given image and to retrieve similar images. The given image properties must be same as train images' properties in order to compare. In other words, their space (that requires their dimensions) must be the same. Train image properties consist of blobs where query image must also be defined by the means of same blobs.

At the query step, initially we only have the original query image. As in the training step same process is applied for query image to convert it to blob space. That is the algorithm can be defined as follows:

- i. Query image is segmented in to regions.
- ii. For each segment q_s of query image,
 q_s is compared with all segments in T_s ,
blob that corresponds to most similar segment to q_s is
assigned as the corresponding blob of q_s .
- iii. The weight of each blob selected as the amount of region
covers in the image.

3.2.3.2 Query Processing

Originally *C3M* algorithm proposes to make query on cluster seeds rather than comparing all documents sequentially. This approach may be quite useful for the text based clusters and retrieval. Because all clusters seeds are representatives of their cluster members and distance to cluster seed is also distance to the cluster. The benefit of this approach is that it decreases the complexity of query step.

Cluster seed based query is not suitable for our proposed methodology. The reason is; we use text clusters but each image described with low-level features. So selected seeds at annotation clustering step, may not be representatives of the images in the clusters that contains low-level features (\bar{D} matrix). This doesn't mean that there is not any seed in the cluster, but there is no guaranty that seeds of text cluster are also the seeds of low-level feature clusters. With this issue, we develop a query methodology for the proposed method.

Once the query vector is defined, vector representation is appended to every \bar{D} matrix (cluster) of training phase as a new member and then the C matrix is calculated for each \bar{D} matrix. We call \bar{C} matrix to C matrix of \bar{D} as defined in eq. 3.13.

$$\bar{C} = C \text{ matrix of } (\bar{D}) \quad (eq:3.13)$$

\bar{C} Matrix can also be used to measure the probabilities of which of those images in \bar{D} are most close to the query image as described below.

Recall that, coupling value gives how much an image is related to others within the same collection. In another words, an image that shares a lot of common features with the other images has a high coupling, but low decoupling coefficient otherwise. Diagonal entries of C Matrix show the decoupling coefficient of each image in the cluster. Recall that decoupling coefficient represents how an image is not related

with others in the cluster. So if we add the feature vector of query image to any cluster and calculate C matrix, we will obtain the information about how the query image is not-related (also related) with other images in the considered cluster. Also C matrix will give information about the probability of each image in the cluster, how they are related to query image. With this information we specify a distance metric to evaluate how an image in the cluster is related with the query image. We have specified the distance metric as in eq.3.14. In eq: 3.14, $m-1$ is the number of images in cluster \bar{D} , \bar{C}_{mm} is the decoupling value of query image in the considered cluster and \bar{C}_{im} is coupling with image i and query image.

$$\begin{aligned} &\text{For each image } i \text{ in matrix } \bar{D} \\ &dist_i = \bar{C}_{im} * \frac{1}{\bar{C}_{mm}}, \quad dist_i = 1, \text{ if } \bar{C}_{mm} = 0 \end{aligned} \quad (eq:3.14)$$

In eq: 3.14, \bar{C}_{mm} , measures how the query image is not related with the images in the considered cluster. If \bar{C}_{mm} is high (that means it is distinct to other images) $dist_i$ will be low, and if \bar{C}_{mm} is low $dist_i$ will be high. Also, if \bar{C}_{im} that indicates coupling of I_i with query image is high, $dist_i$ will be high. This distance metric implies that if the distance is 1, query image is completely similar with the considered image.

Once the distances of all images are evaluated, images having the highest value are retrieved and high frequent annotations of those retrieved images are selected as auto-annotation.

Theorem: Distance of two images is not greater than 1.

Proof: Recall that distance was defined as in eq: 3.14.

From the properties of $C3M$ (theorem 2, in the original publication), we know that $\min(\bar{C}_{mm}) = \frac{1}{m}$. We also know that $\bar{C}_{im} \leq \bar{C}_{mm}$ again from the definitions of $C3M$.

To make the distance maximum we need to have \bar{C}_{mm} minimum and \bar{C}_{im} maximum.

$$\max(dist_i) = \max(\bar{C}_{im}) * \frac{1}{\min(\bar{C}_{mm})}$$

$$\max(dist_i) = \max(\bar{C}_{im}) * \frac{1}{1/m}$$

$$\text{That is, } \max(dist_i) = \max(\bar{C}_{im}) * m$$

$$\text{Let say, } t = \max(\bar{C}_{im})$$

As mentioned above, we know that $\bar{C}_{im} \leq \bar{C}_{mm}$

$$t = \max(\bar{C}_{im}) \Rightarrow t \leq \bar{C}_{mm} \Rightarrow t \leq \frac{1}{m}, \text{ (we are considering the condition that}$$

$$\bar{C}_{mm} = \frac{1}{m})$$

This implies, $tm \leq \frac{1}{m}m$, where $1 \leq m$

$$\max(dist_i) = tm$$

$$\max(dist_i) \leq \frac{1}{m}m$$

$$\max(dist_i) \leq 1 \quad \square$$

Theorem: Minimum distance of two images is 0.

Proof:

From the definition of $C3M$ we know that $C_{im}, C_{mm} \geq 0$

$$\min(dist_i) = \min(C_{im}) * \frac{1}{\max(C_{mm})}$$

$$\min(dist_i) = 0 * \frac{1}{\max(C_{mm})}$$

$$\min(dist_i) = 0 \quad \square$$

3.3 Numerical Example for Proposed Method

Let us consider a trivial example to demonstrate clearly how our approach works and $C3M$ algorithm is used. In this subsection, variable values (i.e., M_i , $C3M$

clusters) are hypothetical and they are not a result of our real experimentations or calculations. Let us assume that we have 8 images and clustered into 4 clusters ($n_{c-text} = 4$) according to their annotations by *C3M* as shown on Table 3.1. In this example we only consider the color features as the low-level feature of the images.

Assuming that the training set, T , has 8 images and the total number of regions belong to set T is 30, where an image has minimum 2 and maximum 5 regions. Visual features of training images are clustered by using k -means into 7 clusters ($n_{c-color} = 7$). So we have 7 blobs in this simple example.

Table 3.1 Images, their annotations and clusters for simple example.

Image Id	Blob Id	Image Annotation
1	1	Sky, sun, water
2	1	Sky, sun, trees
3	2	Tiger, grass
4	2	Tiger, grass, trees
5	2	Tiger, grass, water
6	3	Car, road, people
7	4	Horse, mare, trees
8	4	Horse, trees

Table 3.2 shows images and their regions, the percentage of region area (W) covered in the image, the average RGB color values of the regions' pixels and respective blob (M_i). Notice that sum of all region areas of any image is 100 that can easily normalized to 1. *C3M* considers the percentage of region within the entire image as the region's weight. Notice that more than one region of the same image may be classified into the same cluster. For example image 1's regions 2, 3 and 4 are fall into cluster 5. So, they are in the same cluster and the weight is the sum of W values of those regions, which have same M_i values. A new feature $m \times n_{c-color}$ matrix I (Figure 3.5), is created where each entry, i_{jk} , ($1 < j < m$, $1 < k < n_{c-color}$, m is the number of images), indicates the weight of each blob in j^{th} image. The final $Image \times blob$ matrix is created as in Figure 3.5.

$$I = \begin{bmatrix} 0 & 40 & 0 & 0 & 60 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 70 & 0 \\ 0 & 0 & 0 & 0 & 10 & 10 & 80 \\ 70 & 0 & 0 & 0 & 0 & 10 & 20 \\ 0 & 50 & 0 & 0 & 0 & 50 & 0 \\ 0 & 40 & 5 & 25 & 30 & 0 & 0 \\ 20 & 0 & 0 & 80 & 0 & 0 & 0 \\ 0 & 0 & 65 & 35 & 0 & 0 & 0 \end{bmatrix}$$

Figure 3.5 *Image* \times *blob* matrix for the given example. Each entity denotes the region in the image represented by the corresponding blob.

At the end of the first phase, we have trained the system according to text-annotations, created a new matrix I . We separate the images in to \overline{D} matrixes by considering annotation clusters of images in Table 3.1 that are obtained with $C3M$. \overline{D} matrixes are as in Figure 3.6.

$$\begin{aligned} \overline{D}_{c1} &= \begin{bmatrix} 0 & 40 & 0 & 0 & 60 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 70 & 0 \end{bmatrix} & \overline{D}_{c2} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 10 & 10 & 80 \\ 70 & 0 & 0 & 0 & 0 & 10 & 20 \\ 0 & 50 & 0 & 0 & 0 & 50 & 0 \end{bmatrix} \\ \overline{D}_{c3} &= [0 \quad 40 \quad 5 \quad 25 \quad 30 \quad 0 \quad 0] & \overline{D}_{c4} &= \begin{bmatrix} 20 & 0 & 0 & 80 & 0 & 0 & 0 \\ 0 & 0 & 65 & 35 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Figure 3.6 Clusters generated with $C3M$ for the images shown in Table 3.1. The images in each cluster \overline{D}_{ci} are distinct.

Assume that we have been given a query image Q and asked to find out best annotations and/or retrieve similar images for Q , where the properties of the Q is as shown in Table 3.3. For our simple example, assume that we have calculated a query vector for the given query image as follows:

$$q' = \langle 0, 50, 0, 0, 20, 30, 0 \rangle$$

M_i values for the query image are evaluated according to $K(t)$ function as in the training phase (eq. 3.11) and the new feature vector q' is obtained for query image Q . Let us continue on the example and assume that q' is the member of each cluster by adding query vector to \overline{D}_{ci} clusters as the last image as in Figure 3.7.

Table 3.2 Image segments and their corresponding k -means clusters (blobs).

Image ID	Segment ID	W	R	G	B	$M_i(\text{blob})$
1	1	40	255	117	0	2
1	2	30	228	217	200	5
1	3	10	228	216	195	5
1	4	20	220	210	200	5
2	1	30	225	119	0	2
2	2	70	37	20	6	6
3	1	30	2	1	0	7
3	2	50	7	5	0	7
3	3	10	228	218	194	5
3	4	10	22	29	6	6
4	1	50	6	5	4	1
4	2	20	5	3	1	1
4	3	10	41	31	30	6
4	4	10	210	105	62	3
4	5	5	219	97	57	3
4	6	5	215	100	60	3
5	1	20	36	18	5	6
5	2	30	33	19	4	6
5	3	50	255	111	4	2
6	1	40	250	121	6	2
6	2	20	210	216	200	5
6	3	10	214	216	202	5
6	4	20	45	30	5	4
6	5	5	202	107	59	3
6	6	5	80	65	37	4
7	1	80	53	30	7	4
7	2	20	7	5	2	1
8	1	30	55	30	8	4
8	2	65	205	104	60	3
8	3	5	52	27	9	4

Table 3.3. Query Image and its segments for simple example.

Segment ID	W	R	G	B	M_i
1	20	253	111	1	2
2	20	223	214	201	5
3	30	235	121	3	2
4	30	35	23	7	6

$$\begin{aligned}
\overline{D}_{c1} &= \begin{bmatrix} 0 & 40 & 0 & 0 & 60 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 & 70 & 0 \\ 0 & 50 & 0 & 0 & 20 & 30 & 0 \end{bmatrix} & \overline{D}_{c2} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 10 & 10 & 80 \\ 70 & 0 & 0 & 0 & 0 & 10 & 20 \\ 0 & 50 & 0 & 0 & 0 & 50 & 0 \\ 0 & 50 & 0 & 0 & 20 & 30 & 0 \end{bmatrix} \\
\overline{D}_{c3} &= \begin{bmatrix} 0 & 40 & 5 & 25 & 30 & 0 & 0 \\ 0 & 50 & 0 & 0 & 20 & 30 & 0 \end{bmatrix} & \overline{D}_{c4} &= \begin{bmatrix} 20 & 0 & 0 & 80 & 0 & 0 & 0 \\ 0 & 0 & 65 & 35 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 & 20 & 30 & 0 \end{bmatrix}
\end{aligned}$$

Figure 3.7 Cluster representations after appending query vector into each cluster.

Then, the correlation of the query image q' with each clusters' image is calculated. In another words, how query image is correlated with other images in each cluster is founded from $C3M$ properties. As described in Section 3.2.2, $C3M$ algorithm defines the correlation of q' with coupling coefficient ($\psi_{q'}$) and the result of C matrixes of each \overline{D}' shown in Figure 3.8.

$$\begin{aligned}
\overline{C}_1 &= \begin{bmatrix} 0.58 & 0.10 & 0.32 \\ 0.10 & 0.56 & 0.34 \\ 0.32 & 0.34 & 0.34 \end{bmatrix} & \overline{C}_2 &= \begin{bmatrix} 0.68 & 0.17 & 0.05 & 0.10 \\ 0.17 & 0.75 & 0.05 & 0.03 \\ 0.05 & 0.05 & 0.50 & 0.40 \\ 0.10 & 0.03 & 0.40 & 0.47 \end{bmatrix} \\
\overline{C}_3 &= \begin{bmatrix} 0.66 & 0.34 \\ 0.34 & 0.66 \end{bmatrix} & \overline{C}_4 &= \begin{bmatrix} 0.76 & 0.24 & 0.00 \\ 0.24 & 0.76 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}
\end{aligned}$$

Figure 3.8 C Matrixes for each cluster in Figure 3.7.

If we apply the distance metric given in eq. 3.14 to C matrixes in Figure 3.8 we obtain the results as in Table 3.4.

Table 3.4 Query results for the example

Image	1	2	3	4	5	6	7	8
<i>Dist</i>	<i>0.94</i>	<i>1</i>	<i>0.21</i>	<i>0.06</i>	<i>0.85</i>	<i>0.51</i>	<i>0</i>	<i>0</i>

From these results, images (number of retrieved images depends on threshold) having the highest distance are chosen for annotation/retrieval. So, we can say that I_1 and I_2 are most similar to query Image Q and it can be annotated with keywords of *sky*, *sun*, *water* and *trees*. This concludes explanation of the trivial example.

3.4 Maintenance of Training Set

The need for maintenance (update) of training set arise when clustering structures is modified due to the addition of new images or deletion of old images (or both) from the training set. In the literature there are very few maintenance algorithms. In general, these algorithms are developed for growing databases; most of them, however, can also be used for document deletion that they are mostly developed for text documents.

The *cluster-splitting* approach that is one of the traditional methods in the literature treats a new document as a query and compares it with the existing clusters and assigns it to the cluster that yields the best match (Salton, G., & Wong, A., 1978). This approach starts to degenerate clusters when there is relatively small increases (such as 25% to 50%) in the size of the database.

When our proposed method considered, annotation clusters must be modified when there is a need to update training set. Recall that we used $C3M$ to make clusters of image annotations. Can, F., (1993) introduce an algorithm for maintaining $C3M$

clusters named Cover Coefficient-Based Incremental Cluster Maintenance algorithm (C^2ICM).

A brief description of C^2ICM is given in the following; the symbols “ \cup ” and “ $-$ ” indicate the well-known set operations union and difference, respectively.

C^2ICM consist of 3 steps as follows:

- i. Compute n_c and the cluster seed powers of the documents in the updated image database, $D_m = D_m \cup D_{m'} - D_{m''}$ and pick the cluster seeds. (In general $m' \gg m''$).
- ii. Determine D_m the set of documents to be clustered. Cluster these documents by assigning them to the cluster of the seed that covers them most.
- iii. If there were documents not covered by any seed, then group those together into a ragbag cluster.

Once the training set maintained, \bar{D} matrixes are created as described before and same algorithm applies for querying.

3.5 Conclusion

In this chapter we have described the details of our proposed novel strategy to auto-annotation and image retrieval problem. The method has got two main steps where they are training and querying. Both in these two steps we have used the properties of C3M, so the details of C3M also described. In addition to retrieval and auto-annotation, modifying (insert, delete) data set and using multiple feature issues are also discussed. Also, an example is demonstrated to make the proposed method more clear.

CHAPTER FOUR

EXPERIMENTS

4.1 Introduction

In the previous chapter we have described the details of method “Image auto-annotation based on combination of textual and visual clustering”. There is a need to make a performance evaluation for the method in order to show the effectiveness and to make benchmarking. Performance evaluation is not a trivial task because of it’s visually inspection has large number of images which is not being practical. An application has been developed to make performance evaluation for the proposed method’s experiments.

In this chapter we first described the data set we have used in our experiments. Then we show the experimental results of image retrieval techniques that are not based on semantic image retrieval to evaluate the strengths of our method. Then we compare our results with some other similar techniques in the literature. And at last section we have discuss performance of our proposed method.

4.2 Dataset

In our experiments, we have used 4500 images from Corel image dataset to train the system and select 500 images that are distinct from training set to perform evaluations. In the image set, 10 largest regions are extracted from the images and 36 low-level features represent each region. Each image is annotated with at most 5 keywords and there are 371 distinct keywords used for the entire dataset. We have obtained those feature sets from Duygulu et. al. (2002) which is publicly available dataset. Using this dataset, allows us to compare the proposed method with similar models in the literature in a controlled manner.

In this dataset images are first divided into image regions with normalized cuts (Shi and Malik (2000)) algorithm. After the segmentation, size, position, color, texture and shape information of each image region is represented.

The 36 features are:

- Area, x, y, boundary/area, convexity, moment-of-inertia (6)
- Average RGB (3)
- Average RGB (3, duplicated!)
- RGB stdev (3)
- Average L*a*b (3)
- Average L*a*b (3, duplicated!)
- Lab standard deviation (3)
- Mean oriented energy, 30 degree increments (12)

RGB and *Lab* features were duplicated to increase their weight for a specific experiment and we did not subsequently remove the duplicated columns. Also *RGB* features of each region were used to represent the image regions. Texture features of each segment are described with 12-dimensional vector that is constructed from 30-degree increments of mean oriented energy.

In our experiment we have used *size*, *color* and *texture* features to describe each image region where these properties are explained in table 4.1.

Table 4.1 Low-level features used in the experiments.

Low-level Feature	Definition
<i>Size</i>	Represents the portion of the image covered by the region.
<i>Color</i>	Average and standard deviation of (r,g,b), (L,a,b) and (r=R/(R+G+B), g=G/(R+G+B)) over the region.
<i>Texture</i>	Represents by 12 oriented filters, aligned in 30-degree increments.

Size property is used as the weight of each region. In other words *size* is used to identify region's weight in the image in our experiments. We have used color and texture features individually and as combined manner for distinct experiments. When we want to use both color and texture as a single descriptor, corresponding feature vectors are concatenated.

Recall that “the distance metric used to identify relationship between images applies to whole vector and in the case of vector contains more than one feature, obtained distance will not be meaningful” as described in section 3.2.1.1. For that reason we have used color and texture features as combined and individually for the experiments presented in the following sections. Table 4.2 summarises the features used for the following experiments.

Table 4.2 Low-level features used for experiments.

Used Features	Experiment
Color, texture	Traditional image retrieval with minimum region distance (section 4.3.1)
Color, texture	Traditional image retrieval with average minimum distance (section 4.3.1)
Color, color combined with texture	Blob based image retrieval (section 4.3.2)
Color, color combined with texture	Translating Text Space to Image Space (section 4.3.3)

4.2.1 Descriptions of Dataset-files

Data set contains the following files where they are the components of the dataset.

- **Words:** The vocabulary used. We count the words starting at 1, so "city" is word 1. There are 374 distinct keywords used for training images and 260 distinct keywords used for test images.

- *Blob_counts / test_1_blob_counts*: One number per line for the 4500 training / 500 test images, giving the number of blobs used for that image.
- *Blobs / test_1_blobs*: The features for the blobs for the 4500 training / 500 test images, listed in order of images, then decreasing blob size. In order to tell which blob goes with which image, you need either the file *blob_counts*, or the file *document_blobs*.
- *Document_blobs / test_1_document_blobs* : The blob token for each of the 4500 training / 500 test images. Each line has a list of numbers representing indices into the file "blobs". If the image has fewer blobs than the maximum, the row is padded with -99's so that the file can be read as a Matlab matrix.
- *Document_words / test_1_document_words*: The words for each of the 4500 training / 500 test images. Each line has a list of numbers that are indices into the vocabulary file "words". Counting starts at 1. If the image has fewer blobs than the maximum, the row is padded with -99's so that the file can be read as a Matlab matrix.
- *Word_counts / test_1_word_counts*: The number of words for each of the 4500 training / 500 test images.
- *Image_nums / test_1_image_nums*: The Corel image number for the 4500 training / 500 test images. The data can be used with some extent without the images. We provide the image numbers for those who have access to the Corel images.

4.3 Method for Evaluating Performance of Retrieval Effectiveness

Annotation results have been used to perform the evaluation of the proposed methodology since the absence of test beds of used data set (Corel Images). Accuracy of image auto-annotations will also mean accuracy on image retrieval because of; annotations are obtained from the retrieval results.

In our experiments we used precision and recall tests to evaluate the auto-annotation results. Precision and Recall tests for image retrieval is not an easy task in the absence of standart test beds for used image database. Also, it is not easy for auto-annotation, because of semantic similarity of keyword pairs such as *sunset* and *sky*, or *horse* and *mare*. For that reason we need to find out the synonyms of keywords, if there is any, in the dataset to make the evaluations of results better.

We have constructed a two level thesaurus for keywords used in the dataset with *C3M*. There are some other studies in the literature like Maddii, et. al. (2001), Pantel, et. al., (2002) and Maedche, et. al. (2004) that are also studied the thesaurus.

C' matrix in *C3M* algorithm is used to make term clusters. C' matrix also gives information about term correlations in the considered data set, so C' is meaningful to use the matrix to find similar terms (for the dataset) that will yield us synonym terms. For each keyword in the data set, we select two synonym keywords as in eq. 4.1 and and eq. 4.2 with a threshold of 0.05.

$$\text{thesaurus}_1(\text{keyword}_i) = \text{keyword}_j \quad (\text{eq. 4.1})$$

$$\text{thesaurus}_2(\text{keyword}_i) = \text{keyword}_k \quad (\text{eq. 4.2})$$

For eq.4.1 and eq.4.2 following conditions hold:






$$\text{for } \forall m, n \quad c'' = \max(c''_{im}) \quad c''_{im} < \text{threshold}$$

$$c''_{ik} = \max(c''_{in}) \quad c''_{ik} < \text{threshold} \quad n \neq m \quad c''_{ik} < c''_{ij}$$

We have modified the original annotation of Corel test images by adding the synonym of each keyword, after thesaurus constructed specific to the dataset as defined in eq. 4.1 and eq. 4.2. The full list of two level thesaurus list can be found at the appendix section B.1. In this table first column is the index of corresponding keyword in the *words* file (section 4.1.1). Second, is keyword itself, third and the fourth column contains the similar keyword of keyword in column two. Few examples from test image files that their original annotations are modified with two-level thesaurus list are presented in table 4.3.

Querying the test images and comparing the annotations of 500 test images with query image is time consuming task. We have developed software that we called *querying application* and *testing application* to overcome this issue. The querying application is used to make retrieval and auto-annotation based on the discussed methods in the following sections. Querying application applies the methods to 500 distinct test images and stores the retrieval and annotation results in a database. Testing application compares the results of queries with modified annotations of test images and stores the results in *keyword_hit* table by the evaluation of whether it is correctly annotated or not. One word query on *keyword_hit* table gives precision and recall values for the evaluated method. Figure 4.1 presents the database table structures and relations used by the applications.

Table 4.3 Modification of original annotations with thesaurus list.

Image	Original Annotation	Thesaurus applied Annotation
 <p>Image name: 10021.jpeg</p>	<i>Plane, prop, runway</i>	<i>Plane, prop, runway, jet, sky</i>
 <p>Image name: 100031.jpeg</p>	<i>Bear, polar, snow, tundra</i>	<i>Bear, polar, snow, tundra</i>
 <p>Image name: 119064.jpeg</p>	<i>Buildings, skyline, street</i>	<i>Buildings, skyline, street, sky, people</i>
 <p>Image name: 122063.jpeg</p>	<i>Beach, horizon, people, water</i>	<i>Beach, horizon, people, water, sky, sand, swimmers, sunset, pool</i>
 <p>Image name: 10021.jpeg</p>	<i>Field, foals, horses</i>	<i>Field, foals, horses, tree, mare</i>

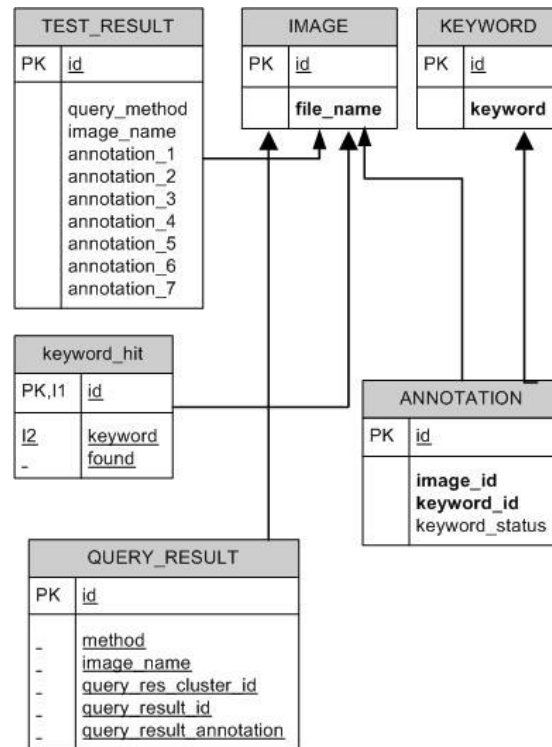


Figure 4.1 The database structure for the test application.

4.4 Experimental results

In this section the result of the experiments where they are performed by using traditional techniques and blob (feature-cluster) based techniques. The common properties of those methods are, they do not consider the high level and low level relation in images, and thus they are not semantic image retrieval techniques. In addition to these experiments we have presented the experiments of proposed method in this section. In the experiments we have used color and texture features separately and/or together to describe an image region. Features that are used for the experiments were described in table 4.2. In all experiments we have used same parameters (such as number of color, texture clusters) to auto-annotate and retrieve images.

For convenience, we prefer to make the formal definitions of variables and functions that have been used in the experiments. In the following sub-sections definitions presented in table 4.4 will be used.

Table 4.4 Definitions of variables used to describe experiments.

Variable	Definition
q	Query image.
T	Set of images that are used for comparisons and training. There are 4500 images in T .
q_i	$\forall q_i \in q$, where q_i is the i^{th} region of query image q .
t	$\exists t \in T$, where t is an image from Image set T .
t_j	$\forall t_j \in t$, where t_j is the j^{th} region of image t .

4.4.1 Traditional Image Retrieval and annotation

In this experiment, only low-level features of images have been used as descriptor to find the similar images. For distinct experiments with traditional methods color and texture features have been used separately. In this experiment query image q is compared with a set of images (T) exhaustively and hence there is no need to training. We have used two different distance measurements that will result to two experiments in this section. The experiments are performed with *averages of minimum distances* and *minimum distance* of regions where we call them *alg-trad1a* and *alg-trad1b* respectively.

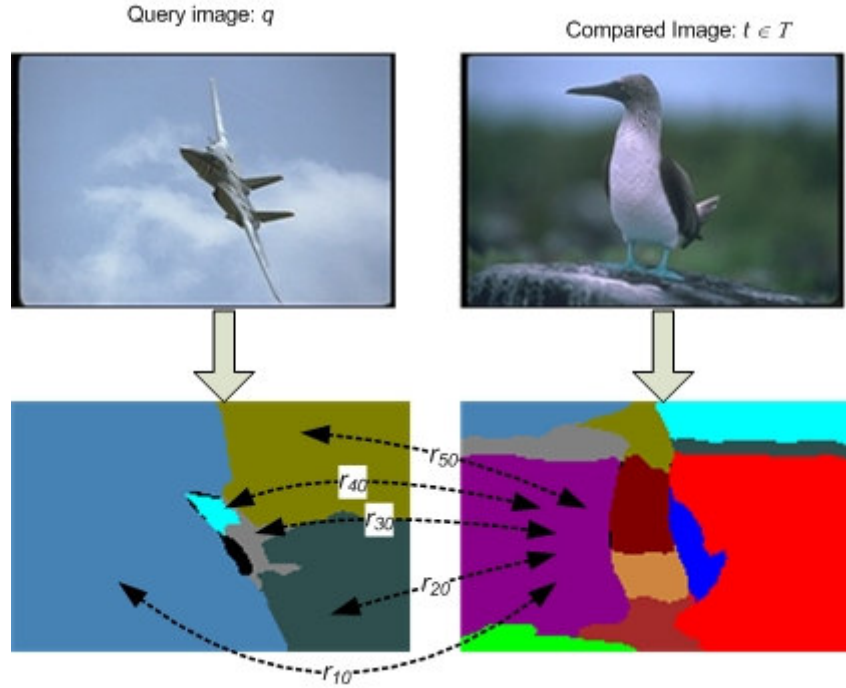


Figure 4.2 All regions of query image are compared with all image regions of image t . In this example, comparison of all regions in q and a region (identified as 0^{th}) in target image is illustrated.

Once the query image is submitted for the auto-annotation and retrieval task, regions of query image are identified. Each region of the query image is compared with the regions of images T (Figure 4.2). The aim of comparison is to find how regions are similar respectively that will yield us the image similarity. In experiment *alg-trad1a* and *alg-trad1b* Euclidean distance used to find similarity between regions.

For *alg-trad1a* we have defined the distance between two images as the smallest distance of image regions. This means that all query image regions are compared with all region of an image from T , and the measured smallest distance is assigned as the distance of those two images. On the other hand, we have defined the distance between two images as the average of minimum distances of image regions for *alg-trad1b*. This means that all query image regions are compared with all region of an image from T , and the average of smallest distances is assigned as the similarity of those two images.

From these definitions we have defined similarity between query image q and image t for *alg-trad1a* as in eq. 4.5.

$$d_{ij} = \text{dist}(q_i, t_j),$$

where d_{ij} is the region similarity of region q_i and t_j (eq. 4.4)

$$\text{dist}(q, t) = \min(d_{ij})$$
(eq. 4.5)

We have also, defined similarity for *alg-trad1b* as in eq. 4.6.

$$\text{dist}(q, t) = (\sum_{k=0}^{K-1} \min(d_{kj})) / K, \text{ where } K = \# \text{ of image regions in } q$$
(eq. 4.6)

Both for *alg-trad1a* and *alg-trad1b*, once the distances between all images in set T and query image are identified images are ranked as their similarities. Top 7 images retrieved as similar images and their high frequent 5 keywords are selected as the auto-annotation of query image.

Results of this method's experiments are presented in the appendix section where A.1.1 shows the recall graph when only color features are considered with *alg-trad1a*, A.1.2 shows precision graph when only color features are considered with *alg-trad1a*, A.1.3 shows recall graph when only texture features are considered with *alg-trad1a* and A.1.4 shows precision graph when only texture features are considered with *alg-trad1a*. And the performance results for *alg-trad1b* can be found at appendix section where A.1.5 shows recall graph when only color features are considered with *alg-trad1b*, A.1.6 shows precision graph when only color features are considered with *alg-trad1b*, A.1.7 shows recall graph when only texture features are considered with *alg-trad1b* and A.1.8 shows precision graph when only texture features are considered with *alg-trad1b*.

4.4.2 Blob Based Image Retrieval and Auto-annotation.

In this experiment feature vectors of all image regions in image set T are clustered and they are categorized in to finite set of clusters that are called *blobs* as in Duygulu, et. al.'s study (2003). We have used *K-means* clustering algorithm (Duda, Hart, & Stork, 2001) to make the clusters of low-level descriptors of each region. We have defined the number of color and texture clusters (blobs) as 200 that will be denoted by $n_{c-color}$ and $n_{c-texture}$ respectively. Once the blobs are defined, image descriptors are organized as each item in feature vector denotes if the corresponding blob contains the region of that image (Figure 4.3). In this representation weight of each item (blob) in the feature vector is defined by the percentage of region in the image. We called *alg-blob* to this method to distinguish it among the other methods.

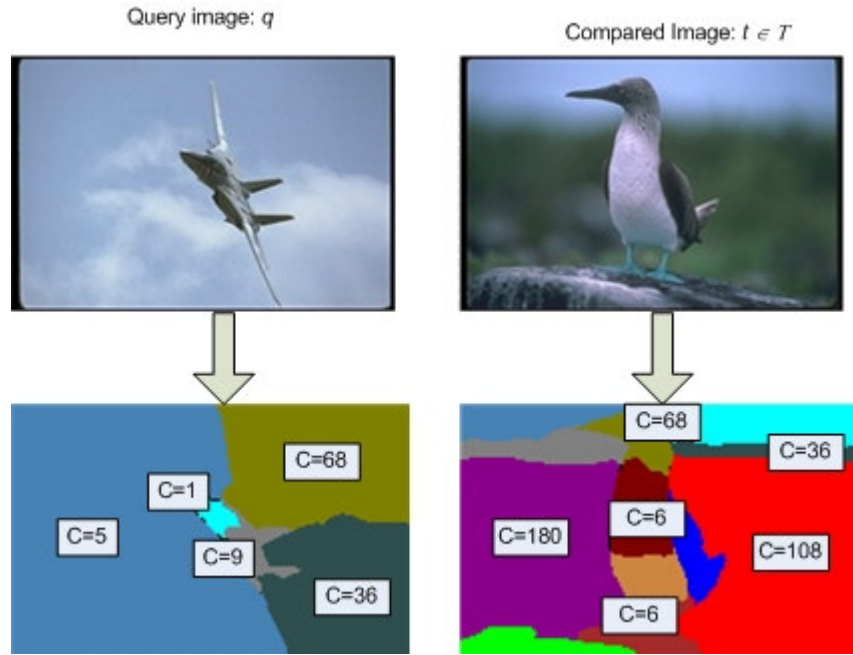


Figure 4.3 First step of cluster-based retrieval is to identify the regions, to evaluate which clusters they belong to.

For convenience, let $K(t)$ define the k -means function and T_s define the set of regions. Clustering can be formally defined as in eq.4.7 and eq.4.8.

$$K_c(t):T_s \rightarrow Mc, s(Mc)=n_{c-color} \quad K_c: \text{k-means function for color features} \quad (eq. 4.7)$$

$$K_t(t):T_s \rightarrow Mt, s(Mt)=n_{c-texture} \quad K_t: \text{k-means function for texture features} \quad (eq. 4.8)$$

In eq. 4.7 and eq. 4.8, $T_s=\{t_s: t_s \text{ is the segment of image } t, \forall t \in T\}$ is a set of regions of all images. Mc and Mt contain all color and texture blobs respectively. The dimension of image feature vectors after $K_c(t)$ transformation is equal to the number of elements in Mc (color blob set). Then, each image, I_j , is represented as a vector in $n_{c-color}$ dimensional space, if we consider the color features only.

$$I_j = \langle i_{j1}, i_{j2}, \dots, i_{jnc-color} \rangle$$

Each entry of new feature vector signifies the contribution of corresponding color cluster to the image j . Formally, let i_{jk} indicates the k^{th} entry of vector I_j which is for j^{th} image in collection. More formally, an arbitrary entry of vector I_j can be defined as in eq. 4.9.

$$i_{jk} = \begin{cases} \sum w_t & \text{if } K(s_t) = m_k, K(s_p) = m_k \text{ for } \forall s_t \in I_j, \exists s_p \in I_j, p \neq t \\ w_t & \text{if } K(s_t) = m_k, K(s_p) \neq m_k \text{ for } \forall s_t \in I_j, \forall s_p \in I_j, p \neq t \end{cases} \quad (eq. 4.9)$$

The vector is normalized so that sum of the entries of vector I_j is equal to 1. In another say, in this step, each image is transformed into a dimension, called region space. We have constructed new feature vectors for each image in the training set T by using k -means clustering. The new features for each image are consisting of *cluster ids* that represent the segments of images.

Once the feature vectors of images are constructed according to image clusters, the system can be queried. At the query phase, query image's (q) feature vector is re-organized by constructed clusters as follows. Feature vectors of each image (q) region, are compared with all image regions in T_s . The blob that is most similar image region is assigned to the corresponding image region. This process is applied for all regions of image; q and $n_{c-color}$ dimensional vector is constructed if only the color features are considered. With this feature vector, similarities with all images

are measured and most similar 7 images are retrieved as the similar images to query image. And annotations of high frequent 5 images are selected as annotation of query image q as in the previous experiment.

The precision and recall measures for this experiment can also be found in appendix section in A.2 section.

4.4.3 *Image Retrieval and Auto-annotation by Translating Text Space to Image Space*

In this section we will describe the experiment of proposed method. Actually the steps of this method contain the process described in the section 4.3.2.

This method consists of two steps that are training and querying. At the training phase, first, images are clustered according their text annotations with C^3M . In our experiments C^3M evaluates the n_{c-text} (number of clusters) as 89 for train set's annotations. Although, each image in train set is annotated with at least 1 and at most 5 keywords, C^3M resulted with few huge clusters. We have specified n_{c-text} as 315 by experimentally to overcome this issue. We choose n_{c-text} as 315 because of it is the maximum number of clusters that C^3M does not generate empty clusters. Secondly image regions are clustered according to their selected low level features with k -means as described above. We selected number of clusters, and $n_{c-color}$ and $n_{c-texture}$ as 200 experimentally. We named the proposed method as *TSIS* that stands for Text Space to Image Space.

Whilst the query phase, images that are most similar to query image according to our proposed methodology are retrieved as the details are described in chapter 3. Retrieved images are ranked and first 7 images are selected as query result as in the previous experiments. Annotations of retrieved images are selected as candidate annotations. We select 5, 7 or 10 (three distinct experiments) high frequent keywords from candidate annotations to auto-annotate the query image. We have considered the experiment that makes 5 word annotations in order to compare the results with







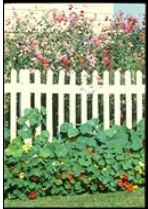
other studies. A total of 260 one-word queries are possible in the test dataset. In other words 260 words that are used in the test data set are used to query the annotation results of test images.

The precision and recall measures for this experiment can also be found in appendix section in A.3 section.

A few query results of proposed method are presented in Table 4.5. First column of Table 4.5 shows the submitted *query image* and second column contains the auto-annotation of query image that the system estimates. Third column contains the original annotations with modified annotations where there are obtained from the thesaurus list described in the previous sections. In the third column, keywords in the parentheses are annotations that are obtained from the thesaurus list. For example if we consider Table 4.5 (b) we can see that original annotations of 13055.jpeg are “*flowers*”, “*leaf*”, “*petals*”, and “*stems*”. Specific to used dataset, similar keywords for each keyword in the original annotations can be obtained from Appendix B.1. In the considered example, “*Flowers*” is some how similar with “*garden*” and “*plants*” when we use the thesaurus list. So, “*garden*” and “*plant*” is added to original annotations where they are in the parentheses. For the second keyword, “*leaf*” similar keywords are “*plants*” and “*flowers*”. Therefore both “*plants*” and “*flowers*” are already in the annotation list; there would not be any change on the original annotations. Also for keywords “*petals*” and “*stems*” nothing is changed in the annotations because of their similar keywords are already in the annotations.

Few examples of image retrieval results are presented in appendix section C. In these examples first image is query image and image that is right hands side of query image, visualizes the regions of query image. Images that follow those images are retrieval results.

Table 4.5 Example Query Results

Image	Auto-annotation	Original Annotation
 (a) 34065.jpeg	Sky, plane, jet, water, people	Jet, plane, sky
 (b) 13055.jpeg	Tree, flowers, bush, ruins, relief	Flowers, leaf, petals, stems, (plants, garden)
 (c) 22004.jpeg	Sky, water, tree, plane, clouds	Sky, water, (tree, people)
 (d) 113046.jpeg	Horses, mare, foals, field, grass	Field, foals, horses, mare, (tree)
 (e) 119036.jpeg	Water, tree, street, people, cars	Buildings, cars, sky, street (sky, tracks, turn, tree, water, people)
 (f) 276019.jpeg	Sky, park, clouds, mountain, tree	Mountain, sky, snow, water, (tree, people, bear, polar)
 (g) 131003.jpeg	Plants, leaf, tree, garden, birds	Fence, flowers, grass, vines, (tree, sky, horses, foals, garden, petals, plants, leaf)

4.5 Evaluating results

We automatically annotate each test image using top 5 words among retrieved images and then simulate image retrieval tasks using all possible (260 queries) one-word queries. We calculate the mean of recalls and precisions for each query.

Results of methods described in the previous subsections are compared to show that proposed method is better among other methods, where low-level and high-level relation has not been used. Count of keyword recalls that are greater than zero are used to measure the performance of each experiment.

Performance results of following methods are presented in the figure 4.3.

- Proposed method with color
- Proposed method with color and texture
- *Alg-blob* with color and texture features
- *Alg-blob* with color features
- *Alg-trad1b* with color features
- *Alg-trad1b* with textures features
- *Alg-trad1a* with color features
- *Alg-trad1a* with textures features

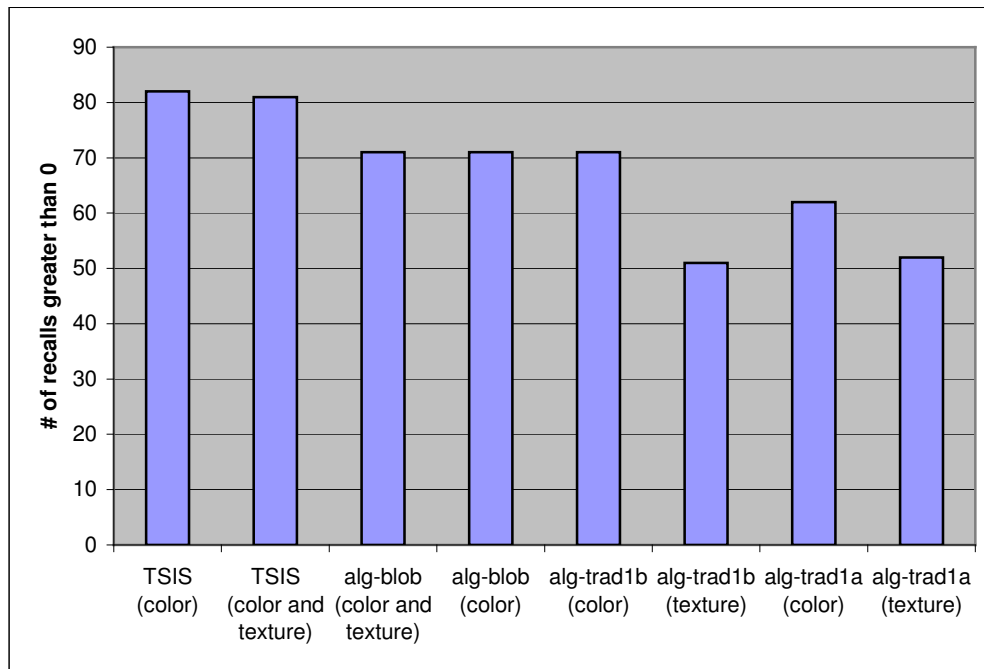


Figure 4.4 Performance comparisons of experiments.

With these experiments it is easy to observe that the proposed method outperforms all other classical methods on the same dataset. Information that can be obtained from these experiments is that texture features are not stronger than color features to describe images. In other words, extracted color features are more descriptor than texture features in the used dataset. Because the performances of queries that are base on color features are better than queries base on texture feature as can be seen in Figure 4.4. When color and texture features are used together, the performance is also lower then when only color features are considered. With these experiments one cannot expect to have better performance when texture features are used in the proposed method.

Similar to the previous studies on automatic image annotation, the quality of automatic image annotation is measured by the performance of retrieving auto-annotated images regarding to single-word queries. For each single-word query, precision and recall are computed using the retrieval lists that are based on the true annotations and the auto-annotations. For all experiments presented, the precision and recall values of each keyword that has recall greater than 0 are given in Appendix A.

Table 4.6 Performance comparison on the task of automatic image annotation on the Corel dataset.

Model	Co-occurrence	Translation	CMRM	MBRM	MixHier	TSIS
#words with recall>0	19	49	66	122	137	82
Single word query results on all 260 words.						
Mean recall	0.02	0.04	0.09	0.25	0.29	0.10
Mean precision	0.03	0.06	0.10	0.24	0.23	0.11

We also compare the annotation performance of the similar models in the literature where they have used the same data set as in our study (Table 4.6). The values of recall and precision were averaged over the set of testing words, as suggested by Carneiro, et. al., 2005 and Feng, et. al., 2004. Table 4.6 presents results (borrowed from Carneiro, et. al., 2005 and Feng et. al., 2004) obtained with various other methods under the same experimental set. In table 4.6, numbers of keywords that have recall values greater 0, and precision recall averages for similar studies are shown. Especially we consider Co-occurrence Model (Mori et. al.), the Translation Model (Duygulu, et. al., 2002), Cross-Media Relevance Models (CMRM) (Jeon et. al.), Multiple-Bernoulli Relevance Model (MBRM) (Feng, et. al., 2004) and Mix-Hier (Carneiro, et. al., 2005). MBRM and Mix-Hier have better performance than the proposed method (TSIS), if we consider the recall values that are positive. On the other hand, complexity is another important issue for the annotation process. In our experiments for the proposed method, the average annotation time was 14 seconds where it is 268 seconds for Mix-Hier. In these studies same data set and same evaluation method has been used, however there are little differences. For example the aim of Translation Method is to auto-annotate image regions instead of entire image. In studies of MBRM and Mix-Hier methods, they extracted low-level features by dividing image into regular grids, instead of using the low-level features provided with the used dataset. So, the used data set is the same but low-level features are different from our study. The detail descriptions of these methods were discussed in chapter two.

In this study image segmentation and feature extraction was out of our scope because they have to be studied within their topic's research area. Our belief is that by using well-defined image segments, low-level features and text annotations the proposed method will yield better performance on image query and auto-annotation.

CHAPTER FIVE

CONCLUSION

5.1 Conclusion

In this study, we presented a new solution to (1) auto-annotate images based on similarity with existing annotated images (2) semantically retrieve images using keywords. Our main hypothesis is that images that fall in the same text cluster can be described with common visual features of those images. The system is highly relies on the overlapping of the similar parts of images in both textually and visually. We have show that our proposal is capable to be used in auto-annotation of images and improve the retrieval effectiveness. The proposed methodology has the following contributions to the field:

- i. It is a novel image retrieval method that attempts to bridge the semantic gap by providing image clusters organized by their annotations, instead of a set of ordered images. The image clusters are obtained from an *unsupervised learning process* that contains both low-level and high-level image information. In this sense, our methodology aims to find textual descriptions (annotations) of given image by considering images' low-level features clustered by their annotations.
- ii. Both in training and querying steps, there is no need to human interventions such as relevance feedback. Also in our experiments we didn't make any manual contribution or annotation to increase the performance.
- iii. It is a cluster-based approach trying to find out of which cluster can contain the query image and of which images in that cluster are most similar to query image. So comparison step considers images by means of their *distance to query vector* and *correlation in the cluster*, instead of exhaustive search as in traditional methods.

- iv. The proposed approach does not have boundaries, and hence can be extended with other techniques such as relevance feedback, to use their benefits. To do so, information obtained from user feedbacks can be used to modify coupling values in the image clusters, thus increase the retrieval effectiveness.
- v. The proposed algorithm does not depend on the specific feature types (colour, texture etc.). It depends on clusters of that features that are blobs to identify the images instead. So, any clustered feature could be used as the descriptors of the images.
- vi. *C3M* originally developed for text clustering and it relies on the cover coefficient concept, which indicates relationships among text documents. The presented novel approach shows that, *C3M* can also be used for querying images by considering relationships among images (cover coefficient of images).
- vii. In this method more than one low-level feature can be used as the descriptor for any image, which is one of the biggest issues in traditional image retrieval. This is not the same property as property in the previous paragraph, because we didn't use annotation information in the image vector, we use it for the first step in the training that helps to cluster images instead. What we mean here is once the blobs of different features are created, then they can be assign to images with weights easily.
- viii. The proposed method generates ranked images after query process, so best matching image is retrieved at the first place.
- ix. The retrieval performance of the system is evaluated by considering test images and their annotations. So the performance relies on the manual annotations of test image and missed annotations in the test set will decrease the retrieval effectiveness. We solve this problem by modifying the manual annotations of test images by applying a thesaurus list that is specific to the considered data

set. We believe that this approach will help the researchers in the same area while they are evaluating their system's abilities.

- x. There are also some limitations for proposed method, which the performance depends on extracted regions, feature extraction, parameters and selected clustering algorithms. The performance changes where those issues change.

The proposed system was trained with a dataset containing 4500 images from COREL image database and tested with 500 images from outside the training database. Benchmarking experiments have demonstrated that good accuracy of proposal and its high potential use in auto-annotation of images and for improvement of content-based image retrieval. The performance results of proposed method are compared with similar studies in the literature where they used the same dataset. Experiments have demonstrated that good accuracy of proposal and its high potential of use in annotation of images and for improvement of content based image retrieval.

5.2 Future Directions

The work presented in this thesis should be considered as suggesting a novel strategy that fills the semantic gap. Therefore, there remain number of open issues that each requires an individual research.

For the further studies;

- Low-level Image descriptors were obtained from a dataset that is provided by Duygulu, et. al. (2002) to compare the proposed method with other similar studies. Image features are directly affects the system performance. Studies on image segmentation and automatic feature extraction are remains as open research.
- Keywords are clustered with C3M and low-level features are clustered with k-means clustering algorithm. It is likely that better clustering will increase

the system performance. Clustering methods used in the proposed method could be replaced with other clustering algorithms such as expectation maximization, agglomerative clustering and etc., and hence need to be investigated.

- As we told in the previous subsection, the proposed approach can be extended with many other image retrieval techniques such as, relevance feedback and semi-automatic image clustering approaches, thus they need to be investigated.
- The proposed solution can lead to new researches including semantic web, semantic indexing, and development of image ontology automatically and extend to video.

As we discussed, there are many open research issues that are not covered in this thesis and they need to be investigated.

REFERENCES

- Assfalg, J., Bertini, M., Colombo, C. & Bimbo, A. D (2002): Semantic Annotation of Sports Videos. *IEEE Multimedia*. Vol:9.2, pp:52-60.
- Bach, J., Fuller, C., Gupta, A., Hampapur, A., Gorowitz, B., et. al. (1996). Virage image search engine: an open framework for image management. *Proceedings of the SPIE, Storage and Retrieval for Image and Video Databases IV*. San Jose, CA, pp. 76–87.
- Berry, M.W., Dumais, S. T., & O’ Brien, G. W. (1995). Using linear algebra for Intelligent Information Retrieval. *SIAM Review* 37:4, pp. 573-595.
- Bronson, R. (1989). *Matrix Operations, Schaum’s Outlines*. McGraw-Hill. Chapter 21.
- Brown, P., Pietra, S. D., Pietra, V. D., & Mercer, R. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*. 19(2), pp. 263-311.
- Can, F., & Ozkarahan. E. A. (1990). Concepts and Effectiveness of the Cover Coefficient Based Clustering Methodology for Text Databases. *ACM Transactions on Database Systems*, Vol. 15, No. 4.
- Can, F. (1993). Incremental Information Clustering for Dynamic Processing. *ACM Transactions on Information Systems*. Vol. 11, No. 2, pp: 143-164.
- Carneiro, G., & Vasconcelos N. (2005). Formulating Semantic Image Annotation as a Supervised Learning Problem. *IEEE CVPR 2005*.

- Carson, C., Thomas, M., Belongie, S., Hellerstein, J. M. & Malik, J. (1999) Blobworld: A system for region-based image indexing and retrieval. *Third International Conference on Visual Information Systems*.
- Celebi, E. & Alpkocak, A. (2000). Clustering of Texture Features for Content Based Image Retrieval. *Lecture Notes in Computer Sciences Springer-Verlag*. Vol:1909, ISSN:0302-9743, pp.216-225.
- Deb, S. & Zhang, Y. (2004). An overview of Content Based Image Retrieval Techniques. *18th Int. Conference on Advanced Information Networking and Application (AINA'04)*.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*.
- Duygulu, P., Barnard, K., Freitas, J.F.G. & Forsyth, D. A. (2002): Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. *European Conference on Computer Vision (ECCV2002)*.
- Duda, R. O., Hart, P. E. & Stork, D. G. (2001). *Pattern classification*. John Wiley and Sons Inc.
- Duygulu, P. (2003). *Translating images to words: A Novel Approach For Object Recognition, PhD Thesis*. Middle East Technical University.
- Feng, S.L., Manmatha, R., & Lavrenko, V. (2004) Multiple Bernoulli relevance models for image and video annotation. *IEEE CVPR 2004*.
- Feder, J., (1996). Towards image content-based retrieval for the World-Wide Web. *Advanced Imaging*. 11(1), pp. 26-29.

- Hauptmann, A., & Christel, M. (2004). Successful Approaches in the TREC Video Retrieval Evaluations. *Proc. ACM Multimedia'04*. New York, pp. 668-675.
- Jeon, J., Lavrenko, V., & Manmatha, R. (2003). Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. *ACM SIGIR'03*.
- Jones, K. S., & Willet, P. (1997). *Readings in Information Retrieval*. Morgan Kaufmann, ISBN 1-55860-454-4.
- Landauer, T. K., & Littman, M. L. (1990). Fully automatic cross-language document retrieval using latent semantic indexing. *Proceedings of the 6th Annual Conference of the UW*. Centre for the New Oxford English Dictionary and Text Research, pp. 31-38.
- Leonardi, R. & Migliorati, P. (2002). Semantic Indexing of Multimedia Documents. *IEEE MultiMedia*. Vol.9.2, pp:44-51.
- Maedche, A., & Staab, S. (2004). *Ontology Learning*. Handbook of Ontologies in Information Systems. Springer Verlag. pp: 173-190.
- Maddii, G. P., Velvadapu, C. S., Srivastava, S. & Lamadrid, J. G. (2001). Ontology Extraction from Text Documents by Singular Value Decomposition. *ADMI'01*.
- Mojsilovic, A., & Jose, G. (1999). Semantic Based Categorization and retrieval in medical image databases. *ACM SIGMOD Record*. Vol 28.
- Monay, F., & Perez, G. (2003). On Image Auto-Annotation with Latent Space Model. *ACM Multimedia'03*.
- Mori, Y., Takahashi, H. & Oka, R. (1999). Image-to-word transformation based on dividing and vector quantizing images with words. *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*.

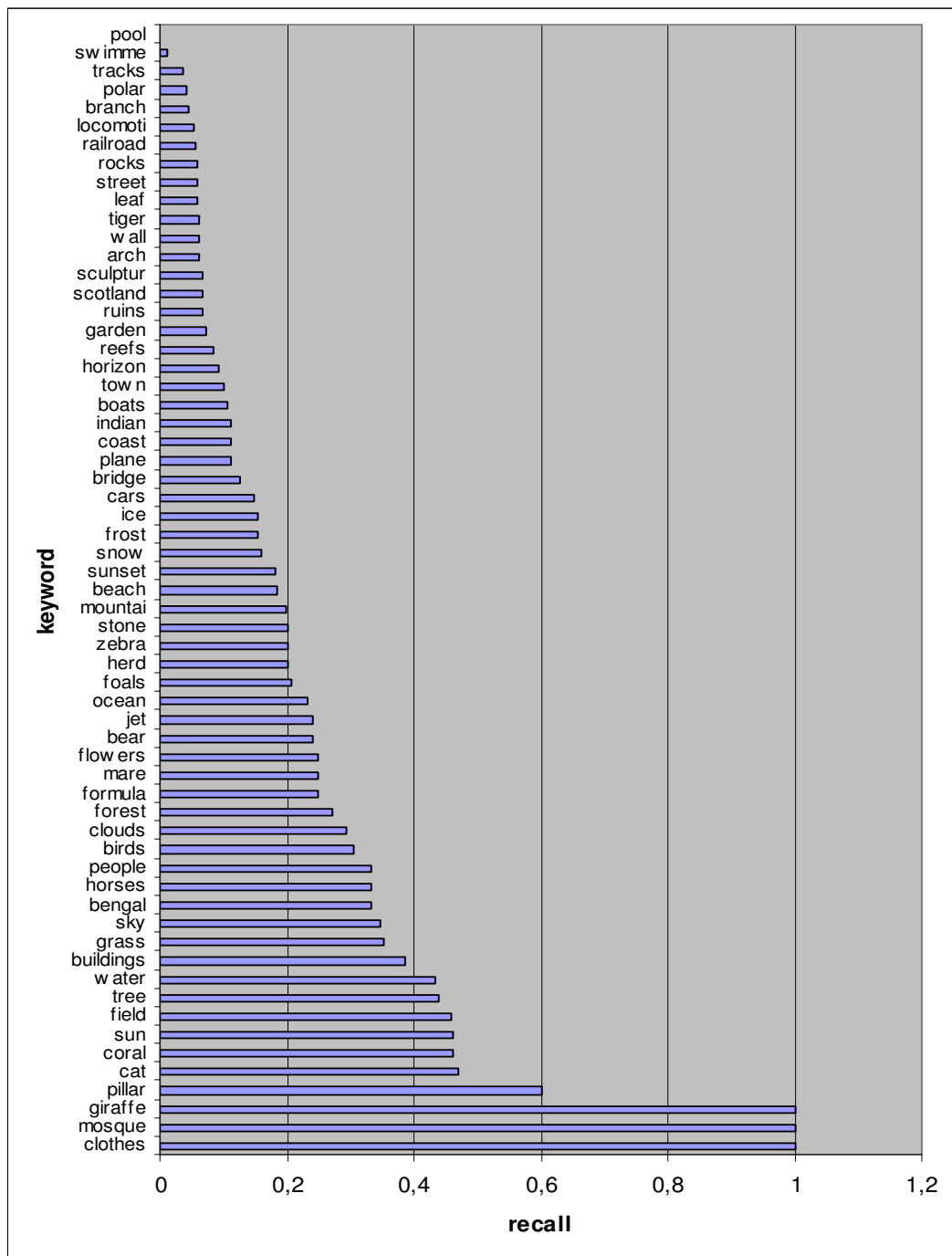
- Mills, T. J., Pye, D., Sinclair, D. & Wood, K. R. (2000). Shoebox: A digital photo management system. *Technical Report 2000.10*, AT&T Research.
- Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., et. al. (1993). The QBIC project: Querying images by content using color, texture, and shape. *Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases 2-3'93*. pp. 173–187.
- Niblack, W., Hafner, J. L., Breuel, T., & Ponceleon, D. (1998). Updates to the QBIC system. *Storage and Retrieval for Image and Video Databases VI Proc SPIE*. vol. 3312, pp. 150-161.
- Pantel, P. & Lin, D. (2002). Discovering Word Senses from Text. ACM Special Interest Group on Knowledge Discovery in Data and Data Mining. ISBN:1-58113-567-X, pp. 613-619.
- Rui, Y., Huang, T. S., & Chang, S. F. (1998) Image Retrieval: Past, Present, and Future. *Journal of Visual Communication and Image Representation*'98.
- Salton, G., & Wong, A., (1978). Generation and Search of Clustered files. *ACM Trans Database Systems*. 3(4), pp: 321-346.
- Sheikholeslami, G. & Chang, W., & Zhang, A., (2002) SemQuery: Semantic Clustering and Querying on Heterogeneous Features for Visual Data. *IEEE Trans Knowledge and Data Engineering*. 14(5), pp. 988-1002.
- Shi, J., & Malik, J., (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp: 888-905.

- Smith, J. R. & Chang, S. F. (1996). Querying by Color regions using the VisualSeek Content-Based Visual Query System. *Intelligent Multimedia Information Retrieval*. IJCAI'96.
- Smith, J. R. (1997). *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis*. PhD. thesis, Graduate School of Arts and Sciences, Columbia University.
- Smeulders, W. M., Worring, M., Santini, S., Gupta, A. & Jain, R.(2000): Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 12.
- Wang, J. Z., Li, J., Wiederhold, G. (2001). SIMPLicity, Semantics-Sensitive Integrated Matching for Picture Libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. vol. 23(9).
- Young, P. G. (1994). *Cross-Language Information Retrieval Using Latent Semantic Indexing*. Master's thesis The University of Knoxville, Tennessee.

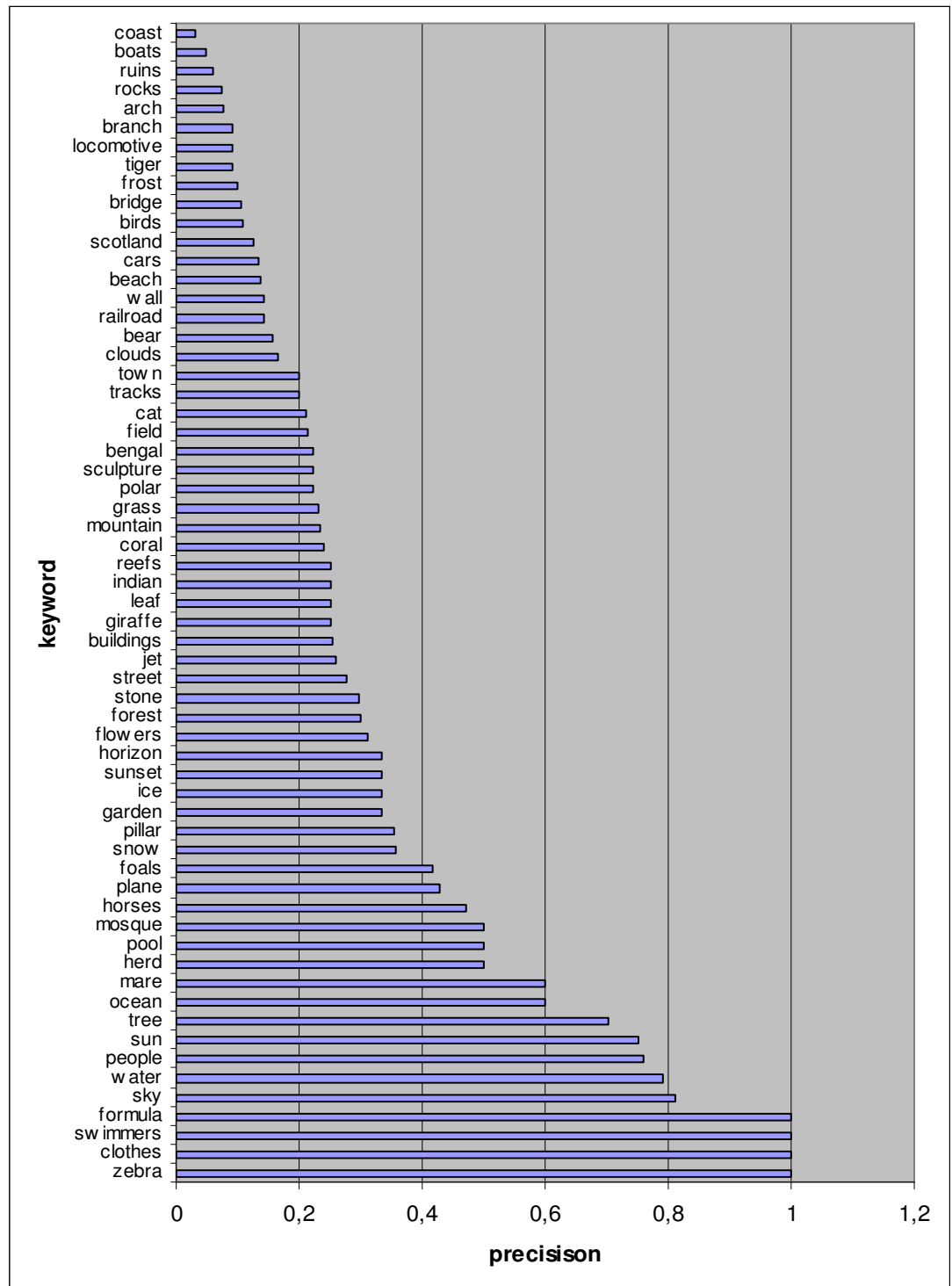
APPENDIX A

A.1 Traditional Image Retrieval and annotation.

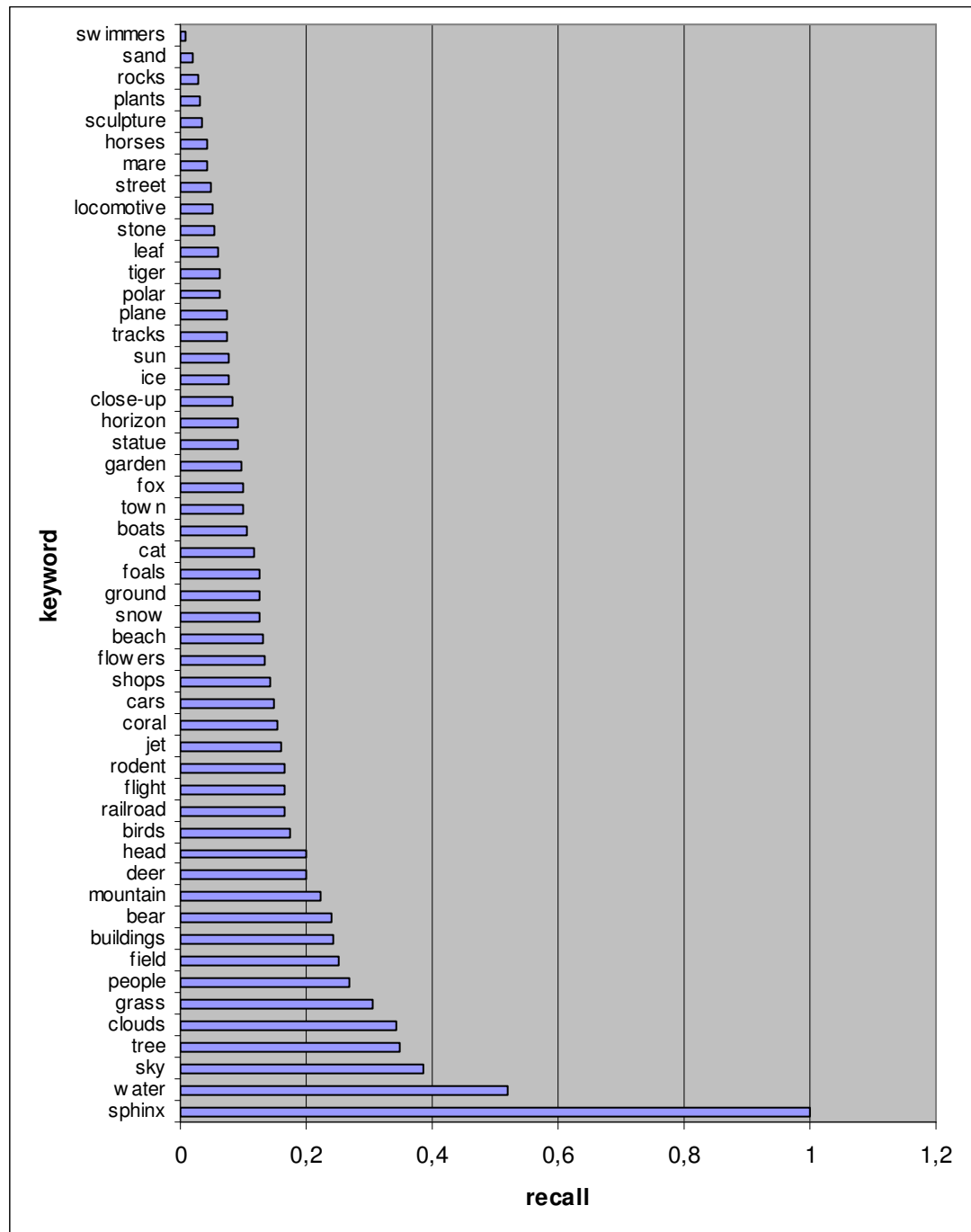
A.1.1 Recall graph when only color features are considered with traditional retrieval (*alg-trad1a*) where 5 keywords are used for auto-annotation task.



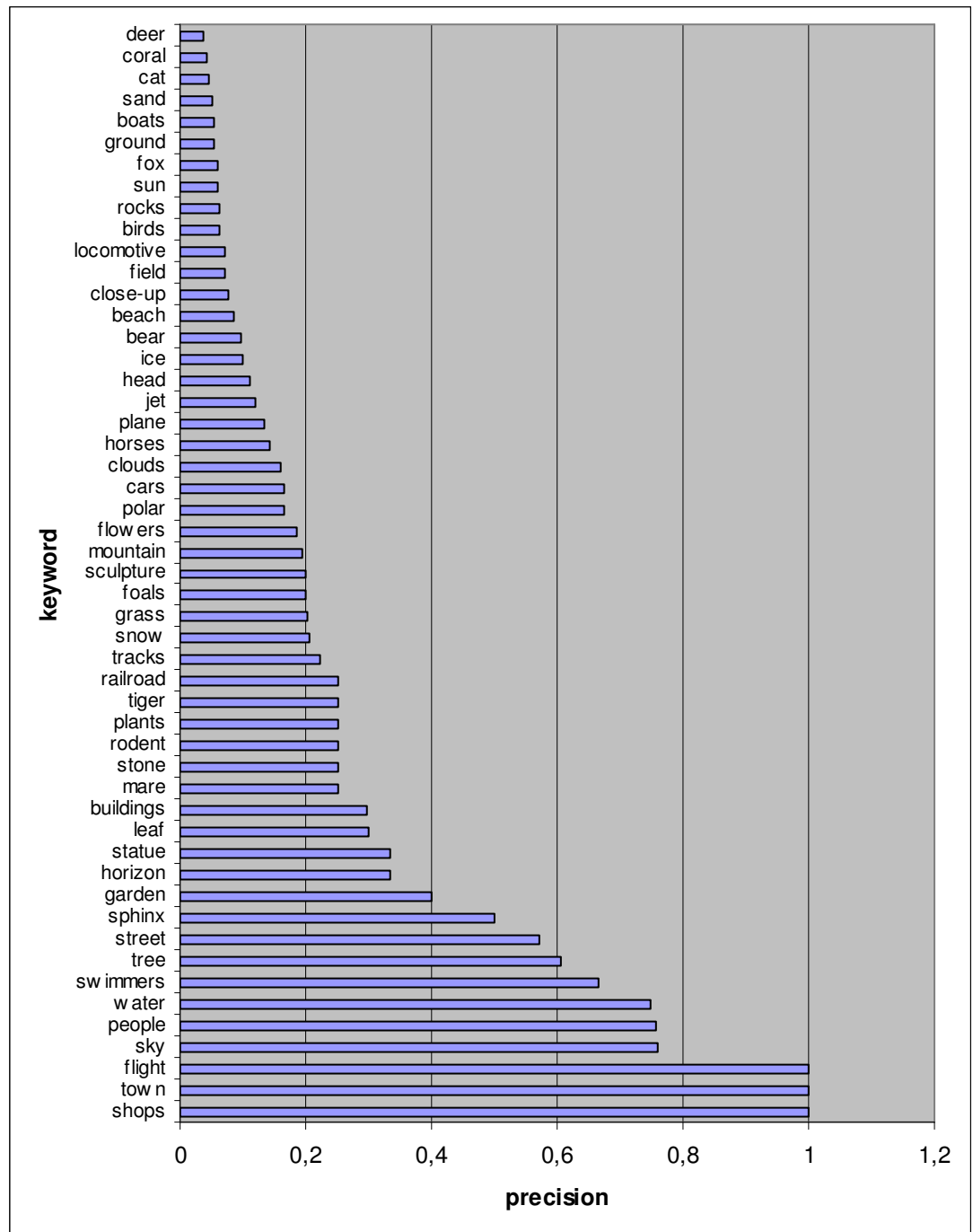
A.1.2 Precision graph when only color features are considered with traditional retrieval (*alg-trad1a*) where 5 keywords are used for auto-annotation task.



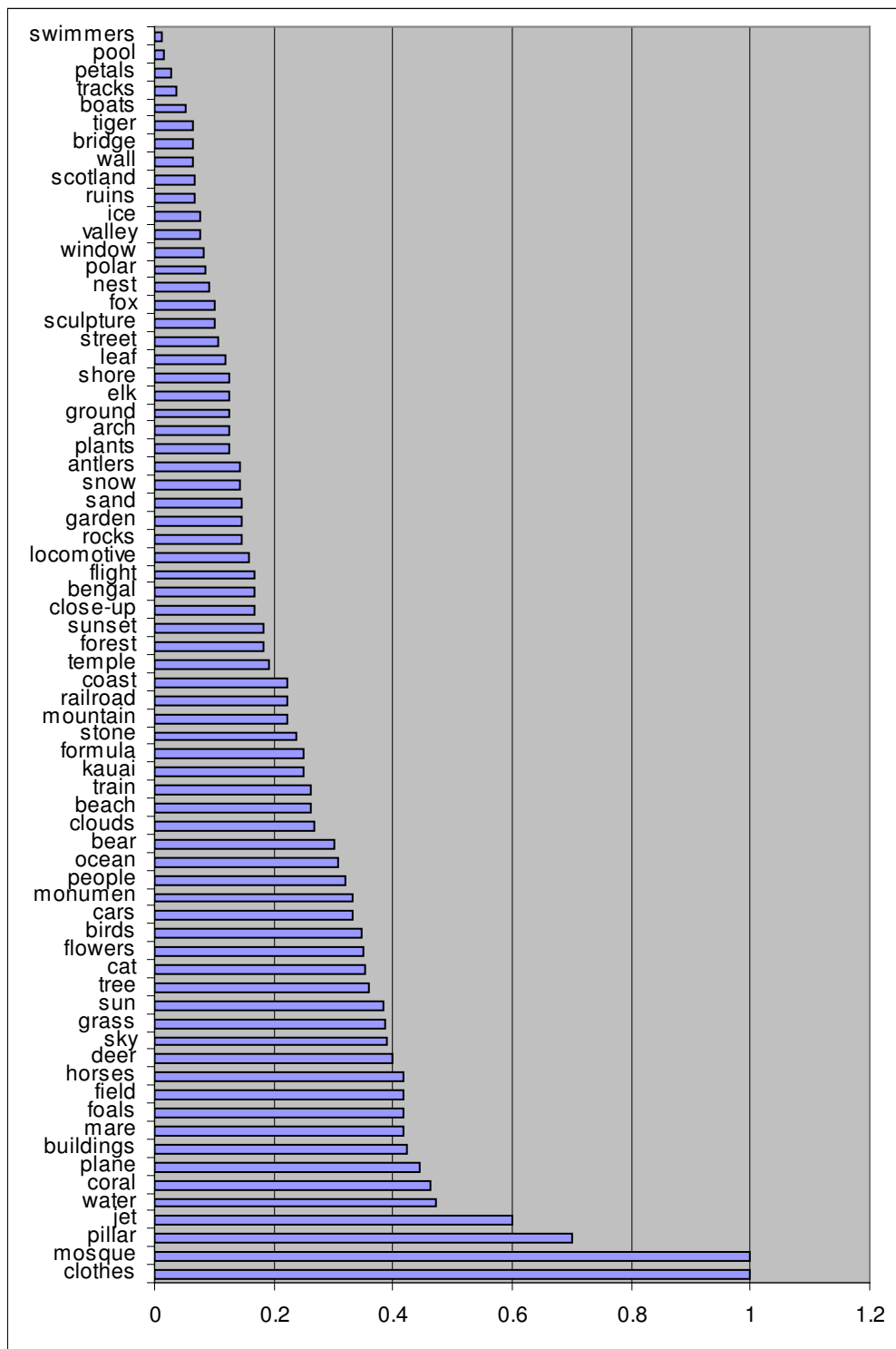
A.1.3 Recall graph when only texture features are considered with traditional retrieval (*alg-trad1a*) where 5 keywords are used for auto-annotation task.



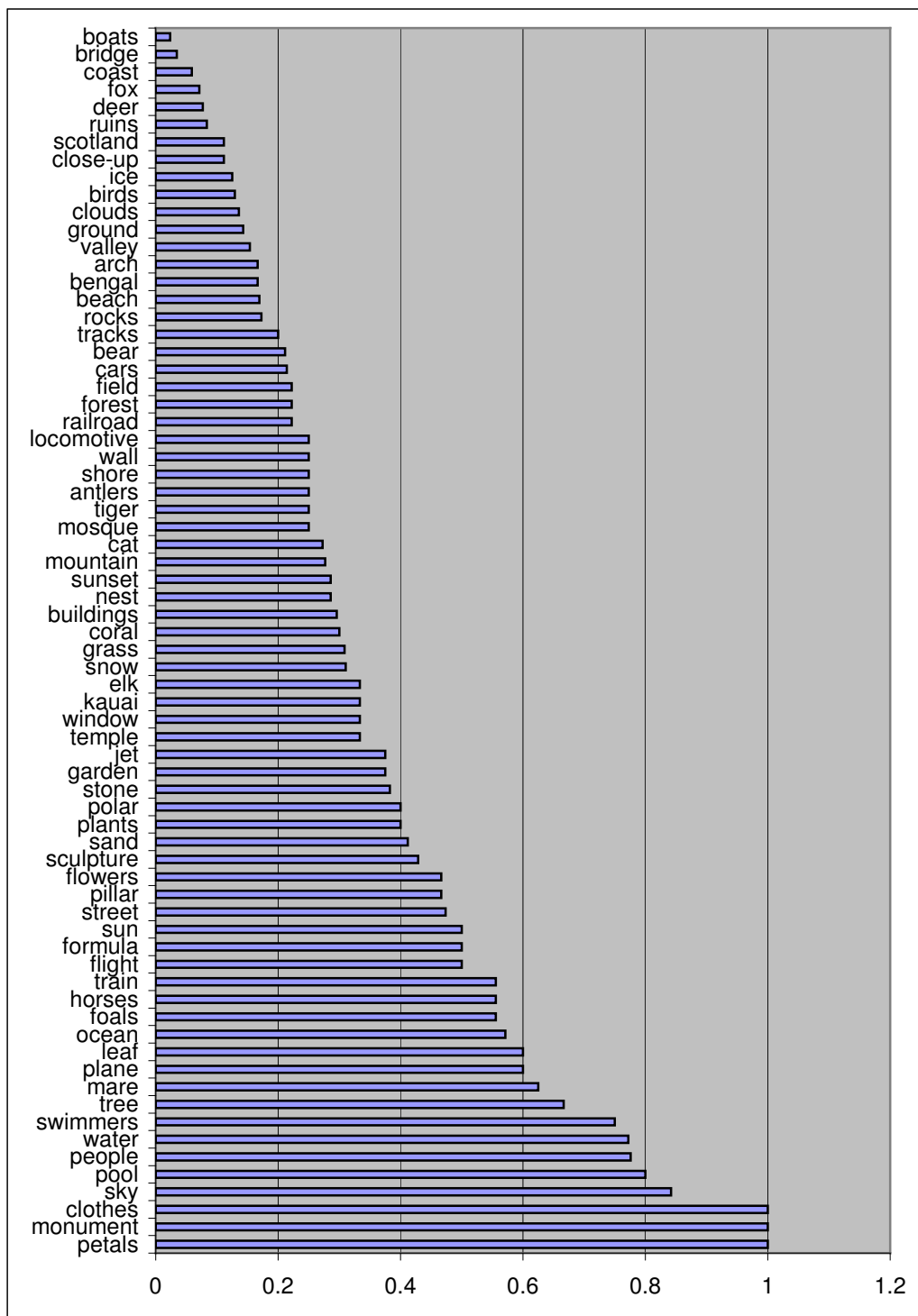
A.1.4 Precision graph when only texture features are considered with traditional retrieval (*alg-trad1a*) where 5 keywords are used for auto-annotation task



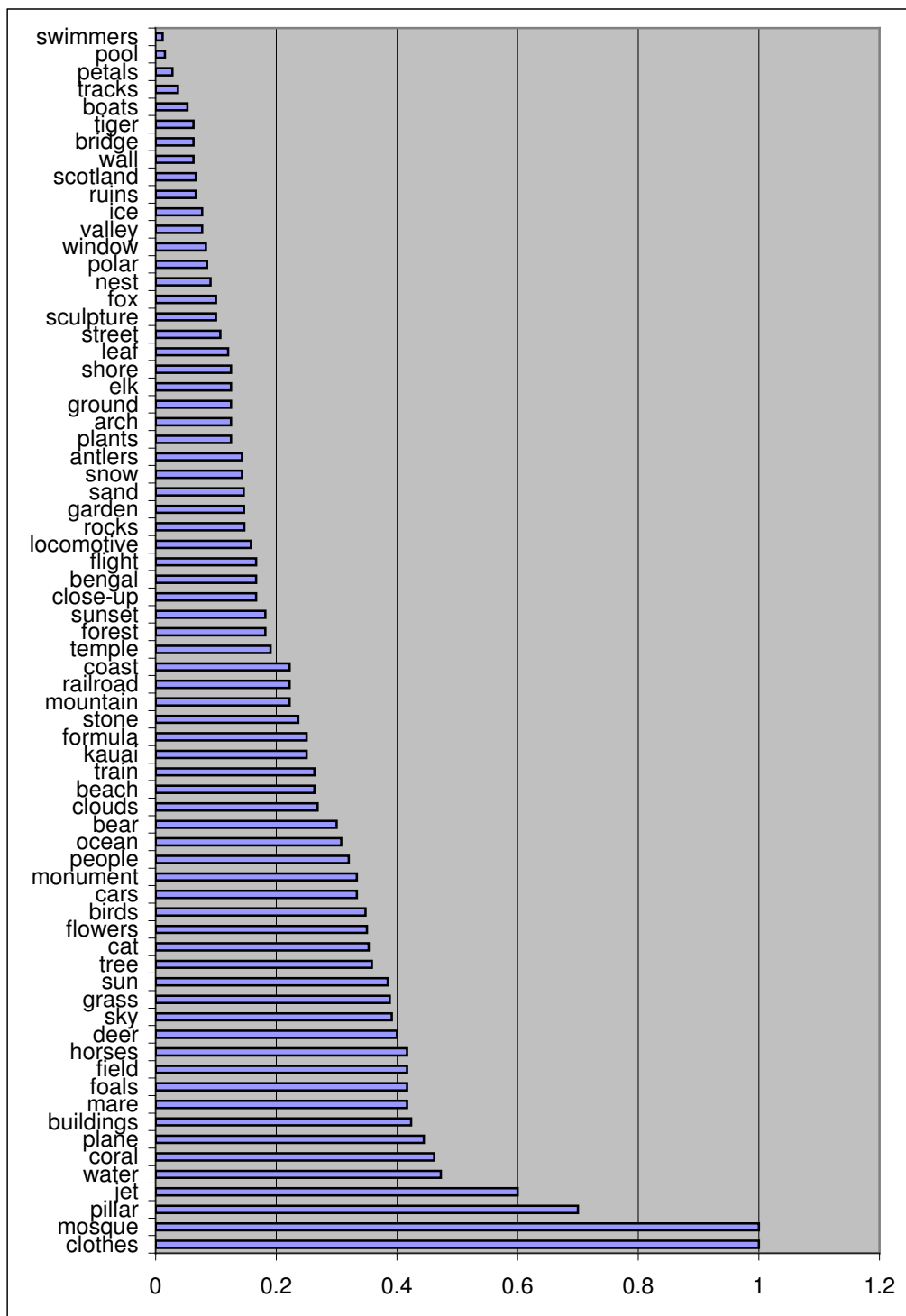
A.1.5 Recall graph when only color features are considered with traditional retrieval (*alg-trad1b*) where 5 keywords are used for auto-annotation task.



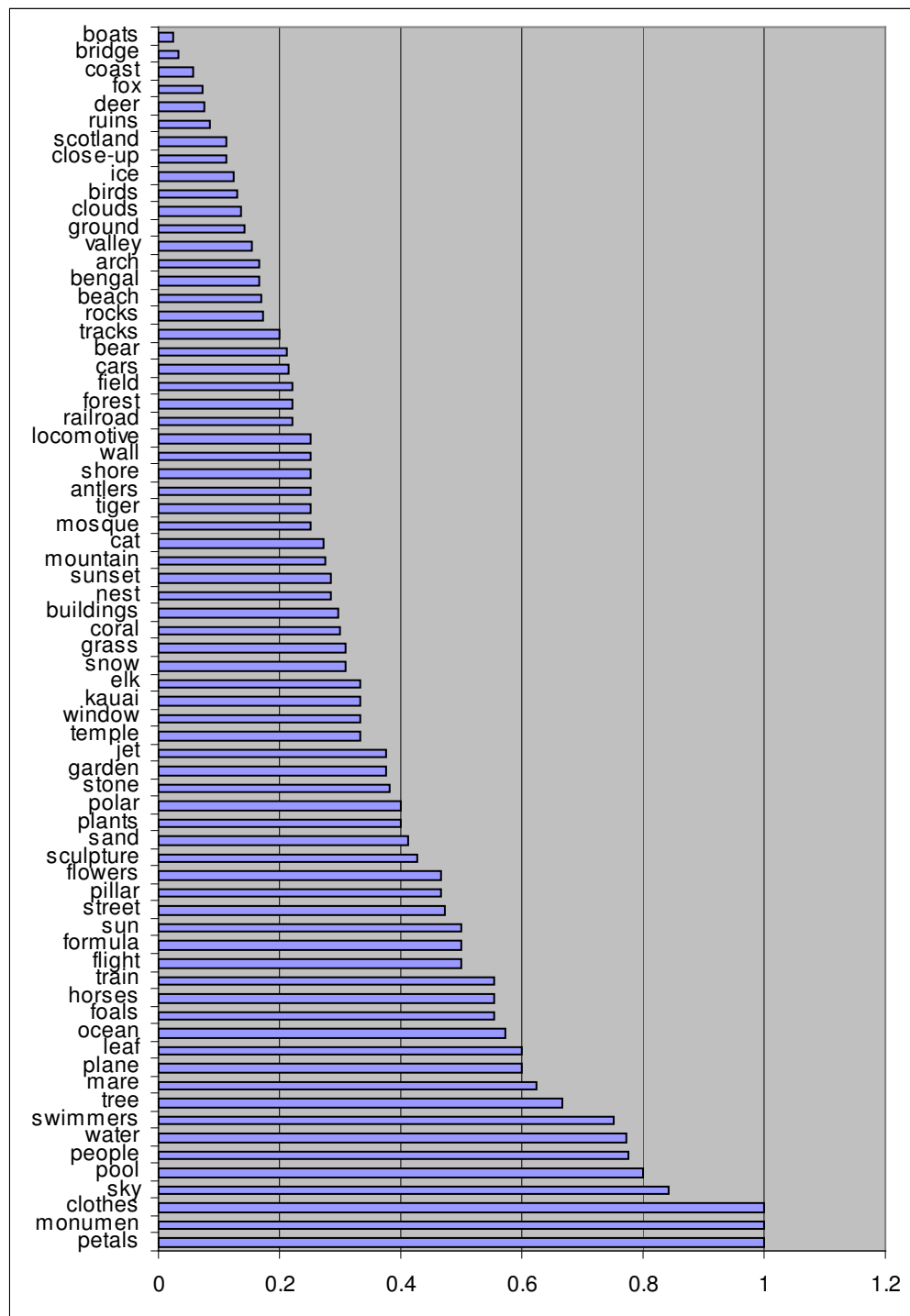
A.1.6 Precision graph when only color features are considered with traditional retrieval (*alg-trad1b*) where 5 keywords are used for auto-annotation task.



A.1.7 Recall graph when only texture features are considered with traditional retrieval (*alg-trad1b*) where 5 keywords are used for auto-annotation task

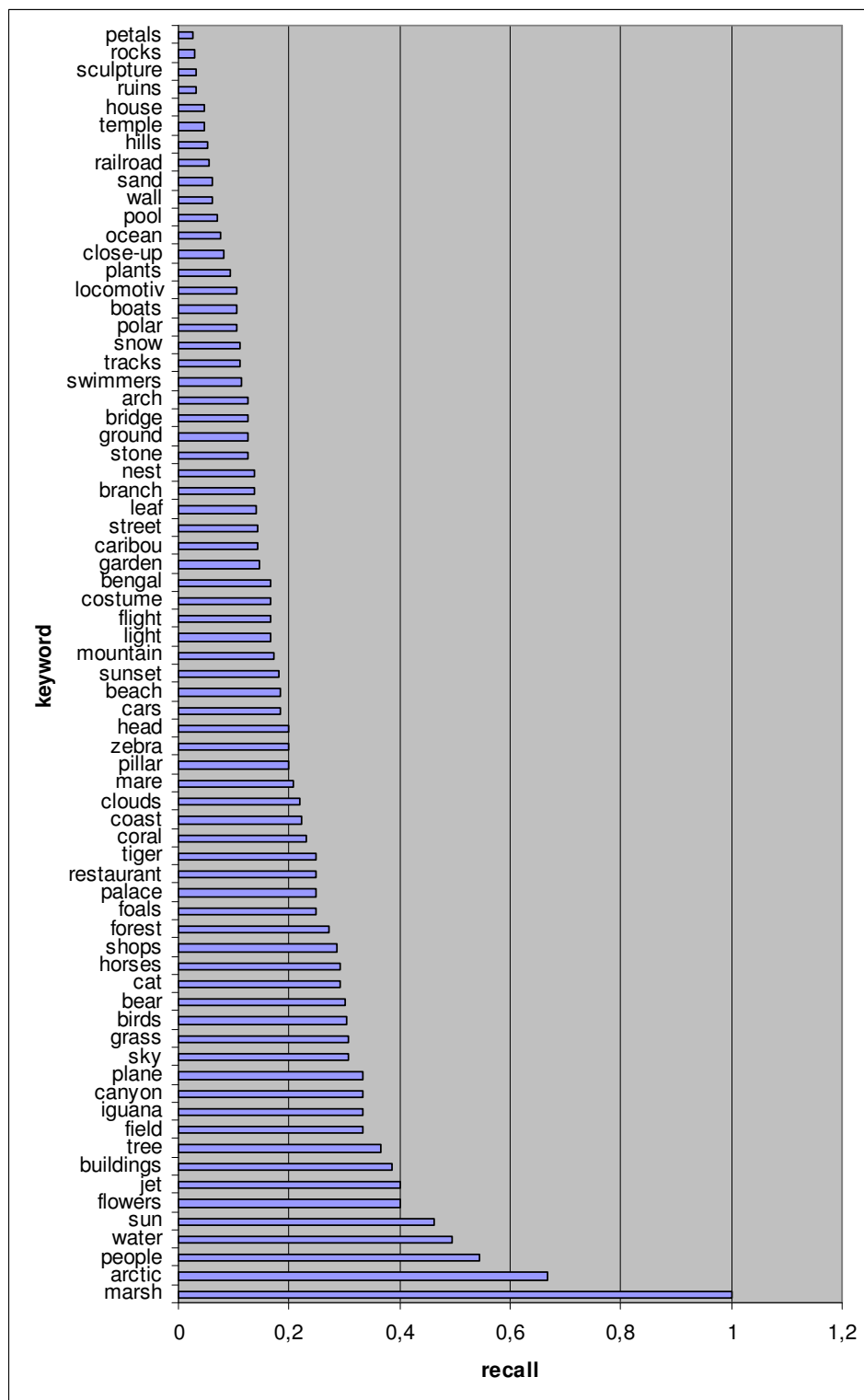


A.1.8 Precision graph when only texture features are considered with traditional retrieval (*alg-trad1b*) where 5 keywords are used for auto-annotation task

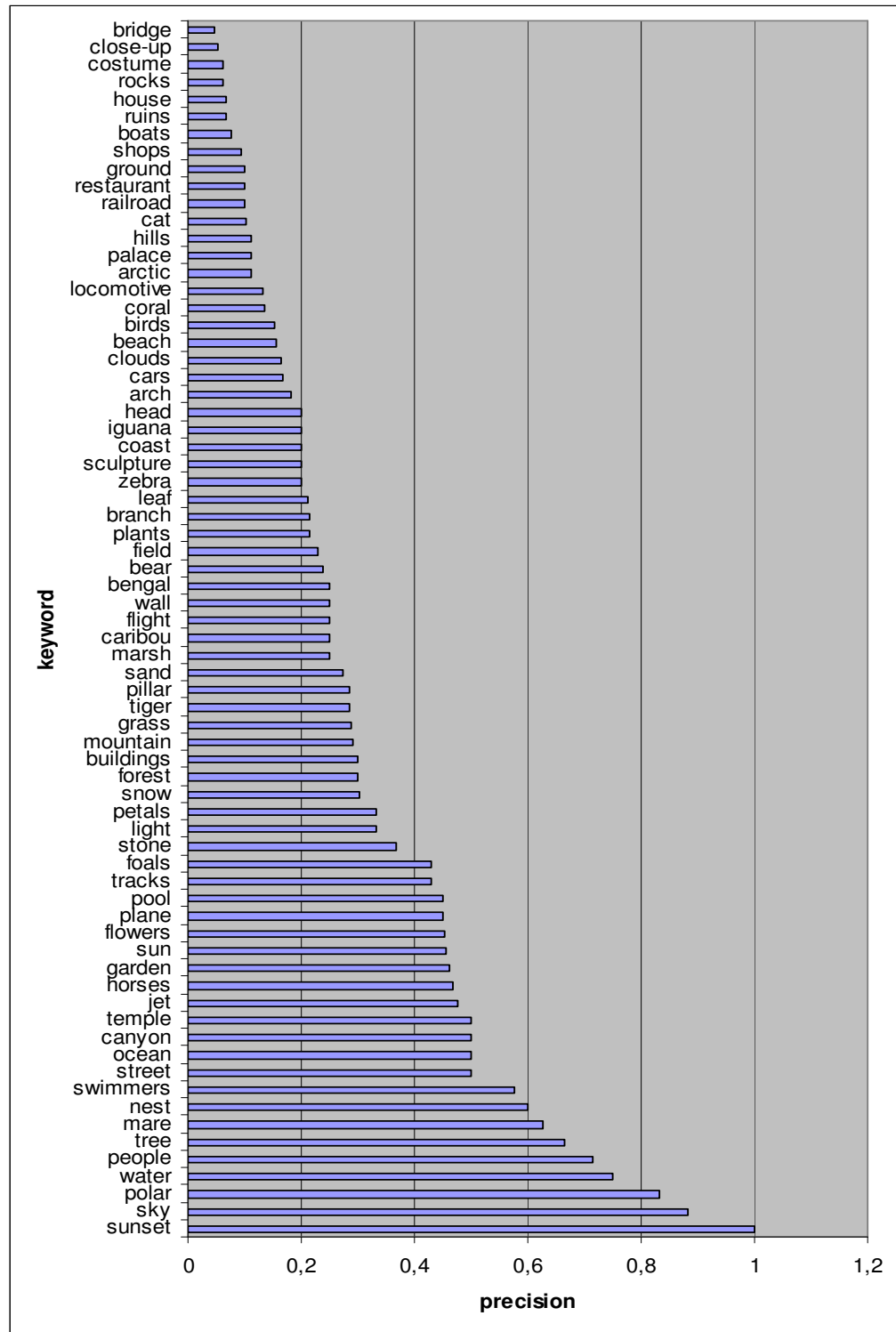


A.2 Blob Based Image Retrieval and Auto-annotation

A.2.1 Recall graph when color and texture blobs are considered and 5 keywords are used for auto-annotation task

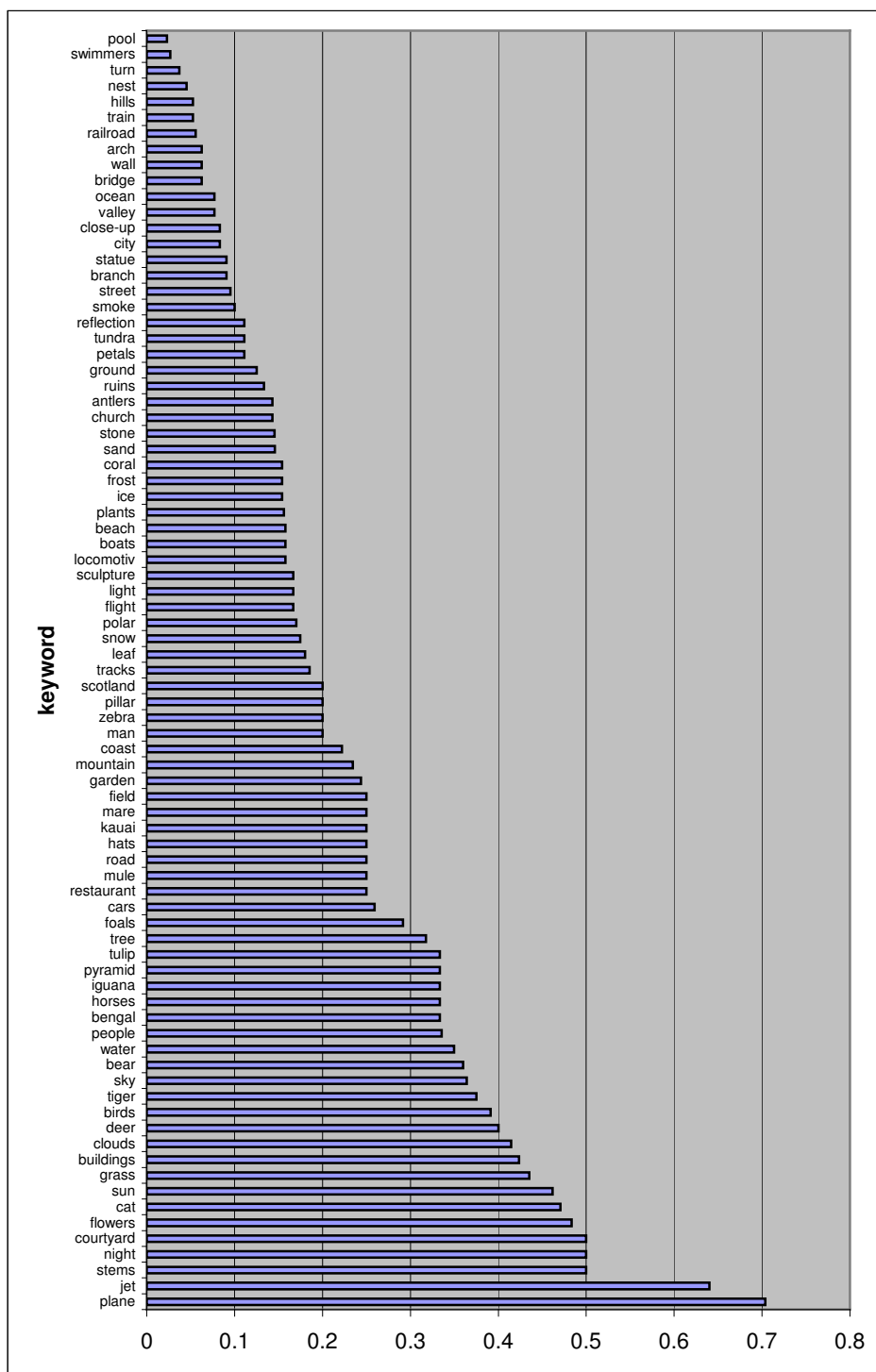


A.2.2 Precision graph when color and texture blobs are considered and 5 keywords are used for auto-annotation task.

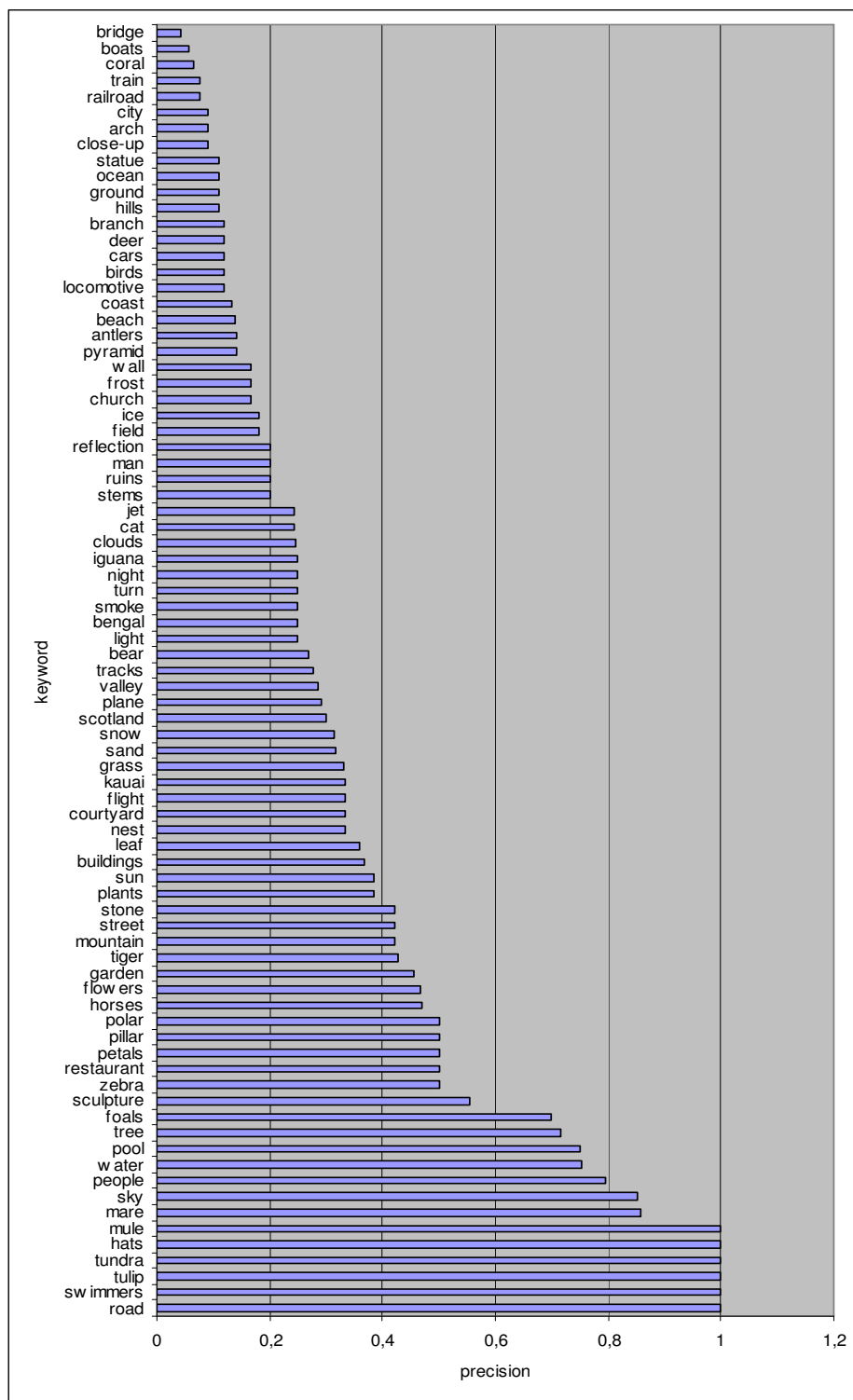


A.3 Image Retrieval and Auto-annotation by Combining High-level and Low Level Features.

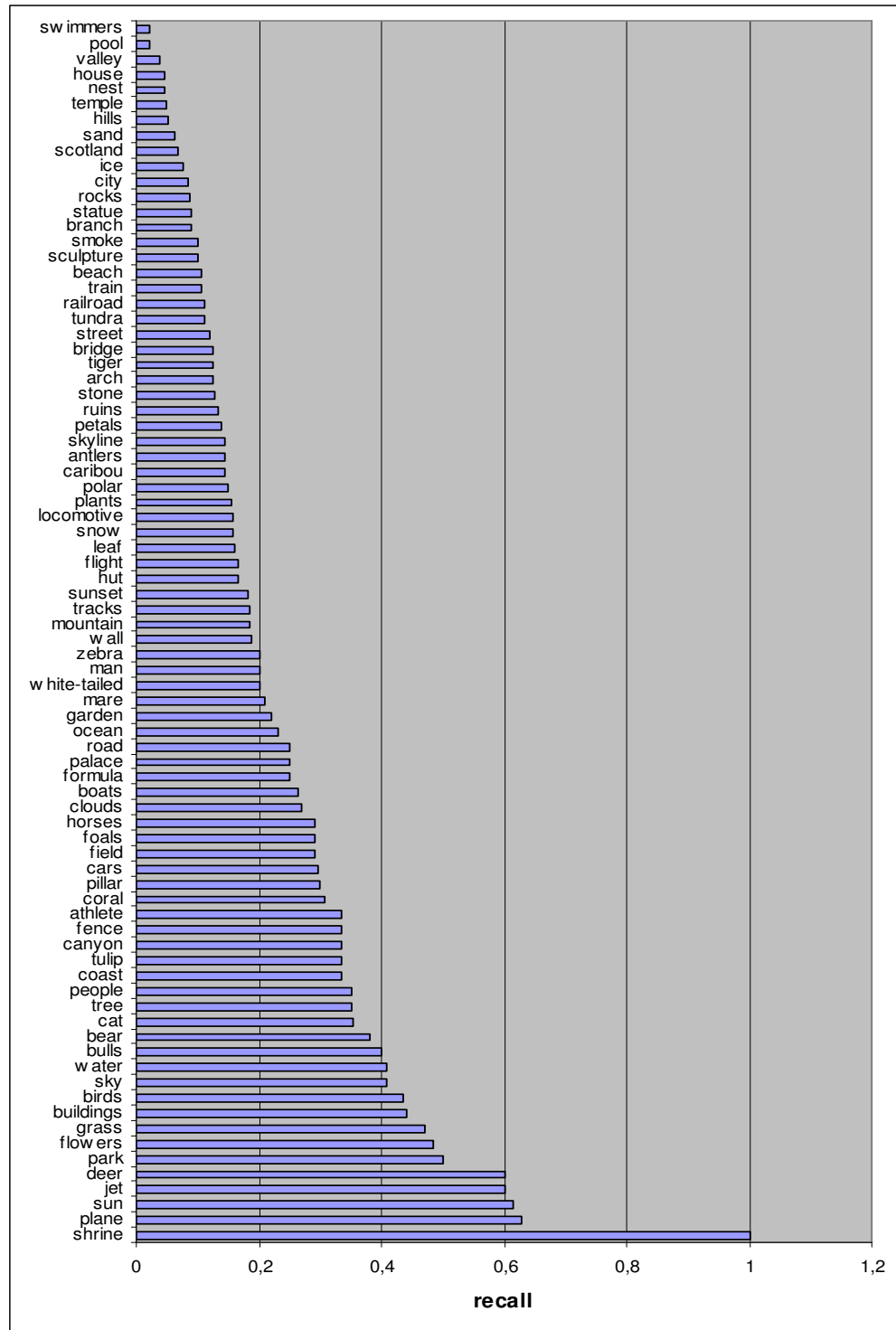
A.3.1 Recall graph for proposed method when only color features are considered and 5 keywords are used for auto-annotation task.



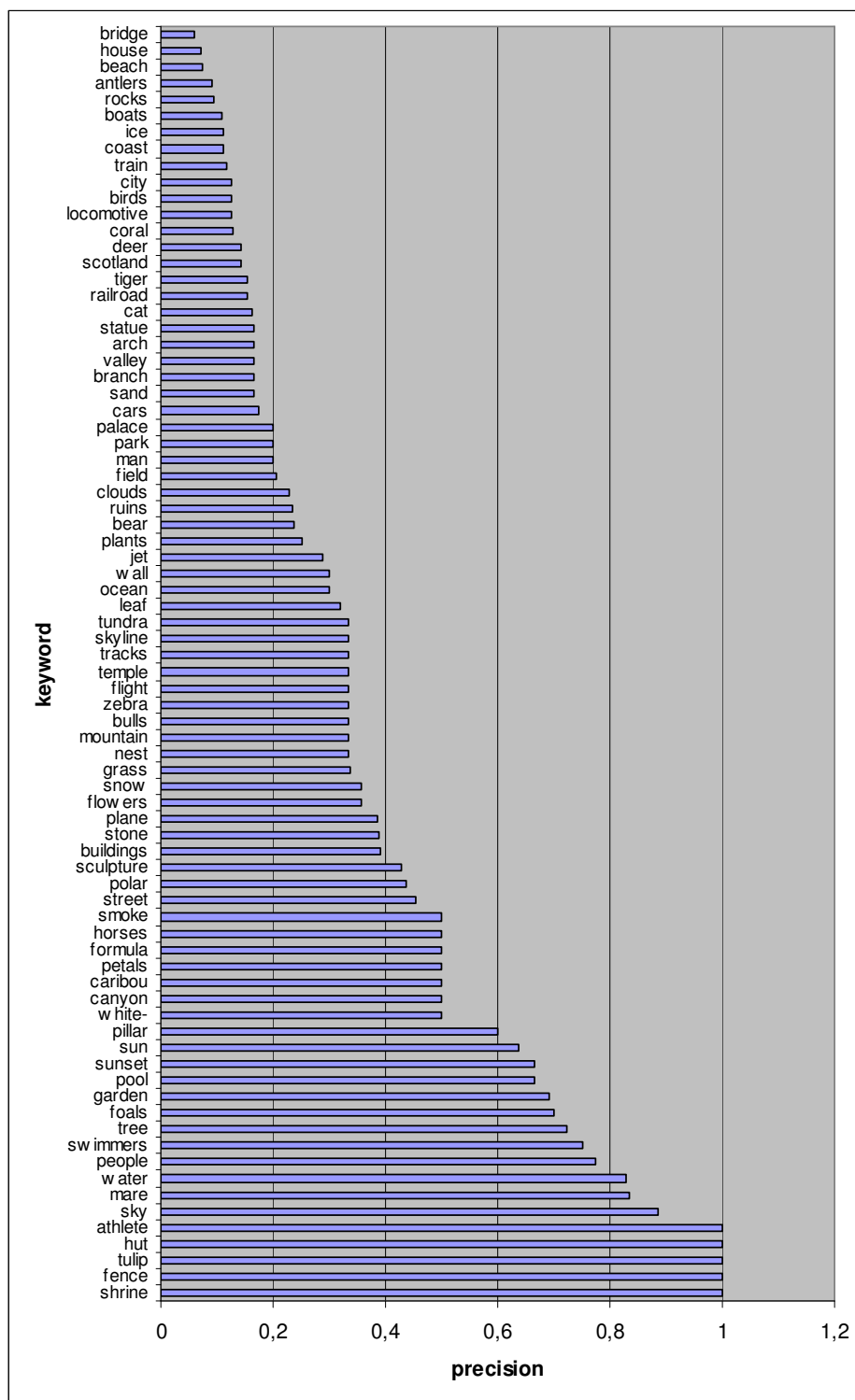
A.3.2 Precision graph for proposed method when only color features are considered and 5 keywords are used for auto-annotation task.



A.3.3 Recall graph for proposed method when both color and texture features are considered and 5 keywords are used for auto-annotation task.



A.3.4 Precision graph for proposed method when both color and texture features are considered and 5 keywords are used for auto-annotation task.



APPENDIX B

B.1 Two level Thesaurus List for Corel Data Set

id	Keyword	Thesaurus-1	Thesaurus-2	id	Keyword	Thesaurus-1	Thesaurus-2
0	city	buildings	sky	187	church	sky	people
1	mountain	sky	tree	188	park	sky	tree
2	sky	tree	water	189	barn	tree	field
3	sun	clouds	sky	190	arch	bridge	stone
4	water	sky	people	191	hats	people	man
5	clouds	sky	mountain	192	cathedral	sky	buildings
6	tree	sky	water	193	ceremony	people	church
7	bay	sun	clouds	194	crowd	people	wall
8	lake	water	sun	195	glass	frost	ice
9	sea	sun	ocean	196	shrine	buddha	people
10	beach	water	sand	197	model	hotel	display
11	boats	water	sky	198	pillar	stone	temple
12	people	water	swimmers	199	carpet	hotel	flowers
13	branch	birds	tree	200	monument	tree	sky
14	leaf	plants	flowers	201	floor	room	pottery
15	grass	tree	sky	202	vines	plants	leaf
16	plain	snow	coyote	203	cottage	tree	grass
17	palm	tree	beach	204	poppies	flowers	grass
18	horizon	water	sunset	205	lawn	garden	tree
19	shell	crab	plants	206	tower	sky	tree
20	hills	water	sky	207	vegetables	garden	tree
21	waves	water	coast	208	bench	garden	flowers
22	birds	nest	tree	209	rose	flowers	leaf
23	land	sun	sky	210	tulip	flowers	petals
24	dog	grass	plane	211	canal	water	boats
25	bridge	water	arch	212	cheese	people	street
26	ships	water	sky	213	railing	tower	sky
27	buildings	sky	street	214	dock	boats	water
28	fence	horses	foals	215	horses	foals	mare
29	island	water	sky	216	petals	flowers	leaf
30	storm	coast	snow	217	umbrella	people	sand
31	peaks	sky	mountain	218	column	stone	ruins
32	jet	plane	sky	219	waterfalls	water	tree
33	plane	jet	sky	220	elephant	sky	water
34	runway	plane	jet	221	monks	people	temple
35	basket	sponges	shops	222	pattern	sand	rocks
36	flight	birds	sky	223	interior	plants	people
37	flag	sky	buildings	224	vendor	people	umbrella
38	helicopter	sky	jeep	225	silhouette	sky	sunset

39	boeing	runway	plane	226	architecture	buildings	sky
40	prop	plane	sky	227	blossoms	flowers	tree
41	f-16	plane	jet	228	athlete	people	swimmers
42	tails	fox	snow	229	parade	people	street
43	smoke	train	railroad	230	ladder	buildings	ships
44	formation	sky	plane	231	sidewalk	people	tree
45	bear	polar	snow	232	store	shops	town
46	polar	bear	snow	233	steps	people	stone
47	snow	bear	polar	234	relief	ruins	sculpture
48	tundra	polar	bear	235	fog	tree	sky
49	ice	frost	snow	236	frost	ice	tree
50	head	close-up	snow	237	frozen	ice	frost
51	black	bear	water	238	rapids	frost	ice
52	reflection	water	buildings	239	crystals	frost	ice
53	ground	grass	water	240	spider	frost	ice
54	forest	cat	tree	241	needles	cactus	blooms
55	fall	tree	water	242	stick	branch	ice
56	river	water	scotland	243	mist	waterfalls	valley
57	field	tree	horses	244	doorway	buildings	people
58	flowers	garden	petals	245	vineyard	field	row
59	stream	water	bear	246	pottery	pots	floor
60	meadow	bear	grass	247	pots	leaf	plants
61	rocks	water	sky	248	military	tree	sky
62	hillside	grass	sky	249	designs	wall	buildings
63	shrubs	desert	sand	250	mushrooms	room	light
64	close-up	leaf	plants	251	terrace	mountain	clouds
65	grizzly	bear	water	252	tent	people	beach
66	cubs	bear	polar	253	bulls	elk	field
67	drum	people	log	254	giant	tortoise	rocks
68	log	reptile	tree	255	tortoise	giant	rocks
69	hut	tree	house	256	wings	birds	albatross
70	sunset	water	horizon	257	albatross	birds	grass
71	display	shops	street	258	booby	birds	flight
72	plants	leaf	flowers	259	nest	birds	branch
73	pool	people	swimmers	260	hawk	birds	branch
74	coral	ocean	reefs	261	iguana	marine	lizard
75	fan	sea	coral	262	lizard	marine	iguana
76	anemone	ocean	coral	263	marine	iguana	lizard
77	fish	ocean	coral	264	penguin	rocks	birds
78	ocean	coral	reefs	265	deer	white-tailed	mule
79	diver	ocean	people	266	white-tailed	deer	tree
80	sunrise	horizon	sun	267	horns	deer	white-tailed
81	face	polar	bear	268	slope	rocks	rodent
82	sand	water	beach	269	mule	deer	snow
83	rainbow	water	waterfalls	270	fawn	deer	grass
84	farms	tree	field	271	antlers	caribou	grass

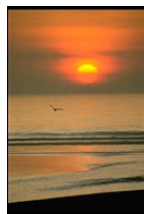
85	reefs	coral	ocean	272	elk	tree	grass
86	vegetation	tree	water	273	caribou	antlers	grass
87	house	tree	water	274	herd	zebra	grass
88	village	water	tree	275	moose	antlers	tree
89	carvings	close-up	wood	276	clearing	moose	tree
90	path	garden	tree	277	mare	horses	foals
91	wood	nest	birds	278	foals	horses	mare
92	dress	people	woman	279	orchid	flowers	grass
93	coast	water	waves	280	lily	flowers	leaf
94	sailboats	water	sky	281	stems	flowers	plants
95	cat	tiger	water	282	row	field	tree
96	tiger	cat	water	283	chrysanthemums	petals	flowers
97	bengal	tiger	cat	284	blooms	flowers	cactus
98	fox	snow	grass	285	cactus	flowers	blooms
99	kit	fox	rocks	286	saguaro	cactus	blooms
100	run	snow	fox	287	giraffe	tree	grass
101	shadows	stone	snow	288	zebra	grass	herd
102	winter	snow	sky	289	tusks	ground	water
103	autumn	coyote	fox	290	hands	bear	people
104	cliff	water	sky	291	train	railroad	locomotive
105	bush	flowers	grass	292	desert	sand	valley
106	rockface	fox	ground	293	dunes	sand	sky
107	pair	fox	snow	294	canyon	valley	rocks
108	den	fox	rocks	295	lighthouse	coast	water
109	coyote	snow	sky	296	mast	boats	water
110	light	buildings	street	297	seals	pups	rocks
111	arctic	fox	snow	298	texture	pattern	sand
112	shore	water	grass	299	dust	tree	head
113	town	street	water	300	pepper	plants	leaf
114	road	tree	people	301	swimmers	people	pool
115	chapel	sky	water	302	pyramid	stone	ruins
116	moon	sky	water	303	mosque	stone	pillar
117	harbor	water	boats	304	sphinx	stone	statue
118	windmills	sky	water	305	truck	people	bear
119	restaurant	people	water	306	fly	birds	tree
120	wall	people	cars	307	trunk	head	elephant
121	skyline	buildings	street	308	baby	cubs	people
122	window	buildings	sky	309	eagle	flight	birds
123	clothes	people	shops	310	lynx	tree	cat
124	shops	street	people	311	rodent	water	rocks
125	street	buildings	people	312	squirrel	rodent	ground
126	cafe	people	fence	313	goat	tree	mountain
127	tables	people	restaurant	314	marsh	water	hills
128	nets	people	bear	315	wolf	snow	tree
129	crafts	people	ground	316	pack	man	wolf
130	roofs	sky	buildings	317	dall	sheep	grass

131	ruins	stone	clouds	318	porcupine	rodent	close-up
132	stone	ruins	sculpture	319	whales	water	beach
133	cars	tracks	turn	320	rabbit	rodent	snow
134	castle	sky	water	321	tracks	cars	turn
135	courtyard	people	buildings	322	crops	fruit	field
136	statue	stone	buildings	323	animals	tree	mountain
137	stairs	buildings	ruins	324	moss	hawaii	race
138	costume	people	hats	325	trail	tree	water
139	sponges	ocean	coral	326	locomotive	train	railroad
140	sign	street	writing	327	railroad	train	locomotive
141	palace	sky	tree	328	vehicle	locomotive	train
142	paintings	wall	people	329	aerial	hawaii	race
143	sheep	grass	dall	330	range	mountain	sky
144	valley	mountain	sand	331	insect	butterfly	plants
145	balcony	buildings	flowers	332	man	people	indian
146	post	buildings	window	333	woman	people	indian
147	gate	sky	people	334	rice	food	field
148	plaza	buildings	people	335	prayer	room	people
149	festival	people	sky	336	glacier	valley	clouds
150	temple	stone	people	337	harvest	road	tree
151	sculpture	stone	ruins	338	girl	people	costume
152	museum	people	tree	339	indian	people	woman
153	hotel	buildings	tree	340	pole	people	crystals
154	art	museum	sculpture	341	dance	people	sky
155	fountain	water	tree	342	african	people	woman
156	market	people	shops	343	shirt	people	man
157	door	window	buildings	344	buddhist	sky	temple
158	mural	tree	garden	345	tomb	pyramid	designs
159	garden	flowers	tree	346	outside	museum	people
160	star	ocean	sea	347	shade	castle	water
161	butterfly	swimmers	pool	348	formula	tracks	cars
162	angelfish	ocean	fish	349	turn	tracks	cars
163	lion	desert	rocks	350	straightaway	tracks	cars
164	cave	rocks	people	351	prototype	tracks	cars
165	crab	rocks	water	352	steel	bridge	arch
166	grouper	hawaii	race	353	scotland	water	mountain
167	pagoda	buddhist	sky	354	ceiling	furniture	nets
168	buddha	sculpture	shrine	355	furniture	ceiling	nets
169	decoration	street	sky	356	lichen	rocks	sky
170	monastery	buildings	people	357	pups	seals	rocks
171	landscape	tree	flowers	358	antelope	desert	sand
172	detail	writing	sign	359	pebbles	rocks	water
173	writing	sign	wall	360	remains	antelope	desert
174	sails	skyline	boats	361	leopard	branch	tree
175	food	people	market	362	jeep	helicopter	flag
176	room	prayer	floor	363	calf	baby	sand

177	entrance	sky	buildings	364	reptile	lizard	log
178	fruit	frost	ice	365	snake	reptile	desert
179	night	buildings	city	366	cougar	head	snow
180	perch	birds	market	367	oahu	people	beach
181	cow	grass	elk	368	kauai	people	beach
182	figures	sculpture	statue	369	maui	water	tree
183	facade	buildings	sky	370	school	oahu	fish
184	chairs	people	tables	371	canoe	race	maui
185	guard	street	people	372	race	canoe	maui
186	pond	garden	flowers	373	hawaii	tree	water

APPENDIX C

C.1 Sample Query Results for the proposed methodology.



1071.jpeg



regions of 1071.jpeg

<p>1043.jpeg: beach people sea sun</p>	<p>1039.jpeg: birds sea sun waves</p>	<p>1044.jpeg: birds dog sea sun</p>	<p>1053.jpeg: clouds sea sun tree</p>
<p>1072.jpeg: boats clouds sun water</p>	<p>12067.jpeg: coral ocean reefs</p>	<p>1045.jpeg: hills sea sky sun</p>	<p>171047.jpeg: frost sky sunset tree</p>
<p>335093.jpeg: sand shrubs</p>	<p>1009.jpeg: boats clouds sun</p>	<p>187039.jpeg: grass park tiger tree</p>	<p>187083.jpeg: close-up clothes people woman</p>
<p>231030.jpeg: hills rocks scotland water</p>	<p>296019.jpeg baby close-up people shirt</p>	<p>335046.jpeg coast ocean rocks sky</p>	<p>173094.jpeg: head rocks rodent</p>
<p>13082.jpeg: flowers grass petals poppies</p>	<p>183019.jpeg: bear cubs grass polar</p>	<p>335025.jpeg: dunes river water</p>	<p>142070.jpeg: close-up wood</p>

C.2.

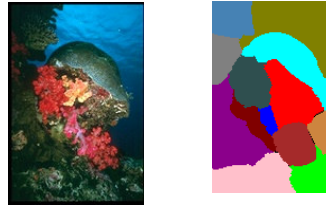


1026.jpeg












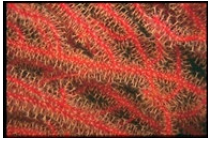








regions of 1026.jpeg

<p>10094: plane prop sky</p>	<p>10092.jpeg: plane prop sky</p>	<p>10008.jpeg: flight helicopter sky</p>	<p>34070.jpeg: helicopter prop sky</p>
<p>34005.jpeg: jet plane sky</p>	<p>34041.jpeg: clouds jet plane</p>	<p>34044.jpeg: clouds jet plane</p>	<p>34039.jpeg: clouds jet plane</p>
<p>10080.jpeg: jet plane smoke</p>	<p>100023.jpeg: bear ice polar snow</p>	<p>10083.jpeg: jet plane sky</p>	<p>34050.jpeg: clouds jet plane</p>
<p>34090.jpeg: clouds jet plane</p>	<p>10011.jpeg: jet plane sky</p>	<p>10097.jpeg: plane prop sky</p>	<p>276015.jpeg: clouds mountain sky tree</p>
<p>100003.jpeg: bear polar snow tundra</p>	<p>276005.jpeg: mountain reflection sky water</p>	<p>130086.jpeg: elephant herd plane sky</p>	<p>34068: jet plane sky</p>

C.3.



















12019.jpeg regions of 12019.jpeg

 <p>12022.jpeg: coral ocean rocks</p>	 <p>13016.: field flowers sky tulip</p>	 <p>13096.jpeg: flowers petals tulip</p>	 <p>12098.jpeg : coral crab ocean reefs</p>
 <p>231063.jpeg: clouds field hills scotland</p>	 <p>102050.jpeg: flowers garden town</p>	 <p>231079.jpeg: bridge river scotland water</p>	 <p>13015.jpeg: field flowers sky tulip</p>
 <p>118072.jpeg: courtyard stairs stone</p>	 <p>143042.jpeg: grass road sky valley</p>	 <p>182058.jpeg: bridge locomotive train water</p>	 <p>12040.jpeg: coral ocean reefs</p>
 <p>12034.jpeg: coral crab ocean reefs</p>	 <p>144049.jpeg: boats house sky water</p>	 <p>130009.jpeg: ground mountain tree</p>	 <p>1012.jpeg: branch leaf sun</p>
 <p>12060.jpeg: coral ocean reefs</p>	 <p>12033.jpeg: coral fish ocean people</p>	 <p>296035.jpeg: branch grass sky</p>	 <p>12025.jpeg coral ocean reefs</p>

C.4.



13055.jpeg regions of 13055.jpeg

 <p>13062.jpeg: flowers grass lily tulip</p>	 <p>12014.jpeg: coral fish ocean</p>	 <p>12076.jpeg: coral ocean reefs</p>	 <p>13015.jpeg: field flowers sky tulip</p>
 <p>13017.jpeg: flowers petals row tulip</p>	 <p>152085.jpeg: close-up flowers leaf</p>	 <p>13009.jpeg: flowers petals stems tulip</p>	 <p>20059.jpeg: buddhist pagoda temple</p>
 <p>140083.jpeg: people tree water</p>	 <p>131038.jpeg: tower tree</p>	 <p>13091.jpeg: flowers grass petals</p>	 <p>152075.jpeg: close-up leaf lily plants spider</p>
 <p>13000.jpeg: flowers mountain sky tulip</p>	 <p>13019.jpeg: flowers petals stems tulip</p>	 <p>152046.jpeg: flowers leaf plants</p>	 <p>17071.jpeg: grass tree</p>

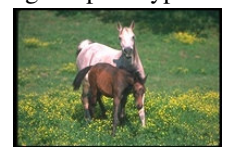
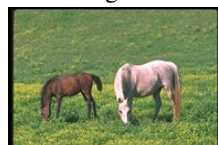
C.5.



105049.jpeg













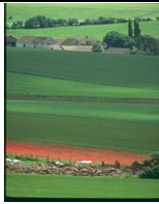









regions of 105049.jpeg

108025.jpeg
cat grass tiger tree163052.jpeg
birds branch house tree108059.jpeg
cat forest grass tiger113004.jpeg
fence foals horses108044.jpeg
cat grass tiger water100084.
bear grass grizzly meadow113006.jpeg
foals horses tree100086.jpeg
bear field grass grizzly21087.jpeg
cars grass prototype tracks108021.jpeg
bengal cat grass tiger108029.jpeg
cat grass tiger water100078.jpeg
bear grass grizzly head113045.jpeg
field foals horses mare113003.jpeg
bush field grass horses113002.jpeg
foals grass horses mare113089.jpeg
field foals horses mare113060.jpeg
field foals horses mare113068.jpeg
field foals horses mare108033.jpeg
cat grass tiger water113058.jpeg
field foals horses mare

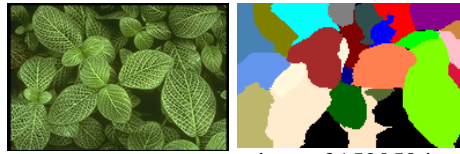
C.6.



113091.jpeg regions of 113091.jpeg














 <p>113070.jpeg field foals horses mare</p>	 <p>103043.jpeg birds rocks</p>	 <p>113066.jpeg field foals horses mare</p>	 <p>113032.jpeg field grass horses tree</p>
 <p>113018.jpeg field foals horses</p>	 <p>113042.jpeg field foals horses mare</p>	 <p>113095.jpeg field foals horses</p>	 <p>113099.jpeg field foals horses mare</p>
 <p>104074.jpeg antlers caribou field</p>	 <p>113073. field foals horses mare</p>	 <p>174025.jpeg farms field flowers poppies</p>	 <p>113063.jpeg field foals horses</p>
 <p>113051.jpeg branch field flowers horses</p>	 <p>113049. field foals horses mare</p>	 <p>108004.jpeg cat forest grass tiger</p>	 <p>113033. fence foals horses mare</p>
 <p>113052.jpeg field foals horses mare</p>	 <p>100085. bear grizzly stream water</p>	 <p>113069. field foals horses mare</p>	 <p>152079.jpeg close-up leaf plants</p>

C.7.



152059.jpeg

regions of 152059.jpeg

			
152091.jpeg leaf palm plants	163018.jpeg birds grass nest	152056.jpeg leaf plants	163020.jpeg birds grass nest
			
152071.jpeg leaf plants	152082.jpeg flowers leaf plants	152054.jpeg branch leaf plants	152099.jpeg leaf plants stems wall zebra
			
131036.jpeg garden lawn tree	102047.jpeg garden plants tree water	13056.jpeg flowers grass tree	13080.jpeg flowers petals stems
			
108023.jpeg cat grass tiger	163029.jpeg birds nest tree	152087.jpeg leaf plants tree	152012.jpeg branch leaf plants
			
152013.jpeg leaf plants stems	152077.jpeg leaf plants	152092.jpeg close-up palm plants	131059.jpeg flowers garden pond water
			
152088.jpeg leaf plants	131031.jpeg flowers garden pond tree	13072.jpeg flowers leaf petals	131089.jpeg bridge flowers garden water




















APPENDIX D

D. Sample Clusters of Image Annotations

D.1.a Cluster 26 and image annotations

Document Id	Annotations
653 (119024.jpeg)	buildings sky statue street
668 (119039.jpeg)	buildings statue street tree
670 (119041.jpeg)	buildings statue street tree
678 (119051.jpeg)	buildings sky statue street
699 (119074.jpeg)	buildings plaza statue street
893 (120093.jpeg)	buildings sky statue
897 (120097.jpeg)	people statue street
927 (121029.jpeg)	reflection rocks statue water
930 (121032.jpeg)	people rocks statue water
1312 (147055.jpeg)	column people statue street
1362 (148014.jpeg)	horses people statue tree
1383 (148037.jpeg)	reflection statue water
1386 (148041.jpeg)	buildings night statue
2159 (142000.jpeg)	buildings people plaza statue
2620 (161012.jpeg)	people statue stone tree
2645 (161038.jpeg)	sphinx statue stone tree
2646 (161039.jpeg)	sphinx statue stone tree
2647 (161040.jpeg)	sphinx statue stone tree
2662 (161056.jpeg)	people statue stone
2667 (161061.jpeg)	people statue stone
2689 (161086.jpeg)	people sphinx statue stone
3427 (201007.jpeg)	buildings designs statue tomb
4451 (46044.jpeg)	buildings sky statue

D.1.b Cluster 26's Images

 119024.jpeg	 119039.jpeg	 119041.jpeg	 119051.jpeg	 119074.jpeg
 120093.jpeg	 120097.jpeg	 121029.jpeg	 121032.jpeg	 147055.jpeg
 148014.jpeg	 148037.jpeg	 148041.jpeg	 142000.jpeg	 161012.jpeg
 161038.jpeg	 161039.jpeg	<not available> 61040.jpeg	 161056.jpeg	 161061.jpeg
 161086.jpeg	 201007.jpeg	<not available> 46044.jpeg		

D.2.a Cluster 38 and image annotations

Document Id	Annotations
1761 (103054.jpeg)	birds booby flight
1762 (103055.jpeg)	birds booby flight
1763 (103056.jpeg)	birds booby
1764 (103058.jpeg)	booby flight
1765 (103059.jpeg)	birds booby flight
1766 (103061.jpeg)	birds booby rocks
1767 (103063.jpeg)	birds booby rocks
1768 (103064.jpeg)	birds booby perch
1769 (103065.jpeg)	birds booby
1770 (103066.jpeg)	birds booby flight
1771 (103068.jpeg)	birds booby nest
1772 (103070.jpeg)	birds booby rocks
1773 (103071.jpeg)	birds booby flight
1774 (103072.jpeg)	birds booby rocks
1775 (103073.jpeg)	birds booby rocks




















D.2.b Cluster 38's Images

 103054.jpeg	 103055.jpeg	 103056.jpeg	 103058.jpeg
 103061.jpeg	 103063.jpeg	 103064.jpeg	 103065.jpeg
 103068.jpeg	 103070.jpeg	 103071.jpeg	 103072.jpeg
 103059.jpeg	 103066.jpeg	 103073.jpeg	

D.3.a Cluster 71 and image annotations

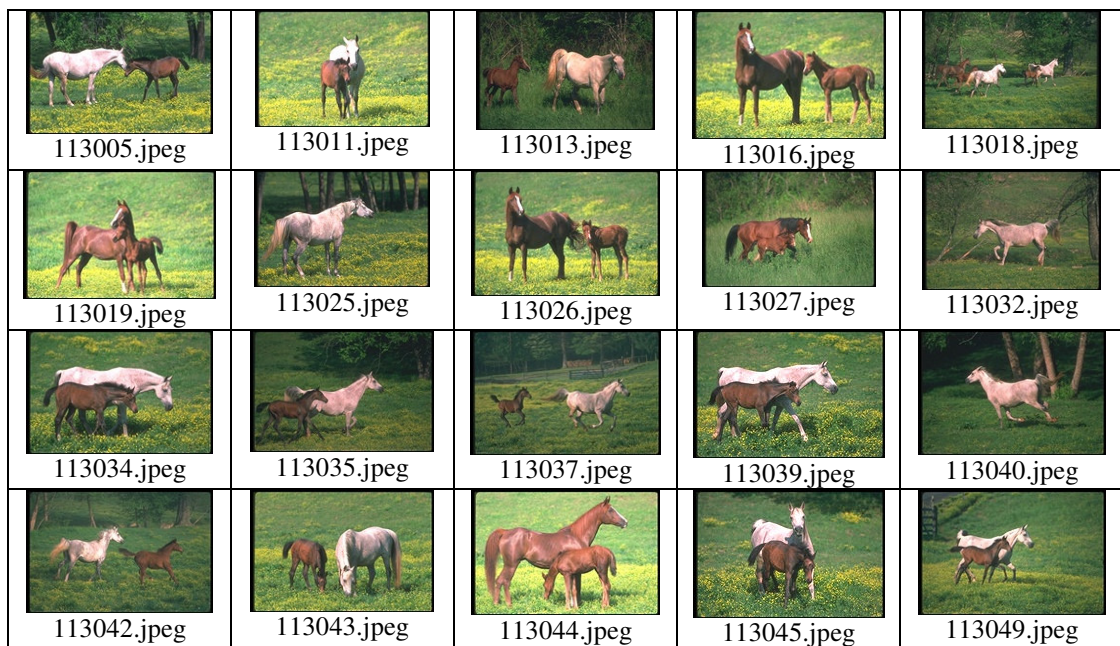
Document Id	Annotations
242 (100069.jpeg)	bear water
244 (100071.jpeg)	bear grass grizzly water
245 (100072.jpeg)	bear
247 (100074.jpeg)	bear grizzly water
249 (100076.jpeg)	bear water
254 (100082.jpeg)	bear water
258 (100086.jpeg)	bear field grass grizzly
260 (100089.jpeg)	bear grass grizzly water
261 (100090.jpeg)	bear grizzly water
262 (100091.jpeg)	bear water
263 (100093.jpeg)	beach bear grizzly water
265 (100095.jpeg)	bear water
266 (100096.jpeg)	beach bear grizzly water
267 (100097.jpeg)	bear water
268 (100098.jpeg)	bear grass grizzly snow
2817 (173029.jpeg)	bear grizzly water
2818 (173032.jpeg)	bear grizzly tree water
3065 (183005.jpeg)	bear snow tracks
3110 (183056.jpeg)	bear hands
4365 (41051.jpeg)	bear grizzly tree water
4368 (41054.jpeg)	bear grizzly water
4369 (41055.jpeg)	bear grizzly water

D.3.b Cluster 71's Images

 100069.jpeg	 100071.jpeg	 100072.jpeg	 100074.jpeg	 100076.jpeg
 100082.jpeg	 100086.jpeg	 100089.jpeg	 100090.jpeg	 100091.jpeg
 100093.jpeg	 100095.jpeg	 100096.jpeg	 100097.jpeg	 100098.jpeg
 173029.jpeg	 173032.jpeg	 183005.jpeg	 183056.jpeg	<not available> 41051.jpeg

D.4.a A part of cluster 55 (totally this cluster contains 77 images)

Document Id	Annotations
1893 (113005.jpeg)	field foals horses mare
1898 (113011.jpeg)	field foals horses mare
1900 (113013.jpeg)	field foals horses mare
1903 (113016.jpeg)	field foals horses mare
1905 (113018.jpeg)	field foals horses
1906 (113019.jpeg)	field foals horses mare
1912 (113025.jpeg)	field grass horses tree
1913 (113026.jpeg)	field foals horses mare
1914 (113027.jpeg)	field foals horses mare
1918 (113032.jpeg)	field grass horses tree
1920 (113034.jpeg)	field foals horses mare
1921 (113035.jpeg)	field foals horses mare
1923 (113037.jpeg)	field foals horses mare
1925 (113039.jpeg)	field foals horses mare
1926 (113040.jpeg)	field grass horses tree
1928 (113042.jpeg)	field foals horses mare
1929 (113043.jpeg)	field foals horses mare
1930 (113044.jpeg)	field foals horses mare
1931 (113045.jpeg)	field foals horses mare
1933 (113049.jpeg)	field foals horses mare

D.4.b A part of images from Cluster 55.

APPENDIX D

D.1 Singular Value Decomposition

SVD takes a document matrix A and represents it as A' in a lower dimensional space such that the “distance” between the two matrices as measured by the 2 -norm is minimized:

$$\Delta = \|A - A'\|_2$$

The 2 -norm for matrices is the equivalent of Euclidean distance for vectors. SVD project an n -dimensional space onto a k -dimensional space where $n \gg k$ (n is much more greater than k). In most application (word-document matrices), n is the number of word types in the collection. Values of k that are frequently chosen are 200 and 300. The projection transforms a document's vector in n -dimensional word space into a vector in the k -dimensional reduced space.

There are many different mappings from high dimensional to low-dimensional spaces. Latent Semantic Indexing chooses the mapping that is optimal in the sense that it minimizes the distance Δ . This setup has the consequence that the dimensions of the reduced space correspond to the axes of greatest variation. (This is closely related to Principal Component Analysis (PCA), another technique for dimensionality reduction. One difference between the two techniques is that PCA can only be applied to a square matrix whereas LSI can be applied to any matrix.)

The SVD projection is computed by decomposing (Figure E.1) the document-by-term matrix $A_{t \times d}$ into the product of three matrices, $T_{t \times n}$, $S_{n \times n}$, $D_{d \times n}$.

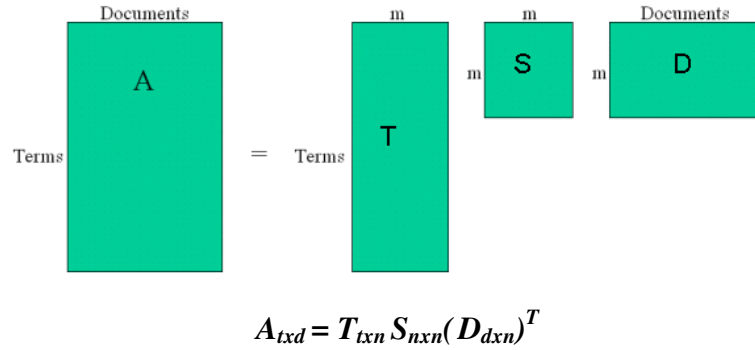


Figure E.1: Decomposition of document matrix for SVD.

Where t is the number of terms, d is the number of documents, $n = \min(t, d)$, T and D have orthonormal columns, *i.e.* $I = TT^T = DD^T$, $\text{rank}(A) = r$. U is $t \times n$ column-orthonormal matrix whose columns are called left singular vectors; and $S = \text{diag}(s_1, s_2, s_3, \dots, s_n)$ is an $n \times n$ diagonal matrix whose diagonal elements are non-negative singular values stored in descending order, and D is an $d \times n$ orthogonal matrix whose columns are called right singular vectors. If $\text{rank}(A) = r$ then S satisfies;

$$s_1 \geq s_2 \geq s_3 \geq \dots \geq s_r > s_{r+1} = \dots = s_n = 0$$

SVD methods are based on the following theorem of linear algebra, whose proof is beyond our scope: Any $t \times d$ matrix A whose number of rows t is greater than or equal to its number of columns d ($n = d$ here), can be written as the product of an $t \times d$ column-orthogonal matrix T , an $d \times d$ diagonal matrix S with positive or zero elements (the *singular values*), and the transpose of an $d \times d$ orthogonal matrix D .

One can also prove that SVD is unique, that is, there is only one possible decomposition of a given matrix. That SVD finds the optimal projection to a low-dimensional space is the key property for exploiting word co-occurrence patterns. It is important for the LSI method that the derived A' matrix does not reconstruct the original term document matrix A exactly. The truncated SVD, in one sense, captures most of the important underlying structure in the association of terms and documents, yet at the same time removes the noise or variability in word usage that plagues word-based retrieval methods. Intuitively, since the number of dimensions, k , is

much smaller than the number of unique terms, t , minor differences in terminology will be ignored. Terms, which occur in similar documents, for example, will be near each other in the k -dimensional factor space even if they never co-occur in the same document. This means that some documents, which do not share any words with a user's query, may nonetheless be near it in k -space. This derived representation, which captures term-term associations, is used for retrieval. Refer to SVD Algorithm (Bronson, 1989) for the algorithm of SVD.