LARGE-SCALE IMAGE COLLECTION CLEANSING, SUMMARIZATION AND EXPLORATION

by

Chunlei Yang

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2012

Approved by:

____

Dr. Jianping Fan

____

Dr. William Ribarsky

____

Dr. Zbigniew W. Ras

____

Dr. Jing Xiao

____

Dr. Nigel Zheng

©2012
Chunlei Yang
ALL RIGHTS RESERVED

ABSTRACT

CHUNLEI YANG. Large-scale image collection cleansing, summarization and exploration. (Under the direction of DR. JIANPING FAN)

A perennially interesting topic in the research field of large scale image collection organization is how to effectively and efficiently conduct the tasks of image cleansing, summarization and exploration. The primary objective of such an image organization system is to enhance user exploration experience with redundancy removal and summarization operations on large-scale image collection. An ideal system is to discover and utilize the visual correlation among the images, to reduce the redundancy in large-scale image collection, to organize and visualize the structure of large-scale image collection, and to facilitate exploration and knowledge discovery.

In this dissertation, a novel system is developed for exploiting and navigating large-scale image collection. Our system consists of the following key components: (a) junk image filtering by incorporating bilingual search results; (b) near duplicate image detection by using a coarse-to-fine framework; (c) concept network generation and visualization; (d) image collection summarization via dictionary learning for sparse representation; and (e) a multimedia practice of graffiti image retrieval and exploration.

For junk image filtering, bilingual image search results, which are adopted for the same keyword-based query, are integrated to automatically identify the clusters for the junk images and the clusters for the relevant images. Within relevant image clusters, the results are further refined by removing the duplications under a coarse-to-fine structure. The duplicate pairs are detected with both global feature (partition based color histogram) and local feature (CPAM and SIFT Bag-of-Word model). The duplications are detected and removed from the data collection to facilitate further exploration and visual correlation analysis. After junk image filtering and duplication removal, the visual concepts are further organized and visualized by the proposed concept network. An automatic algorithm is developed to generate such visual concept network which characterize the visual correlation between image concept pairs. Multiple kernels are combined and a kernel canonical correlation analysis algorithm is used to characterize the diverse visual similarity contexts between the image concepts. The FishEye visualization technique is implemented to facilitate the navigation of image concepts through our image concept network. To better assist the exploration of large scale data collection, we design an efficient summarization algorithm to extract representative examplars. For this collection summarization task, a sparse dictionary (a small set of the most representative images) is learned to represent all the images in the given set, e.g., such sparse dictionary is treated as the summary for the given image set. The simulated annealing algorithm is adopted to learn such sparse dictionary (image summary) by minimizing an explicit optimization function.

In order to handle large scale image collection, we have evaluated both the accuracy performance of the proposed algorithms and their computation efficiency. For each of the above tasks, we have conducted experiments on multiple public available image collections, such as ImageNet, NUS-WIDE, LabelMe, etc. We have observed very promising results compared to existing frameworks. The computation performance is also satisfiable for large-scale image collection applications. The original intention to design such a large-scale image collection exploration and organization system is to better service the tasks of information retrieval and knowledge discovery. For this purpose, we utilize the proposed system to a graffiti retrieval and exploration application and receive positive feedback.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Motivation

The new century has witnessed the information explosion, especially for online and offline digital images and photos. The actual number of images available on the Internet has became impossible to estimate and the only thing we know for sure is that this number keeps growing everyday with the contribution from all the Internet users. Compared to the fast growth of online images, the technique development for supporting organization and visualization of large-scale image collection has been lagging behind. From the end users' point of view, the image search engines such as Google Image or Bing Image provide as convenient image search tools with both text queries and example queries, which lead the users into the exploration process under the specific query category. The images returned would be either based on its relevancy on textual property or visual property. However, such image search engines could not provide the users with an overview of the relationship between the query category and other related categories. The results returned are based solely on the semantic-level relatedness to the query word. Google has developed Google Image Swirl [76] which is able to support exploring multiple related categories in a sequence order. Once the users find the group of interested images, they can visit the neighboring clusters which will be further "swirled" into view and the clusters are linked based on their visual and semantic relevancy. Other online photo-sharing web sites such as Flickr [63] and Picasa [64] support the exploration of the collection with the ways beyond query search, e.g. to explore by upload date, by tag, by group, etc. For online photo-sharing web sites, the more relevant information provided, the more accurate the exploration and knowledge discovery operation will be done.

Besides the image search engines and commercial social photo sharing web sites, there are also a number of image collections for research purpose, such as ImageNet [40], Caltech-256 [55], NUS-WIDE [17], etc. These image collections are organized by semantic hierarchy of the category names. For example, the image categories are organized in a top-down tree structure with the most generative terms on the top and more detailed terms on the branches. For Caltech-256, the first level nodes are “animated entity” and “non-animated entity”, and the “animated entity” could be further partitioned into “animal”, “plants”, etc. A hierarchical structure organization of image categories is straightforward, yet underestimates the complexity of correlation between the categories and between the images. Briefly, the organization of large quantities of visual information is identified as an important research topic and is attracting more and more attention from the researchers in the related fields.

The research work delivered in this thesis is motivated by the fact that, with the exponential availability of high-quality digital images, there is an urgent need to develop new frameworks for efficient and interactive image collection navigation and exploration [136, 59], other than plain list of the search results or a tree structure organization. The image collections should be organized based on the visual concept ontology, rather than a hierarchical structure. However, as mentioned above, most of the current frameworks may not be able to support effective navigation and exploration for visual concept network construction: (a) Current techniques only consider the hierarchical inter-concept relationships. When large-scale online image collections come into view, the inter-concept similarity relationship could be more complex than the hierarchical ones. (b) Only the inter-concept semantic relationships are exploited for concept ontology construction, thus the concept ontology cannot allow users to navigate large-scale online image collections according to their visual similarity contexts. We determined to develop novel image collection organization frameworks to deal with the above difficulties.

An efficient and practical organization of the image categories will further service other research objectives: (a) By incorporating the image search results from multiple sources, relevant cluster and irrelevant clusters could be detected (junk image filtering). (b) By investigating the visual correlation among different categories, visually correlated classes could be associated to support multi-class classifier training (image classification). (c) By organizing inter-concept and intra-concept groups, the relationship between image groups could be discovered (image browsing). (d) By clustering the images within given group or category, find a query image for certain purpose such as duplicate detection (image retrieval and duplicate detection). (e) By selecting the most representative images from a given set, the system can recommend the user with a summary of the collection (image collection summarization). Extensive research work has been done on each of the above research topics, however, these work has been done independently and we have not seen a complete image collection organization framework to incorporate all these research components, or evaluate the

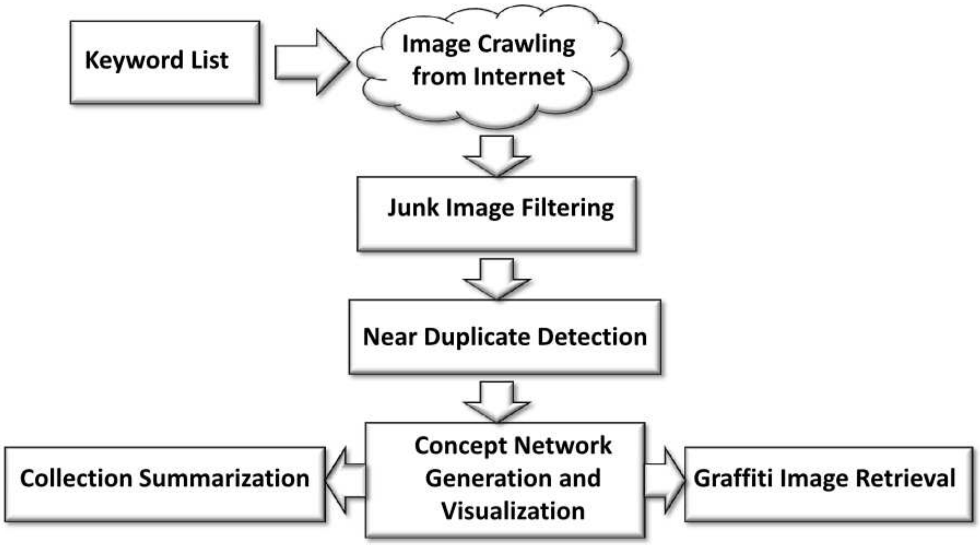


Figure 1: Complete System Work Flow

results of such a navigation system with its associated research tasks. The research and evaluation on such a system will greatly benefit the applications such as large-scale image collection exploration, navigation, image retrieval and recommendation.

1.2 Overview

As discussed in the previous section, a series of research tasks will be studied and presented in this thesis as the key components to construct the image collection organization system. The overview of the complete system is shown in Figure 1. In this section, we will introduce each of these components: junk image filtering, near-duplicate detection, concept network generation and visualization, image collection summarization and a practical information retrieval application: graffiti retrieval. We will also present the basic idea of each of these research tasks and discuss how our proposed algorithms are able to tackle the challenges that most existing works have encountered.

1.2.1 Junk Image Filtering

Keyword-based image search engines have achieved great success on exploiting the associated text terms for indexing large-scale web image collections. Unfortunately, most existing keyword-based image search engines are still unsatisfactory because of the appearances of large amounts of junk images [43, 42, 12, 155, 47, 48]. One of the major reasons for this phenomena is due to the fact that most existing keyword-based image search engines simplify the image search problem as a purely text-based search problem, and their fundamental assumption is that the image semantics are directly related to all these associated text terms (that are extracted from the associated text documents or file names). However, such oversimplified assumption has ignored that not all these associated text terms are related exactly with the image semantics (e.g., most of these associated text terms are used for web content description and only a small part of these associated text terms are used for image semantics description) [41, 158]. If all these associated text terms are loosely used for web image indexing, most existing keyword-based image search engines may return large amounts of junk images which may bring huge information overload for users to assess the relevance between large amounts of returned images and their real query intentions. In addition, a lot of real world settings, such as photo-sharing web sites, may only be able to provide biased and noisy text terms for image annotation which may further mislead the keyword-based image search engines. Figure 2 shows the query result for the

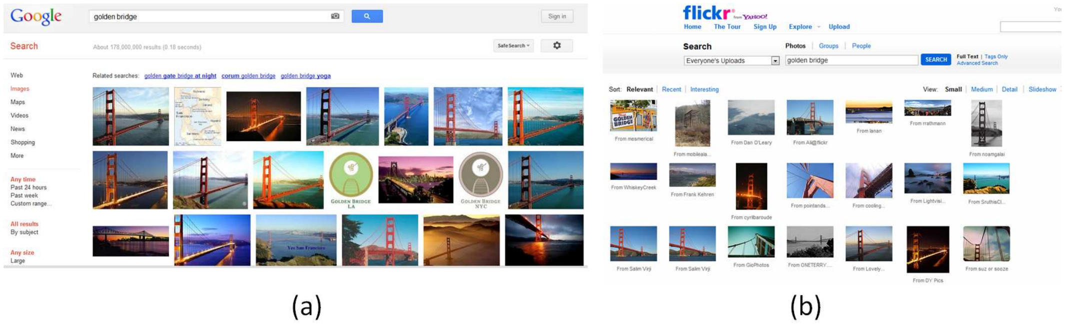


Figure 2: Junk image observation in Google Image and Flickr query results: (a) return result from Google Image with query word “golden gate bridge”; (b) return result from Flickr with query word “golden gate bridge”.

keyword “golden gate bridge” of Google Image and Flickr. We can observe from the result that there are junk images even in the first page of return results. For Google Image, the junk images are from the map of golden gate bridge, an award with the same name, a spiritual village of the same name, and a watch brand with the same name. For Flickr, same tags are associated to a drink brand, and a different place with the same name. Therefore, the existence of large quantity of junk images from text-based return results urge us to develop new algorithms for leveraging other alternative information sources to filter out the junk images automatically.

There are two alternative information sources that can be leveraged for filtering out the junk images from the keyword-based image search results: (a) visual properties of the returned images [43, 42, 12, 155, 47, 48]; (b) visual correlations between the search results for the same keyword-based query which is simultaneously performed on multiple keyword-based image search engines in the same language or even different languages.

The visual properties of the returned images and the visual similarity relationships between the returned images are very important for users to assess the relevance between the returned images and their real query intentions. Unfortunately, most existing keyword-based image search engines have completely ignored such important characteristics of the images. Even the low-level visual features may not be able to carry the image semantics directly, they can definitely be used to filter out the junk images and enhance users' abilities on finding some particular images according to their visual properties [43, 42, 12, 155, 47, 48]. With the increasing computational power of modern computers, it is possible to incorporate image analysis algorithms into the keyword-based image search engines without degrading their response speed significantly. Recent advance in computer vision and multimedia computing can also allow us to take advantages of the rich visual information (embedded in the web images) for image semantics interpretation. Some pioneering work have been done by integrating the visual properties of the returned images to improve the performance of Google Images [43, 42, 12, 155, 47, 48].

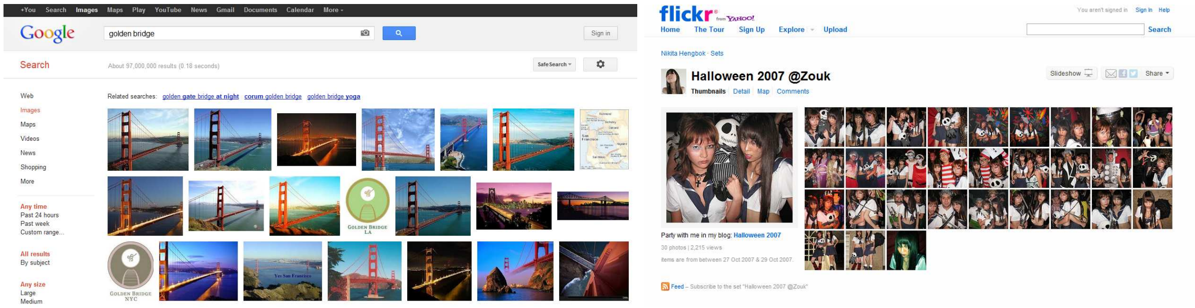
There are many keyword-based image search engines in the same language (such as Google Images and Bing) or in different languages (such as Google Images in English and Baidu Images in Chinese). All these keyword-based image search engines crawl large-scale web images from the same or similar web sources, the relevant images for the same keyword-based query (which is simultaneously performed on different keyword-based image search engines in same language or even in different languages) may have strong correlations on their visual properties. On one hand, the relevant images (one part of the returned images), which are returned by different keyword-based image search engines for the same keyword-based query, should share some common or similar visual properties. On the other hand, the junk images (another part of the returned images), which are returned by different keyword-based image search engines for the same keyword-based query, may have different visual properties. Such phenomena (the relevant images for the same query from different keyword-based image search engines may share some common or similar visual properties) can be treated as an alternative information source for junk image filtering. Besides the keyword-based image search engines (such as Google Images) in English, there are many other keyword-based image search engines in different languages such as Baidu Images in Chinese. Thus it is very attractive to integrate bilingual or multi-lingual image search results for automatic junk image filtering.

Based on the above observations, an interesting approach is developed in this thesis for filtering out the junk images automatically by integrating the bilingual image search results from two keyword-based image search engines, e.g., Google Images in English and Baidu Images in Chinese. Junk image filtering is conducted on the basis of effective image clustering which reveals the visual correlation among set members.

1.2.2 Near Duplicate Detection

After junk image filtering, the output data collections are still redundant by that there exists large amounts of duplicate or near duplicate image pairs, which will cause big burden for any exploration or query operations, as well as for other tasks such as image collection summarization.

The existence of duplicate or near duplicate image pairs is universally observed in text-based image search engine return results (such as Google Image: the return results for a certain query word) or personal photo collection (such as Flickr or Picasa



golden bridge

About 97,000,000 results (0.18 seconds)

Related searches: golden gate bridge at night corum golden bridge golden bridge yoga

Halloween 2007 @Zouk

Party with me in my blog: Halloween 2007

30 photos | 2,215 views

items are from between 27 Oct 2007 & 29 Oct 2007.

Feed – Subscribe to the set "Halloween 2007 @Zouk"

Figure 3: Left-hand subfigure is the Google Image search results for the query word of "golden gate bridge"; right-hand sub-figure is the Flickr search results for the query word of "Halloween". Both return results show a significant amount of near duplications.

personal photo album: photos that are consecutively taken at the same location with slightly different shooting angle), as found in Figure 3. Because of the existance of large quantity of available images, it is very difficult if not impossible to identify such near duplicate images manually. Thus it is very important to develop robust methods for detecting the near duplicate images automatically from large-scale image collections [127, 152, 143].

It would be rather convenient for near duplication detection tasks to utilize heterogeneous features like EXIF data from photos [140], or time duration information in video sequences [164]. In fact, such information is not available for most of the data collections which forces us to seek for solution from visual content of the images only. Among content based approaches, many focus on the rapid identification of duplicate images with global signatures, which are able to handle almost identical images [69, 139]. However, near duplicates with changes beyond color, lighting and editing artifacts can only be reliably detected through the use of more reliable local features [173, 140, 74, 79]. Local point based methods, such as SIFT descriptor, have demonstrated impressive performance in a wide range of vision-related tasks, and are particularly suitable for detecting near-duplicate images having complex variations.

Following the same interest point extraction and SIFT feature descriptor scheme, Xu et al. [166] have extended single layer matching framework to spatially aligned pyramid matching framework. The performance will slightly increase by 2-3%, while computation cost with multi-layer framework is expected to grow dramatically in response. Besides, like many other work, only using SIFT descriptor is not adequate to accurately characterize the visual similarity between images. An integration of multi-modal feature representation will be a boost for accurate image characterization.

Different from local patch extraction framework, Mehta et al. has represented the visual signature with Gaussian Mixture Model (GMM) [103]. Each pixel is characterized with both spatial information and visual information (HSV color). For each image, a GMM distribution is learnt, and the similarity between two images are measured by the coherence between the two corresponding GMM distributions. The problem for this framework is obvious that calculating GMM distribution for each image will result in unacceptable computation burden for large scale image collections.

The potential for local approaches is unfortunately underscored by matching and scalability issues as discussed above. Past experience has guided us to seek for balance between efficiency and accuracy in near duplicate detection tasks. In order to speed up the duplicate detection process without sacrificing detection accuracy, we have designed a two-stage detection scheme: the first stage is to partition the image collection into multiple groups by using computational cheap global features; the second stage is to conduct image retrieval with computational expensive local features to extract the near-duplicate pairs. The output of the first stage is supposed to not separate any potential near duplicate pairs and the number of images participated in the second stage retrieval could be dramatically reduced. The visual presentation of the image used in the second stage is Bag-of-Word (BoW) model. We have conducted the interest point extraction operation to all the available images and represent each interest point with SIFT descriptor. A universal code book is generated with k-means clustering algorithm from millions of such interest point descriptors. Each code word is a center of the k-means clustering result. In actually implementation, we have conducted hierarchical k-means to construct the code book. With vector quantization operation, each interest point in an image is mapped to a certain code word and the image can be represented by the histogram of the code words. For the purpose of interest point matching, we only count the matches by the points that fall into the same bin, thus the actual calculation of the histogram is not required. Besides the SIFT Bag-of-Word model, we also implement the CPAM (Color Pattern Appearance Model) feature which is built from YC_bC_r color space and quantized in a similar fashion as BoW model. We also built a universal code book for the CPAM feature with k-means clustering algorithm and each image is encoded with vector quantization technique. Finally, the detection result from both models will be combined together with our multi-modal integration design.

Another issue that will affect the detection speed is the nearest neighbor search scheme. The similarity search methods and indexing schemes on high-dimensional space includes Locality-Sensitive Hashing (LSH) [67], Inverted File Indexing [165] and kd-tree [159]. LSH, proposed by Indyk & Motwani, solves the following similarity search problem, termed $(r, \epsilon) - NN$, in sub-linear time. If, for a point q (query) in d-dimensional space, there exists an indexed point p such that $d(p, q) \leq r$, then LSH will, with high probability, return an indexed point p' such that $d(p', q) \leq (1+\epsilon)r$. This is accomplished by using a set of special hash functions that satisfy the intuitive notion that the probability of a hash collision for two points be related to the similarity (distance) between the points. Inverted File Indexing is originally used in texture document retrieval applications, and it could be adopted to image retrieval application if the image feature is represented with bag-of-word model. The visual words are treated in a similar way as the text terms used in texture document retrieval. In order to cover large varieties of visual properties, usually a very large dictionary is constructed, and we will use a hierarchical k-means scheme to construct and store such dictionary. The similarity search is employed on the hierarchical tree and the approximate nearest neighbor can be quickly located. The kd-tree, as can be conferred from its name, is designed for nearest neighbor search on high dimensional space. For the purpose of image retrieval and duplicate detection, we found the inverted file indexing scheme most satisfiable.

Considering the complexity of the visual content of the images, using only one feature for image representation is not appropriate. Our benchmark work has shown that certain features may serve better for certain types of images, which could be categorized as object images and scene images based on their visual content. In order to further improve the accuracy of near duplicate detection, we have determined to design a system which will incorporate the near duplicate detection results with both two different visual features, not only the local feature. The output result is a combination of the two detection results. From the indexing scheme introduced above, the output from inverted file indexing operation is a list of similarities ranked from high to low. Considering the complexity of the image content, it is not appropriate to use a universal threshold to determine duplicates for all the images. We have observed that the distance of the query image with the non-duplicate images in the database follow a linear distribution. From this assumption, we construct a linear regression model for the retrieval result of the images, and the possible top few similarities value that does not collide with the predicted line is considered as duplicates.

The near duplicate detection operation will be applied to all the images in the target data set. For each of the query images, if true positive near duplicate pairs are detected, then they will be removed from the data collection and the refined data collection can be further used for concept network generation or summarization.

1.2.3 Concept Network Generation and Visualization

The central component for a complete image navigation and exploration system is an efficient organization structure. Derived directly from text-based concept ontology construction, tree structure concept ontology organizations are widely accepted for database exploration and navigation [15, 53]. The drawbacks for tree structure concept ontology on image collection organization are equally obvious: (a) Only the hierarchical inter-concept relationships are considered for concept ontology construction, while there could be more complicate relationship between concepts in large scale image collections; (b) Only the semantic relationships are considered between concepts, while the the visual correlation are totally ignored which is believed more important for image category organization. From this observation, we determine to exploit the visual correlation between each of the two concepts in the image collection, and build linkage between visually correlated concept pairs, which will lead to a network structured organization of concept ontology. The network structure concept ontology has at least the following two advantages compared to hierarchical tree structure concept ontology.

(a) Supporting collection browsing: The network structure concept ontology is more general than tree structure which cannot characterize inter-concept visual correlations directly. Visually related categories may distribute across different branches in the tree structure, therefore cannot be easily explored or compared. For example, the concepts of “seagull” and “dock” are visually correlated, while semantically far away from each other. The semantic distance will be very large for this concept pair, and it will be very difficult to navigate from one concept to the other through tree structured concept ontology. On the other hand, through network structured concept ontology, these two concepts may be easily connected and can be explored interrelatedly. Figure 4 shows a comparison between the tree structured concept ontology from Caltech-256 and the proposed network structure concept ontology of concept network.

(b) Guiding classifier learning: Without a structured organization, for multi-class classification tasks, SVM assumes the classification of category is made independently, which means potential structured information is lost in categority relatedness. Such relatedness means the appearance of one category often implies the existence of another closely related category. Such inter-relatedness of categories can be explicitly represented with the category correlation network. The neighboring category nodes are strongly correlated and their training instances may share similar visual proper-

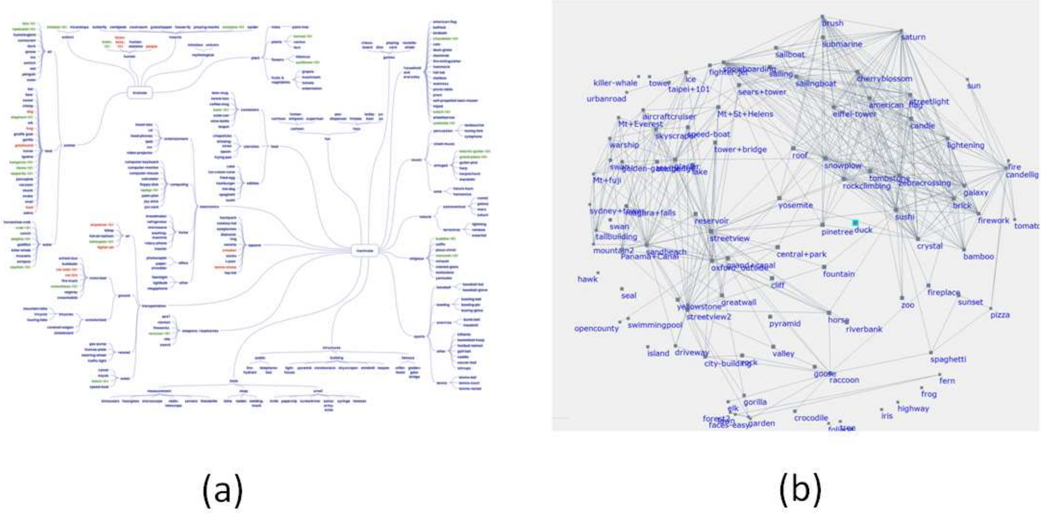


Figure 4: Tree-structure organization vs network structure organization: (a) Caltech-256 tree structure concept ontology [55]; (b) proposed network structure concept ontology of concept network.

ties. Thus isolating these object classes and train their classifiers independently are not appropriate. A multi-task structured learning scheme [130] can be benefited by incorporating neighboring category nodes in such concept network. Furthermore, in multi-label classification tasks [50], if the labels are interdependent, concept network could be used to estimate category co-occurrences. Therefore, it can be used to assist the building of multi-label conditional random field (CRF) classification model where classifiers for the categories are no longer learned independently.

To take full advantage of the network structured concept ontology, we have designed a concept network for image collection organization, which utilize the multi-modal kernel integration for similarity determination, and then visualize with MDS technique for efficient navigation and exploration. The details will be introduced in the rest of this section.

By using high-dimensional multi-modal visual features introduced in the previous subsection for image content representation, it is able for us to characterize the diverse visual properties of the images more sufficiently. On the other hand, the statistical properties of the images in such high-dimensional multi-modal feature space may be heterogeneous because different feature subsets are used to characterize different visual properties of the images, thus the statistical properties of the images in the high-dimensional multi-modal feature space may be heterogeneous and sparse. Therefore, it is impossible for us to use only one single type of kernel to characterize the diverse visual similarity relationships between the images precisely. Therefore, the high-dimensional multi-modal visual features are first partitioned into multiple feature subsets. We have also studied the statistical properties of the images under each feature subset. The gained knowledge for the statistical property of the images under each feature subset has bee used to design the basic image kernel for each feature subset. Because different basic image kernels may play different roles on characterizing the diverse visual similarity relationships between the images, and the optimal kernel for diverse image similarity characterization can be approximated more accurately by using a linear combination of these basis image kernels with different importance. Kernel canonical correlation analysis (KCCA) is a strong tool to analyze the visual similarity between concept nodes, we can build and visualize the concept network which is indicated by the KCCA results. Each concept node together with its first-order neighbor consist a concept clique. The images in the clique share similar visual properties and provide a complete view of the given concept. We will conduct other research topics based on this result, such as efficient data collection exploration, multi-class classification, image retrieval and image recommendation.

To allow users to assess the coherence between the visual similarity contexts determined by our algorithm and their perceptions, it is very important to enable graphical representation and visualization of the visual concept network, so that users can obtain a good global overview of the visual similarity contexts between the image concepts at the first glance. It is also very attractive to enable interactive visual concept network navigation and exploration according to the inherent inter-concept visual similarity contexts, so that users can easily assess the coherence with their perceptions. Based on these observations, our approach for visual concept network visualization exploited hyperbolic geometry [86]. The hyperbolic geometry is particularly well suited for achieving graph-based layout of the visual concept network and supporting interactive exploration. The essence of our approach is to project the visual concept network onto a hyperbolic plane according to the inter-concept visual similarity contexts, and layout the visual concept network by mapping the relevant image concept nodes onto a circular display region. Thus our visual concept network visualization scheme takes the following steps: (a) The image concept nodes on the visual concept network are projected onto a hyperbolic plane according to their inter-concept visual similarity contexts by performing multi-dimensional scaling (MDS) [24] (b) After such similarity-preserving projection of the image concept nodes is obtained, FishEye model is used to map the image concept nodes on the hyperbolic plane onto a 2D display coordinate. FishEye model maps the entire hyperbolic space onto an open unit circle, and produces a non-uniform mapping of the image concept nodes to the 2D display coordinate.

The concept network provides to the users with an overview of the flat-structure

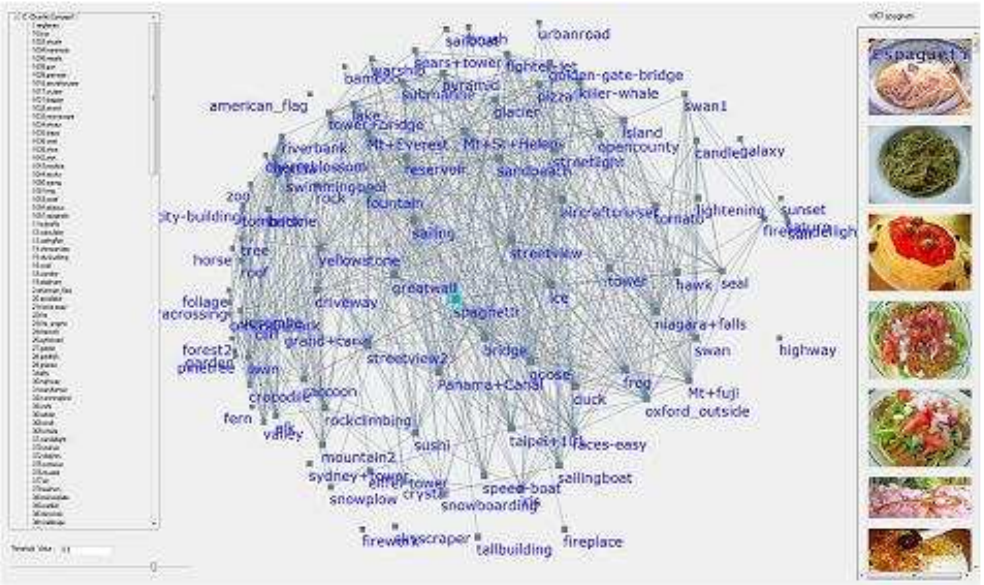


Figure 5: Concept Network Overview: the left tree-view list is the category list in the selected folder; the middle sphere is the concept network with strongly correlated concept nodes linked together; the light blue box in the middle is the currently selected category "spaghetti"; the right panel-view list is the expanded images in the selected category.

of the image collection. The navigation operation could be realized by click action to each of the category nodes of the concept network as shown in 5. For the purpose of further reducing the amount of images within each category and most representative image recommendation, image collection summarization operation can be further conducted on each category, which is also a very important research topic in many other exploration systems.

1.2.4 Image Collection Summarization

Large scale online images are becoming widely available along with the development of search engines such as Google Image and social networks such as Flickr. Such availability, sometimes, leads to a contrary effect and makes useful information "unavailable" or "hidden" from the users that one may easily get lost in the face of huge number of return results from image query, product search, web browsing operations, etc. In such circumstance, automatic image collection summarization, which attempts to select a small set of the most representative images to highlight large amounts of images briefly, becomes critical to enable interactive navigation and efficient exploration of large-scale image collections [75].

Many multimedia applications and business can benefit from automatic image summarization: (a) on-line shopping sites can generate multiple icon images (i.e., image summary) for each category of products by selecting a limited number of the most representative pictures from a large set of product pictures; (b) tourism web sites can generate a small set of the most representative photos from large-scale photo gallery and display on their front page to attract visitors, which may further result in low information overload on user navigation; (c) online image recommendation system learns the user intention at real time and recommends a small amount of most representative images out of a large collection [38]. Such interesting applications have motivated researchers to develop more effective models and mechanisms for achieving more accurate summarization of large-scale image collections.

How to effectively lessen the exploration burden for the end-users, while maintaining the content variety of the original image collection, is the key issue for automatic image collection summarization. A majority of the existing methods use clustering techniques and select the centroids of the clusters as the summaries. The clustering techniques involved in collection summarization task include but not limited to normalized-cut [27], k-means [168, 137], hierarchical clustering [68, 15], SOM [26] and

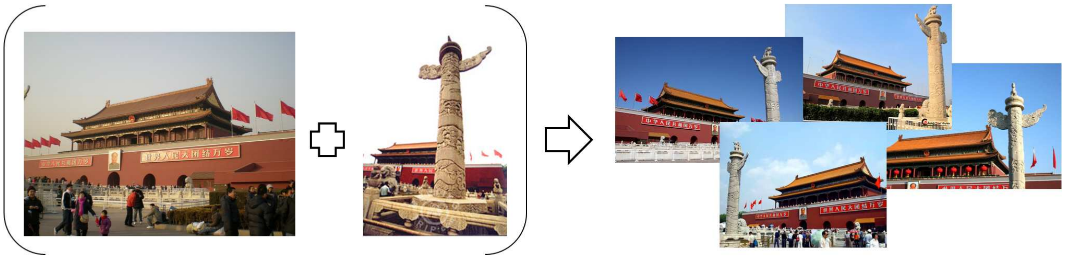


Figure 6: A soft assignment example: right-hand 4 images (rostrum + pillar) can be softly assigned to both two summaries (rostrum and pillar respectively) on left-hand.

similarity graph [1] [75]. One major drawback for the clustering method is that one image is represented by one and only one summary (centroid of the cluster), while in a lot of other cases, soft assignment is necessary. For example, in Figure 6, when tourists took pictures of Tiananmen square, they intend to include as many landmarks in one picture as possible, such as the 4 images on the right-hand of Figure 6. Both the rostrum of Tiananmen and the marble pillar are taken together in one image. This type of images will dominate a Tiananmen square related data collection, and will be selected as a summary by clustering methods. However, it is more straightforward to select the "rostrum" and "marble pillar" as two separate summaries as shown on the left-hand of Figure 6 because they each delivers more clear interpretation of one aspect of the Tiananmen square topic and other images (right-hand 4 images in Figure 6) can therefore be represented in a soft assignment fashion.

Camargo et al. [14] has replaced the clustering model with NMF (Non-negative Matrix Factorization) model. After NMF, the clusters are identified by the activation vector (the column of the coefficient matrix), which means the clusters could overlap with each other (the result of a soft clustering), the summaries are constructed by the top n images from each cluster (determined by the activation factor). The final

[1] Clusters are identified by the connected components in the similarity graph.

summary constructed in this way may have duplications, and therefore violates the conciseness requirement of summarization task. Furthermore, the summary learnt in this method are inclined to present texture patterns, which is more suitable for medical images rather than general images.

A group of iterative methods are also widely observed in collection summarization tasks [133, 134, 132]. The judging criteria of image i with regard to set S includes the quality, the coverage and the diversity of the candidate summary. The summaries are selected iteratively by maximizing Equation 1. Obviously, the judging criteria are hand crafted and lack an objective evaluation metric. Furthermore, the iterative methods only values the absolute number of appearance of the image, rather than the actual distribution of the images on the visual space. As we have illustrated in Figure 6, the dominant image do not always have to be the best summary for a collection. Therefore, a more sophisticated framework is needed to model the actual reinterpretation relationship of images on visual space.

$$i = \arg\max_{i} \{Quality(i,S) + Coverage(i,S) + Diversity(i,S)\} \tag{1}$$

Considering the drawbacks of the existing methods, we have developed dictionary learning for sparse representation framework to model the image collection summarization problem. We intend to interprets the summarization problem as a subset selection problem that a small set of the most representative images can be selected to highlight all the significant visual properties of the original image set [75]. Under this interpretation, the task for automatic image summarization can be treated as an optimization problem, e.g., selecting a small set of the most representative images that can best reconstruct the original image set in large size. If we define $X \in \mathbb{R}^{d \times n}$ as the original image set in large size and $D \in \mathbb{R}^{d \times k}, k \ll n$ as the summary of the given image set X in small size, automatic image summarization is to determine the summary D by minimizing the global reconstruction error:

$$\min_{D} ||X - f(D)||_2^2 \tag{2}$$

The selection of the reconstruction function $f(\cdot)$ is to determine how each image in the original image set X can be reconstructed by the most representative images in the summary D. In this thesis, we have defined the reconstruction function $f(\cdot)$ as a linear regression model that uses the summary D to sparsely reconstruct each image in the original set X. The sparsity means that only limited number of bases will actually be involved in the reconstruction of an image. The idea of "induced sparsity" has already been introduced in Ma's work [161], which also learns the sparse coefficients from a given data set. However, Ma's work fixes the dictionary as the original training set of a given category. In our problem, the dictionary and coefficient matrix are jointly learnt so that the coefficient learning process in [161] can only be considered as an alternative to the sparse coding stage of our proposed work.

From the above description, we now successfully reformulate the task of automatic image summarization into the problem of dictionary learning for sparse representation as shown in Equation 2. Therefore, two research issues, automatic image summarization and dictionary learning for sparse representation, are linked together according to their intrinsic coherence: both of them try to select a small set of the most representative images that can effectively and sufficiently reconstruct large amounts of images in the original image set.

We have discovered that the image collection summarization problem can be interpreted straightforwardly with the dictionary learning for sparse representation model under the SIFT BoW framework. Therefore, the summarization performance can be directly evaluated by the corresponding value of the reconstruction function. Although automatic image summarization and dictionary learning for sparse representation have intrinsic coherence, we need to clarify that they have significant differences as well, e.g., the optimization function for automatic image summarization has some unique constraints such as the fixed basis selection range, nonnegative and L_0-norm sparsity of the coefficients. The constraints are critical and differ the proposed framework from most of the existing works. For the basis learning stage, traditional methods such as MOD [34], K-SVD [2], Discriminative K-SVD [101], online dictionary learning [100], all "learn" or "update" the basis analytically, which does not restrict the search range. The sparse modeling pipelines introduced in Sapiro's work [125] propose similar sparse coefficients model, but do not have a restriction on the bases learned either. On the other hand, the summarization problem requires the bases to be chosen from a pool of given candidates, which results in a "selecting" action, rather than "learning". This observation implies the use of simulated annealing algorithm for discrete bases search, which is the most important difference between the proposed work to other works [125, 34, 2, 101, 100].

Most existing research work for automatic image summarization evaluate their summarization results subjectively by using user satisfaction and relevancy score. There

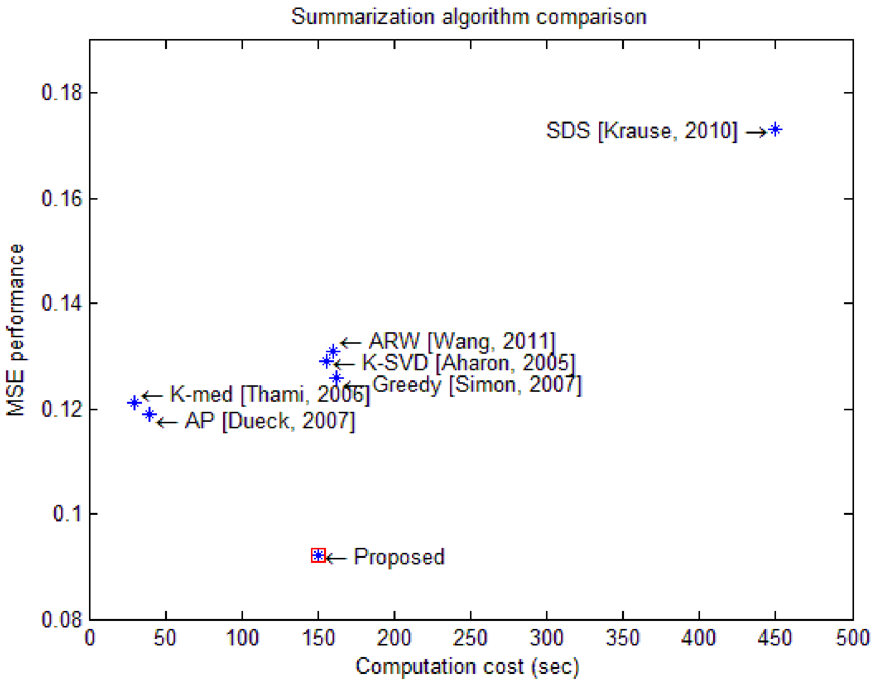


Figure 7: Summarization result comparison with 6 baseline methods in terms of both MSE performance and computation cost. The closer an algorithm is to the origin point, the better the algorithm is.

lacks an objective and quantitative evaluation metric for assessing the performance of various algorithms for automatic image summarization. By reformulating the issue of assessing the quality of summarization results as a reconstruction optimization task, we can objectively evaluate the performance of various algorithms for automatic image summarization in terms of their global reconstruction ability. In addition to the subjective evaluation, the global Mean Square Error (MSE) is defined as the objective evaluation metric to measure the performance of our proposed algorithm for automatic image summarization and compare its performance with that of other 6 baseline methods. Experiment results prove that the proposed framework achieves the best MSE performance with decent computation cost, as shown in Figure 7

1.2.5 An Information Retrieval Practice: Graffiti Retrieval

The proposed image collection exploration and navigation system can be also used to serve the information retrieval tasks. In this thesis, a graffiti image retrieval application is introduced and implemented with the proposed navigation system. Graffiti recognition and retrieval, as an application in public safety, has drawn more and more attention of researchers in the broad field of information retrieval [78]. Graffiti may appear in the form of written words, symbols, or figures, and it has sprung up in most metropolitan areas around the world. Gang-related graffiti is typically composed of mostly characters, conveys lots of information, and often identifies a specific gang territory or threatens law enforcement. The retrieval and interpretation of such information has become increasingly important to law enforcement agencies. With the prevalence of hand-held devices, digital photos of graffiti are easily acquired and enormous data collections of graffiti images are rapidly growing in size. Sifting through and understanding each image in a collection are very difficult, if not impossible, for humans to do. Thus, there is an urgent need to build a visual analytic system that can be used for automatic graffiti image recognition and retrieval from large-scale data collections.

It may seem that simply applying traditional optical character recognition (OCR) on graffiti characters would address the problem. However, because of the artistic appearance of many graffiti characters and the various types of surfaces that graffiti can be painted on, understanding graffiti characters presents many more challenges than traditional OCR can solve. As a compromise, researchers take a shortcut by

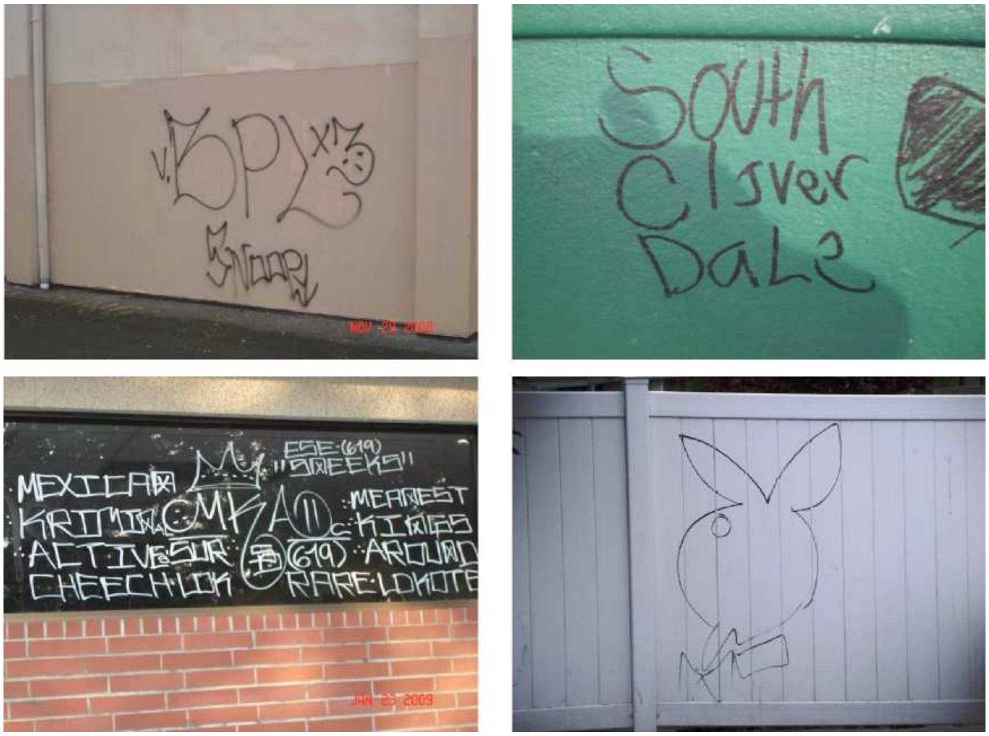


Figure 8: Sample graffiti images: Three of the four images contain text, while the bottom right image contains no textual information but only the "playboy" symbol

not utilizing any particular treatment to localize the graffiti objects in the image. Instead, traditional object localization methods are applied, for example, to extract the so-called "interesting" objects [80] or conduct the retrieval task with local feature matching on the whole image without object localization [70]. There are two primary flaws of such treatment: 1) Graffiti objects may not be "interesting" under the view of traditional object localization and 2) Textual information within the image is missing, making semantic-level understanding of the graffiti impossible. While investigating an actual graffiti image collection as shown in Figure 8, we observed that most of the collected images have textual information (people's names or locations), while some have figures or symbols that are also meaningful, such as the "playboy" and "crown" symbol (in bottom left image in Figure 8; the crown image is a well-known symbol of a gang member). These observations led us to integrate the research work of both semantic and visual understanding of the data, similar to the idea of fusing visual and textual information [13].

To best describe the research tasks of graffiti recognition and retrieval, we need to inspect the challenges and differences between graffiti recognition and traditional OCR as shown in Figure 9. The images (a) to (f) illustrate different aspects of the challenges in character detection, recognition, and image retrieval of graffiti images. Image (a) suggests that graffiti may appear on any type of surface, including walls, wooden fences, door frames, light poles, windows, or even tree trunks. The roughness and complexity of the background may bring in a lot of noise, making the task of character detection very challenging. Image (b) illustrates that graffiti usually appears outdoors and is exposed to various lighting conditions. Shadows and sunlight may dramatically affect the ability to correctly detect characters. Graffiti "words" often appear to be nonsensical because they are formed from acronyms or specially created combinations of letters as shown in image (c). In traditional OCR, the recognition result for certain letters could be used to predict the unrecognized letters by forming potential meaningful words. In graffiti recognition, we do not have such a prediction. Given that we have the ability to detect strokes of painting, we still need to further differentiate texture strokes and non-textures, such as the "playboy" symbols as shown in image (d). As illustrated in images in (e) and (f), the font and artistic writing style of characters make the same words have a very different appearance. This marked variation would impede the technique of template matching or local feature matching.

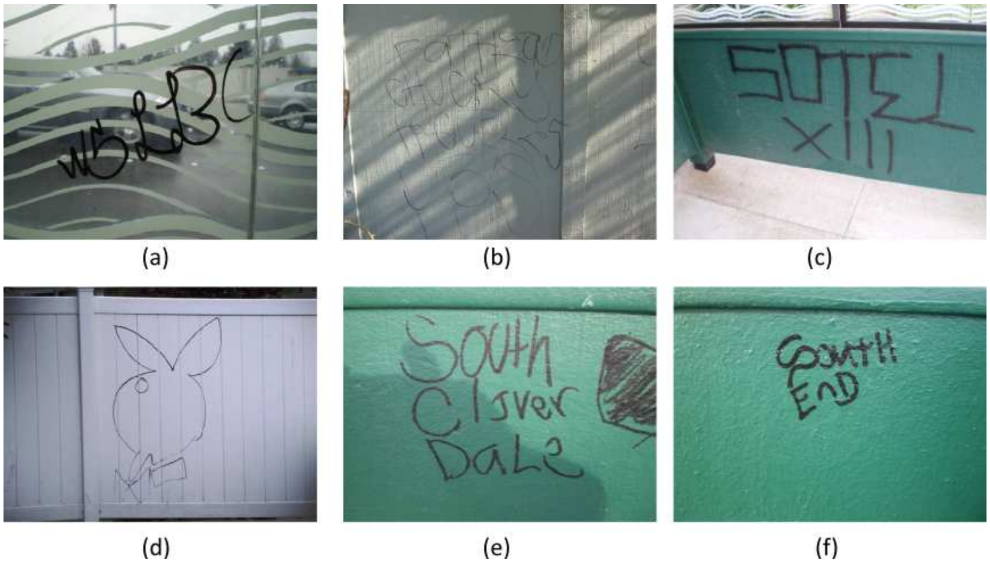


Figure 9: Example challenging graffiti images

In this application, we focus on the research task of graffiti image retrieval. After deep investigation of the challenges of the graffiti recognition task compared to OCR, we design a series of techniques for effective character detection. Next, we conduct semantic-wise and image-wise retrieval on the detected character components rather than the entire image to avoid the influence of the background noise. The visual and semantic matching scores are combined to give the final matching result.

1.3 Contribution

Our system works specially on large-scale image collections and we have made a series of contribution to fulfill large-scale image collection exploration and analysis. The contributions of this paper reside in the following aspects:

For junk image filtering:

i. A bilingual inter-cluster correlation analysis algorithm is developed for integrating bilingual image search results for automatic junk image filtering.

ii. To achieve more accurate partition of the returned images, multiple kernels are seamlessly integrated for diverse image similarity characterization and a K-way min-max cut algorithm is developed for achieving more precise image clustering.

iii. To filter out the junk images, the returned images for the same keyword-based query (which are obtained from two different keyword-based image search engines) are integrated and inter-cluster visual correlation analysis is further performed to automatically identify the clusters for the relevant images and the cluster for the junk images.

For concept network generation and image collection summarization:

i. A visual concept network structure is constructed to allow user to navigate large-scale online image collections according to their visual similarity contexts at the semantic level. Specifically, multiple kernels and kernel canonical correlation analysis are combined to characterize the diverse inter-concept visual similarity relationships more precisely in a high-dimensional multi-modal feature space.

ii. The problem of automatic image summarization is reformulated as an issue of dictionary learning for sparse representation. As a result, we can utilize the theoretical methods for sparse representation to solve the problem of automatic image summarization.

iii. A global optimization algorithm is developed to find the solution of the optimization function for automatic image summarization, which can avoid the local optimum and maintain sufficient computation efficiency.

For graffiti image retrieval:

i. A bounding box framework is designed to localize the graffiti components, which speed-up interest point matching operation and reduce false-positive matches.

1.4 Outline

The remainder of this dissertation begins with Chapter 2, which reviews the related work of the various aspects covered by this dissertation. Chapter 3 introduces a novel junk image filtering in incorporating bilingual image search results. Chapter 4 discusses about our coarse-to-fine framework to speed up near duplicate image pair detection. In Chapter 5, We proposes the design of our visual concept network and the corresponding visualization result and we will present a novel understanding of image collection summarization via dictionary learning for sparse representation in Chapter 6. In Chapter 7, we will introduce an information retrieval application of graffiti image retrieval. The evaluation and discussion of the results will be listed as the last section in each chapter. Lastly ,we will conclude the work of this thesis in Chapter 8 and also list the future research plans.

CHAPTER 2: RELATED WORK

2.1 Junk Image Filtering

Some pioneering work has been done for improving Google Images [43, 42, 12, 155, 47, 48]. To filter out the junk images from Google Images, Fergus et al. have applied constellation model to re-rank the returned images according to the appearances of the object classes and some promising results are achieved [43, 42], where both the appearance models for the distinct object parts and the geometry model for all the possible locations of the object parts are incorporated to learn the object models explicitly from a set of images. Unfortunately, this approach may seriously suffer from at least two key problems: (a) Because both the appearance models of the object parts and their spatial configuration models should be learned simultaneously, the number of model parameters are quite large and thus a fairly large amount of high-quality training images are needed to learn the complex object models reliably; (b) Image search results (returned by Google Images), on the other hand, are usually very noisy and cannot be used as a reliable source for training such complex object models precisely.

To incorporate multi-modal information for junk image filtering, the research team from Microsoft Research Asia have developed several approaches to achieve more effective clustering of web image search results by using visual, textual and linkage information [12, 155, 47]. Instead of treating the associated text terms as the single source for web image indexing and retrieval, they have incorporated multi-modal information sources to explore the mutual reinforcement between the images and their associated text terms. In addition, a tri-parties graph is generated to model the linkage relationships among the low-level visual features, images and their associated text terms. Thus automatic image clustering is achieved by supporting tri-parties graph partition. Incorporating multi-modal information sources for image indexing and retrieval may improve the performance of web image search engines significantly, but such approach may seriously suffer from the following problems: (a) Because only the global visual features are extracted for characterizing the visual properties of the images, the accuracy of the underlying image clustering algorithms may be low, especially when the image objects are salient for image semantics interpretation; (b) The tri-parties linkage graph may be very complex, and thus it may be very hard to achieve junk image filtering in real time.

Because the interpretations of the relevance between the returned images and the users' real query intentions are largely user-dependent, it is very important to integrate human experience and their powerful capabilities on pattern recognition for enhancing image semantics interpretation and web image retrieval. Thus one potential solution for junk image filtering is to involve users in the loop of image retrieval via relevance feedback, and many relevance feedback techniques have been proposed in the past [60, 144, 123, 142, 141, 177]. Another shortcoming for Google Images search engine is that the underlying techniques for query result display (i.e., page-by-page ranked list of the returned images) cannot allow users to assess the relevance between the returned images and their real query intentions effectively. Many pages are needed for displaying large amounts of returned images, thus it is very tedious for users to look for some particular images of interest through page-by-page browsing. Things may become worse when the ambiguous keywords with many potential word senses are used for query formulation. Because the visual properties of the returned images are completely ignored for image ranking, the returned images with similar visual properties may be separated into different pages. Ideally, users would like to have a good global overview of the returned images in a way that can reflect their principal visual properties effectively and allow them to navigate large amounts of returned images interactively according to their nonlinear visual similarity contexts, so that they can assess the relevance between the returned images and their real query intentions interactively. Based on this observation, Gao et al. have developed an interactive approach for junk image filtering [48], where users' feedbacks are seamlessly integrated for hypotheses assessment and junk image filtering. Unfortunately, naive users may not be willing to spend too much time in such an interactive approach for junk image filtering.

Feng et al. [41] and Weston et al. [158] have proposed new text-based approaches for web image representation by using the associated text terms rather than only the captions. Such text-based web image representation approaches may significantly improve the performance of image search engines. However, the text information is not always available and the above techniques have their limitation in the scope of application.

Using template matching techniques can easily detect and remove a special type of images, such as fabricated images or generated images. Wang [157] and Nhung[109] compare the incoming images with the spam image database. The database stores all feature vectors extracted from all known spam images labeled by traditional anti-spam filter. The problem is that such a database may not be sufficient for spam detection because the number of templates cannot be matched to the number of spam images that exist.

2.2 Near Duplicate Detection

Many automatic techniques have been developed for near duplicate image detection [69, 173] and discussed in review works [93, 122, 151]. These existing techniques can be categorized into two representative groups according to their focus of the visual features to be used for near duplicate image detection: global approach and local approach. Nonetheless, if available, heterogeneous features like EXIF data, is also helpful [140] in early stage of partitioning the data set. For most of the other circumstances, such meta data is not available. Therefore, we will only discuss about the visual features that are extracted solely from the content of the image. The global approach focuses on the rapid identification of similar images by using the global visual features, which can effectively handle the images with visually-close objects or background. Global features include color histogram, texture feature, and varieties such as Colored Pattern Appearance Model (CPAM) [139]. The local approach focuses on extracting more reliable local visual features and performing pair-wise image matching via interest point matching, which can effectively detect the near-duplicate images containing similar objects of interest with various backgrounds.

After detecting the local interest points or regions, various features and descriptors can be used such as Ordinal Spatial Intensity Distribution (OSID) descriptor [140], SIFT [74, 173] descriptor, GLOH, and PCA-SIFT [79]. Recent surveys [104] have conducted an extensive comparison between the global approach and the local approach. Local feature is proved to be more accurate on duplicate pair matching tasks although also found to be more computational expensive than global feature.

For global approaches, a straightforward comparison of two images is to calculate the similarity between two feature vectors. Specifically, if two images are near-duplicates, their feature vectors would be very close to each other in the feature vector space [157] . For local approaches, the local interest point descriptors could be used in different ways to service the design of similarity measurement. For simple key point matching, the matching between bags of features is usually done via naive bipartite graph matching [140, 79, 95] to get the initial candidate set of matches.

In [103], Gaussian Mixture Model is used to represent the visual signature of the image. Using JS-divergence as a similarity measurement of two distributions, Mehta et al. performed Labeled Agglomerative Clustering: the idea is to find images which are similar enough in the training set, and to replace them with one signature. Query image from the test set can now be compared to the already identified signatures and if there is a close match, then a positive detection can be made.

Zhang [173] use a parts-based representation of each scene image by building Attributed Relational Graphs (ARG) between interest points. They then compare the similarity of two images by using Stochastic Attributed Relational Graph Matching, to give impressive matching results. Unfortunately, they only demonstrate their method on a few hundred images and do not discuss any way to scale their system to larger data sets of images.

In [166], a new framework, termed spatially aligned pyramid matching, is proposed for near duplicate image detection. Images are divided into both overlapped and non-overlapped blocks over multiple levels. In the first matching stage, pair-wise distances between blocks from the examined image pair are computed using SIFT features and Earth Movers Distance (EMD). In the second stage, multiple alignment hypotheses that consider piecewise spatial shifts and scale variation are postulated and resolved using integer-flow EMD. As shown in many pyramid matching works [54, 87, 91, 167], the best results can be achieved by combining the result from multiple resolutions. The fusion formulation is often found as below or in a similar formation:

$$S^{Fuse}(x \to y) = h_0 S(x^0 \to y^0) + \sum_{l=1}^{L-1} h_l S(x^{2l-1} \to y^{2l}) \tag{3}$$

where h_l is the weight for the level-l. There could be two different weighting schemes: a) equal weights, and b) unequal weights. Unequal weighting scheme is often preferable, with the weight proportional to the scale of the block.

Another representation model is the visual bag-of-word model [18, 116, 110, 135]. The idea is borrowed from text retrieval system, and has achieved great success in image retrieval community. Images are scanned for “salient” regions or interest points and a high-dimensional descriptor is computed for the specified region. These descriptors are then clustered into a vocabulary of visual words, and each region or interest point is mapped to the visual word closest to it in the vocabulary. An image is then represented as a bag of visual words, and these are entered into an index for later querying and retrieval. Typically, no spatial information of the visual words is considered in the retrieval task. To overcome the limitation of the bag-of-word model which ignore spatial relationships among visual words, spatial constraints are frequently applied afterwards which will dramatically improve the matching accuracy. Such as in [165], Wu et al. proposed bundled feature where image features are bundled into local groups. Each group of bundled features becomes much more discriminative than a single feature, and within each group simple and robust spatial constraints can be enforced. Other techniques such as RANSAC [44] and its variation LO-RANSAC used in [116] are also widely acknowledged, although it is computationally expensive and would not be suitable for large-scale applications.

By seamlessly integrating content and context, Wu et al. have recently developed an interesting approach for near-duplicate video detection [164]. The proposed approach considers the time duration information to avoid comparison between videos with considerably different length; then compare the local points extracted from thumbnail images. In the case of image duplicate detection, there is no such information as time duration, which implies us to use some computationally cheap image information.

Min-Hash is another straightforward choice for large-scale duplicate detection. It is a method originally developed for text near-duplicate detection, and is adopted to near-duplicate detection of images. Chum et al. [18] proposed method uses a visual vocabulary of vector quantized local feature descriptors (SIFT) and for retrieval exploits enhanced min-Hash techniques. The method represents the image by a sparse set of visual words. Similarity is measured by the set overlap (the ratio of sizes between the intersection and the union).

To support more effective detection of the near duplicate images from large-scale image collections, Locality Sensitive Hashing(LSH) is widely used for image database indexing and achieving fast approximate search [79, 171, 67, 145, 71, 45]. In [145] Torralba et al. proposed to learn short descriptors to retrieve similar images from a huge database. The method is based on a dense 128D global image descriptor, which limits the approach to no geometric / viewpoint invariance. Foo et al. [45] use LSH index scheme to index a set of 36-dimensional PCA-SIFT descriptors. Jain et al [71] introduced a method for efficient extension of LSH scheme, and with Mahalanobis distance used in [66] and L1 distance used in [171]. All aforementioned approaches use bit strings as a fingerprint of the image. In such a representation, direct collision of similar images in a single bin of the hashing table is unlikely and a search over multiple bins has to be performed. This is feasible for approximate nearest neighbor or range search when the query example is given. However, for clustering tasks (such as finding all groups of near duplicated images in the database) the bit string representation is less suitable.

Another type of high dimensional indexing scheme is the tree-structured indexing scheme which include: kd-tree, priority kd-tree [159] and R-tree [57] dimension reduction. Nister and Stewenius [110] propose generating a “vocabulary tree” using a hierarchical k-means clustering scheme (also called tree structured vector quantization [49]). The hierarchical k-means clustering scheme is used in [116] and also in our work, for its ability to handle large scale of data and quick response to retrieve approximate nearest neighbor. In [174], Zhang and Zhong proposed using self-organization map (SOM) neural nets as the tool for constructing the tree indexing structure in image retrieval. The advantages of using SOM were its unsupervised learning ability, dynamic clustering nature, and the potential of supporting arbitrary similarity measures.

When large-scale web images come into view, the tremendous volume of web images may pose new challenges to the scalability of all these existing algorithms for automatic near duplicate image detection, e.g., the significant changes on the volume of web images may dramatically affect the performance of all these existing techniques for automatic near duplicate image detection. To deal with large-scale image collection, one straightforward idea is to roughly partition the data set without separating the possible duplicate pairs. If such meta data is not available, the images will be organized in categories by their query keywords, and the scale of images under each category will be reduced to several thousands. If we seek to further down scale the category with visual features, we have to make sure the computationally cheap features are used in the initial stages, and relatively expensive features used in later stages [140, 178]. Based on this idea, Tang proposed a computation-sensitive cascade framework in [140] to combine stage classifiers trained on different feature spaces with different computation cost. This method can quickly accept easily identified duplicates using only cheap features, such as simple global feature, without the need to extract more sophisticate but expensive ones. Specifically, Tang bootstraps the training data and the stage classifiers are trained on progressively more expensive, yet more powerful feature spaces. Zhu et al. [178] suggest an effective multi-level ranking scheme that filters out the irrelevant results in a coarse-to-fine manner. The first two ranking levels (Nearest Neighbor Ranking, Semi-Supervised Ranking) are based on global features (Grid Color Moment, Local Binary Pattern, Gabor Wavelet Texture and Edge) for efficiently filtering out the irrelevant results, and the last level (Nonrigid Image Matching) provides a fine re-ranking based on the local features (SURF [8]).

2.3 Concept Network Visualization

For large-scale image collection organized in concepts, an essential issue is to build a concept network, which visualize the relationship between concepts. The organization and visualization of such concept network includes the following operations: multi-modal feature extraction, concept network generation and interactive navigation system design.

For multi-modal feature extraction issues, different frameworks have been proposed for image content representation [154, 176, 90, 97, 147, 35] and multiple types of visual features [98, 112, 118, 175, 95] (such as colors, textures and interest points) are extracted for image representation. Because many image content representation frameworks and many different visual features exist, there is an urgent need to provide a benchmark of their discrimination power for object and concept detection (i.e., image classification) [149, 56]. Snoek and his team have done a wonderful benchmark work for color features [149], but they did not consider other types of visual features which are widely used for image classifier training. When multiple types of visual features are used for image classifier training, it is also very important to benchmark the relative importance between different types of visual features for the same image classification task, which may further provide good solution for feature subset selection.

When multiple types of visual features (with high dimensions) are used for SVM image classifier training, the statistical properties of the images in such high-dimensional multi-modal feature space could be heterogeneous, which should be seriously considered in the procedure for kernel design and selection [36]. Multiple types of kernel functions, such as linear kernel and RBF kernel, are available for image similarity characterization. It is worth noting that different kernels may be suitable for different image categories (i.e., images with different statistical properties in the high-dimensional multi-modal feature space). Unfortunately, most existing SVM classifier training tools use only one kernel for diverse image similarity characterization and fully ignore the heterogeneity of the statistical properties of the images in the high-dimensional multi-modal feature space [36]. Obviously, it is very attractive to design different kernels for characterizing the diverse similarity relationships between the images under different feature spaces and combine multiple types of kernels for SVM image classifier training. When multiple types of kernels are used for diverse image similarity characterization, it is also very important to provide a benchmark of multiple approaches for kernel combination for the same image classifier task. Based on these observations, we have developed a novel benchmark framework to evaluate the performance of multiple types of visual features and kernels for SVM image classifier training.

Many work has been done to construct frameworks for image summarization and interactive image navigation and exploration [136, 59]. The project of Large-Scale Concept Ontology for Multimedia (LSCOM) is the first one of such kind of efforts to facilitate more effective end-user access of large-scale image/video collections in a large semantic space [11, 107]. By exploiting large amounts of image/video concepts and their inter-concept similarity relationships for image/video knowledge representation and summarization, concept ontology can be used to navigate and explore large-scale image/video collections at the concept level according to the hierarchical inter-concept relationships such as “IS-A” and “part-of” [107].

Concept ontology may also play an important role in learning more reliable classifiers for bridging the semantic gap [37, 65, 150, 88, 40, 5, 108]. By exploiting only the hierarchical inter-concept or inter-object similarity contexts, some pioneer work have been done recently to integrate the concept ontology and multi-task learning for improving image classifier training, and the concept ontology can be used to determine the inter-related learning tasks more precisely [37, 146].

Chen et al. [15] use hierarchical tree-structures to both speed-up search-by-query and organize databases for effective browsing. Chen presents a method for designing a hierarchical browsing environment which is called similarity pyramid. The similarity pyramid groups similar images together while allowing users to view the database at varying levels of resolution. Graham et al [53] developed two browsers: Calendar browser and Hierarchy browser, which takes advantage of photo time stamps and meta data information respectively. The photos in Hierarchy browser are organized in a cluster tree.

Because of the following issues, most existing techniques for concept ontology construction may not be able to support effective navigation and exploration of large-scale image collections: (a) Only the hierarchical inter-concept relationships are exploited for concept ontology construction [19, 138]. When large-scale online image collections come into view, the inter-concept similarity relationships could be more complex than the hierarchical ones (i.e., concept network) [162]. (b) Only the inter-concept semantic relationships are exploited for concept ontology construction [19, 138], thus the concept ontology cannot allow users to navigate large-scale online image collections according to their visual similarity contexts at the semantic level. It is well-accepted that the visual properties of the images are very important for users to search for images [136, 59, 11, 107, 162]. Thus it is very attractive to develop new algorithm for visual concept network generation, which is able to exploit more precise inter-concept visual similarity contexts for image summarization and exploration.

For concept network visualization, the relationship of the concept nodes is characterized by a similarity matrix and Multi-Dimensional Scaling [82] can be used to project data from high dimensional space onto lower dimensional space, such as 2D space. The distance on the original space is kept as much as possible on the projected 2D space. Recently, several researches have applied multidimensional scaling (MDS) to database browsing by mapping images onto a two dimensional plane. MacCuish et al. [99] used MDS to organize images returned by queries while Rubner et al. [121] used MDS for direct organization of a database. Stan et al. [137] buld the *eID* system which utilized MDS to display the image for image collection exploration. After MDS projection, the user may be able to view the relationship between concept nodes on 2D plane. Considering we are dealing with large-scale image collection, the produced concept network may have thousands of nodes and are still cluttered for clear exploration, therefore, we employ the traditional visualization technique of FishEye

[46, 114] to highlight the center parts and overlook the unfocused parts. Specifically, the 2D concept network is further plotted to 3D a revolving sphere so that the center part is always enlarged and displayed with a high resolution, while the nodes on the border of the sphere is contracted and displays less in detail. FishEye technique is believed to be suitable for network topologies and extremely large structures. The focus+context [7] technique is an extension to FishEye which displays a selected region with high resolution in separate window. A display in separate windows is not necessary for concept network visualization, which has evenly distribution of concept node and doesn't provide much detail information for display.

2.4 Image Collection Summarization

We want to emphasis once again the definition of "summarization" used in this thesis that it is the outcome from the down sampling action. There is another type of work also tagged as image summarization, known as "digital tapestry" or "picture collage". For such problems, the set of images has already been decided , and the visual layout to be determined. The collection summarization problem, on the other hand, focus on the summarization selection issue only.

There is one category of image collections utilizes tag information, rather than visual information: Clough et al.[21] construct a hierarchy of images using only textual caption data, and the concept of subsumption. A tag t_i subsumes another tag t_j if the set of images tagged with t_i is a superset of the set of images tagged with t_j. Schmitz [126] uses a similar approach but relies on Flickr tags, which are typically noisier and less informative than the captions. Jaffe et al. [68] summarize a set of images using only tags and geotags. By detecting correlations between tags and geotags, they are able to produce “tag maps”, where tags and related images are overlaid on a geographic map at a scale corresponding to the range over which the tag commonly appears. For a larger range of general images, the tag information is sometimes unavailable, thus the summarization is extracted based on visual information of the images in the collection.

Most existing algorithms for automatic image collection summarization can be classified into two categories: (a) simultaneous summarization approach; and (b) iterative summarization approach.

For the simultaneous summarization approach, the global distribution of an image set is investigated and image clustering techniques are usually involved [27, 68]. In particular, Jaffe et al. [68] have developed a Hungarian clustering method by generating a hierarchical cluster structure and ranking the candidates according to their relevance scores. Denton et al.[27] have introduced the Bounded Canonical Set (BCS) by using a semidefinite programming relaxation to select the candidates, where a normalized-cut method is used for minimizing the similarity within BCS while maximizing the similarity from BCS to the rest of the image set. Other clustering techniques such as k-medoids [168], affinity propagation [3] and SOM [26] are also widely acknowledged. The global distribution of an image set can also be characterized by using a graphical model. Jing et al. [75] have expressed the image similarity contexts with a graph structure, where the nodes represent the images and the edges indicate their similarity contexts, finally, the nodes (images) with the most connected edges are selected as the summary of a given image set. Chen et al.’s work [15] focuses on the use of hierarchical tree-structures to both speed-up search-by-query and organize databases for effective browsing. The hierarchical structure is built with agglomerate (bottom-up) clustering. The specific icon image (or summary of the cluster) is chosen to minimize the distance to the corresponding cluster centroid. Stan et al.'s summarization work [137] also use hierarchical structure with k-means algorithm and select the centroid (center of the cluster) as the most representative image. Camargo et al. [14] develop the clustering framework with NMF and selects the most important image in each cluster based on the text terms.

For the iterative summarization approach, some greedy-fashion algorithms are applied to select the best summary sequentially until a pre-set number of the most representative images are picked out [133]. Simon et al. [133] have used a greedy method to select the best candidates by investigating the weighted combinations of some important summarization metrics such as likelihood, coverage and orthogonality. Sinha [134] proposed a similar algorithm with the metrics of quality, diversity and coverage. Shroff et al.'s work [132] introduce an optimization function linearly composed by the coverage term and diversity term. The basis is updated by selecting randomly from the rest of the set and only those bases which strictly decrease the objective will be remained. Fan et al. [38] proposed " JustClick" system for image recommendation and summarization which incorporates both visual distribution of the images and user intention discovered during exploration. Wong et al. [153] integrated the dynamic absorbing random walk method to find diversified representatives. The idea is to use the absorbing states to drag down the stationary probabilities of the nearby items to encourage the diversity, where the item with the highest stationary probability in the current iteration is selected. The above greedy methods focus on selecting the current most representative images at each iteration while penalizing the co-occurrence of the similar candidates (images). Our proposed model for automatic image summarization takes the benefit of both two types of approaches, e.g., we use the explicit measurements in the iterative approaches to characterize the property of a summary and we learn the bases (candidates) simultaneously to avoid the possible local optimum solution. He et al. [61] developed a unified framework for structural analysis of image database using spectral techniques, which follows similar idea as Wong's work [153] by using random walk and seek for stable distribution.

Recently, Krause et al. [81] proposed the submodular dictionary selection method for sparse representation and have proved that the dictionary (which is selected greedily) is close to the global optimum solution in the case that the original data set satisfies the submodular condition. However, most of the real-world image sets do not satisfy the submodular condition which makes Krause's algorithm less convincing for automatic image summarization application and corresponding results do not guarantee to be global optimum.

Most existing techniques for dictionary learning and sparse coding use machine learning techniques to obtain more compact representations, such as PCA [160], the Method of Optimal Direction (MOD) [33] and K-SVD [1]. The MOD algorithm is derived directly from Generalized Lloyd Algorithm (GLA) [49], which iteratively updates the codebook and the code words are updated as the centroids from a nearest neighbor clustering result. The K-SVD algorithm follows the same style by updating the bases iteratively and the new basis is generated directly from the SVD calculation result. The K-SVD method is not applicable to our proposed approach for automatic image summarization because our model only takes discrete bases rather than numerical outputs from SVD. The methods of Matching Pursuit (MP) [102] and Lasso (forward stepwise regression and least angle regression) [32] are widely accepted for sparse coding. These methods could provide us with some ideas on the design of an appropriate sparse coding algorithm.

The sparse representation of images is not rarely seen [170, 161, 125]. Wright et al. [161] proposed a sparse representation model used in face recognition application. The representation of images is similar to the model used in this thesis, however, Wright's work does not serves specifically for the summarization task, therefore, the bases do not process a practical meaning as the coefficients do not have to be non-negative. The non-negativity constraint is necessary for the summarization task and makes the problem more challenging. The sparse representation-based classification (SRC) algorithm proposed in [161] is barely the sparse coding step and it is targeted on dictionary learning tasks. The sparse representation of images is also used in classification applications such as in Sapiro's work [125]. The dictionary learning and sparse coding are implemented in a similar way as K-SVD. Our previous work [170] also proposed a L0-norm sparse representation of the images. The major difference is that optimal solution is searched in a greedy fashion and cannot avoid local optimum. In this thesis, a simulated annealing algorithm is adopted as the proposed approach to achieve global optimum with a high probability when enough search steps are performed. A summarization of the frameworks and algorithms discussed in this subsection can be found in Table 1 and 2

Works	Similarity metric	Content representation model	Feature/descriptor
Camargo et al. [14]	Euclidean distance	Bag-of-Word	DCT
Stan et al. [137]	Block similarity [128]	Block	Color-WISE [128]
Simon et al. [133]	Cosine distance	Interest point matching	SIFT
Chen et al. [15]	Manhattan distance	Entire image	Color, Texture, Edge
Denton et al. [27]	Earth Mover's distance	Many to many matching	Silhouettes
Jaffe et al. [68]	Euclidean distance (geographic)	N/A	Location
Jing et al.[75]	Euclidean distance	Interesting point matching	SIFT
Sinha [134]	Jiang-Conrath distance [72]	Entire image	Color
Shroff et al. [132]	Euclidean distance	Bag-of-Word	Descriptor [29]
Wang et al. [153]	Local scaled visual similarity	Entire image	Color, Shape, Texture
Yang et al. [169]	Euclidean	Entire image	SIFT
Proposed	Euclidean distance	Bag-of-Word	SIFT

Table 1: Summarization frameworks comparison: in terms of Content, Similarity Metric, Content Representation Model and Feature/Descriptor.

Works	Objective	Algorithm
Camargo et al. [14]	Most important image of each cluster	NMF(Non-negative Matrix Factorization) [129]
Stan et al. [137]	Clustroid: center of the cluster	Hierarcical clustering with K-means [31]
Simon et al. [133]	Likelihood + Coverage + Orthogonality	Greedy
Chen et al. [15]	Clustroid: center of the cluster	Agglomerated clustering (tree-structure)
Denton et al. [27]	BCS (bounded canonical set)	Max-cut [51]
Jaffe et al. [68]	Bottom up ranking	Hierarchical Hungarian clustering [52]
Jing et al.[75]	Most heavily connected node in the cluster	Similarity graph (connected component)
Sinha [134]	Quality + Diversity + Converge	Greedy
Shroff et al. [132]	Coverage + Diversity	Generalized EM (like k-means)
Wang et al. [153]	ARW (Absorbing Random Walk)	Greedy
Yang et al. [169]	Relevancy + Orthogonality + Uniformity	Affinity Propagation + Greedy
Proposed	Dictionary learning(Quality + Diversity)	Simulated Annealing

Table 2: Summarization frameworks comparison Cont'd: in terms of Objective and Algorithm used.

2.5 Graffiti Retrieval

Graffiti image retrieval research lies in the intersection of OCR and image retrieval. The techniques from both fields may benefit the graffiti retrieval work.

Graffiti image retrieval is closely related to the handwriting recognition and retrieval work in OCR. The graffiti characters are essentially handwritten characters, although they often have an artistic appearance and are usually found in more challenging environments. OCR techniques recognize and match characters based on their shape and structure information, such as skeleton feature [148, 115], shape context [9], and order structure invariance [20]. There are also a bunch of unsupervised feature learning architectures, such as the deep learning architectures of Multi-Layer Perceptions (MLP) and Stacked Denoising Auto-encoder (SDA) which achieves multi-level of representation [10, 22]. After all, the foundation for the effectiveness of these techniques is the correct separation of the characters from strings or words detected. The encoding of the word is not an easy task, and the methods available are often trivial and may not apply to the graffiti data. Another issue is that simply measuring the similarity between two individual characters as designed in [148] is inadequate. We intend to evaluate the similarity between two strings or words to derive semantic-level understanding. The proposed evaluation metric, longest common subsequence (LCS), is designed to overcome this flaw by considering the sequence of the characters in the string [172].

The visual difficulties introduced in Chapter 1 and the artistic appearance of graffiti images have motivated researchers to try routes other than OCR. Jain et al. [70]

have proposed a system named Graffiti-ID and treats the graffiti purely as images on the retrieval task. The Graffiti-ID system does not specifically locate the character components in the images, and thus some false-positive matches may occur. Furthermore, the potential semantic relationship between the graffiti characters is completely ignored; thus, Graffiti-ID does not distinguish itself from general image retrieval frameworks.

Our proposed system works on the graffiti character components that are detected in the image. Our proposed framework may have the potential to not only achieve better retrieval accuracy by eliminating as much background noise as possible, but also significantly reduce the computation burden by eliminating unrelated interest points.

CHAPTER 3: JUNK IMAGE FILTERING WITH BILINGUAL SEARCH RESULT

In this Chapter, we will introduce the junk image filtering framework, which utilize the bilingual search results from Google Image and Baidu Image. The efficient elimination of junk images or irrelevant images of a given data collection is an essential step for further operation of image collection organization and exploration.

3.1 Data Collection and Feature Extraction

The images used in this research task and the corresponding experiment are partly from Caltech-256 [55], LabelMe [124], NUS-WIDE [17], Event Dataset [89] and partly crawled from the Internet. To determine the meaningful text terms for crawling images from the internet like Google or Flickr, many people use the keywords which are sampled from WordNet. Unfortunately, most of the keywords on WordNet may not be meaningful for image concept interpretation. Based on this understanding, we have developed a taxonomy for nature objects and scenes interpretation. Thus we follow this pre-defined taxonomy to determine the meaningful keywords for image crawling as shown in Figure 10.

For feature extraction, to avoid the issue of image segmentation while allow the visual features to provide the object information at certain accuracy level, four grid resolutions are used for image partition and feature extraction which is to partition the image into a 4 by 4 mesh. In order to characterize various visual properties of the

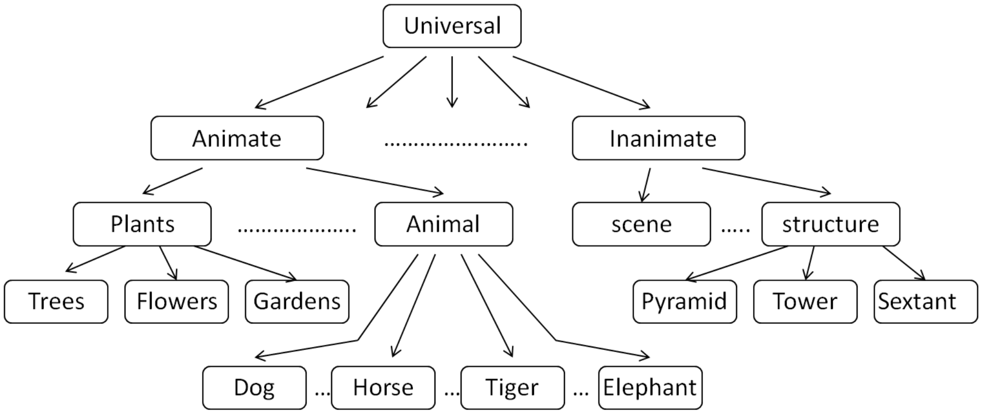


Figure 10: The taxonomy for text term determination for image crawling.

images more sufficiently, three types of visual features are extracted for image content representation: (a) grid-based color histograms [92]; (b) Gabor texture features; and (c) SURF (Speeded Up Robust Feature) features [73, 118, 96, 8].

For the color features, one color histogram is extracted from each image grid, thus there are $\sum_{r=0}^{3} 2^r \times 2^r = 85$ grid-based color histograms. Each grid-based color histogram consists of 72 HSV bins to represent the color distributions in the corresponding image grid.

To extract Gabor texture features, a Gabor filter bank, with 3 scales and 4 orientations, is used. The Gabor filter is generated by using a Gabor function class. To apply Gabor filters on an image, we need to calculate the convolutions of the filters and the image. We transform both the filters and the image into the frequency domain to get the products and then transform them back to the space domain. This process can calculate Gabor filtered image more efficiently. Finally, the mean values and standard deviations are calculated from 12 filtered images, making up to 24-dimensional Gabor texture features.

SURF algorithm is used to reduce the computational cost for traditional SIFT feature extraction [96]. For each image, a number of interest points are detected and their corresponding 64-dimensional descriptors are extracted.

One major advantage of our fast feature extraction approach is that it can achieve a good trade-off between the effectiveness for image content representation (i.e., characterizing both the global visual properties of the entire images and the local visual properties of the image objects) and the significant reduction of the computational cost for feature extraction, thus it can be performed in real time. It is also important to note that our color histograms focus on extracting the regional visual properties of the image objects for achieving more accurate image clustering by reducing the misleading effects of the background on image similarity characterization at the object level.

By using high-dimensional multi-modal visual features (color histogram, Gabor wavelet textures, SURF features) for image content representation, it is able for us to characterize the diverse visual properties of the images more sufficiently. Because each type of visual features is used to characterize one certain type of the visual properties of the images, the visual similarity contexts between the images are more homogeneous and can be approximated more precisely by using one particular type of the base kernels. Thus one specific base kernel is constructed for each type of visual features (i.e., one certain feature subset).

For two images u and v, their color similarity relationship can be defined as:

$$\kappa_c(u,v) = e^{-d_c(u,v)/\sigma_c}, d_c(u,v) = \sum_{r=0}^{R-1} \frac{1}{2^r \times 2^r} D_r(u,v) \tag{4}$$

where σ_c is the mean value of $d_c(u, v)$, $R = 4$ is the total number of grid resolutions for image partition, $D_r(u, v)$ is the color similarity relationship between two images u and v according to their grid-based color histograms at the rth resolution.

$$D_r(u, v) = \sum_{t=1}^{2^r \times 2^r} \sum_{i=1}^{72} D(H_i^t(u), H_i^t(v)) \tag{5}$$

where $H_i^r(u)$ and $H_i^r(v)$ are the ith component of the grid-based color histograms for the images u and v at the rth image partition resolution.

For two images u and v, their local similarity relationship can be defined as:

$$\kappa_s(u, v) = e^{-d_s(u,v)/\sigma_s} \tag{6}$$

$$d_s(u, v) = \frac{\sum_i \sum_j \omega_i(u)\omega_j(v)ED(s_i(u), s_j(v))}{\sum_i \sum_j \omega_i(u)\omega_j(v)} \tag{7}$$

where σ_s is the mean value of $d_s(u, v)$ in our test images, ω_i and ω_j are the Hessian values of the ith and jth interesting points for the images u and v (i.e., the importance of the ith and jth interesting points, $ED(s_i(u), s_j(v))$ is the Euclidean distance between two SURF descriptors.

For two images u and v, their textural similarity relationship can be defined as:

$$\kappa_t(u, v) = e^{-d_t(u,v)/\sigma_t}, \;\; d_t(u, v) = \sum_{r=0}^{R-1} \frac{1}{2^r \times 2^r} \sum_{t=1}^{2^r \times 2^r} ED(g_i^t(u), g_j^t(v)) \tag{8}$$

where σ_t is the mean value of $d_t(u, v)$ in our test, $ED(g_i(u), g_j(v))$ is the Euclidean distance between two Gabor textural descriptors.

The diverse similarity contexts between the images can be characterized more pre-

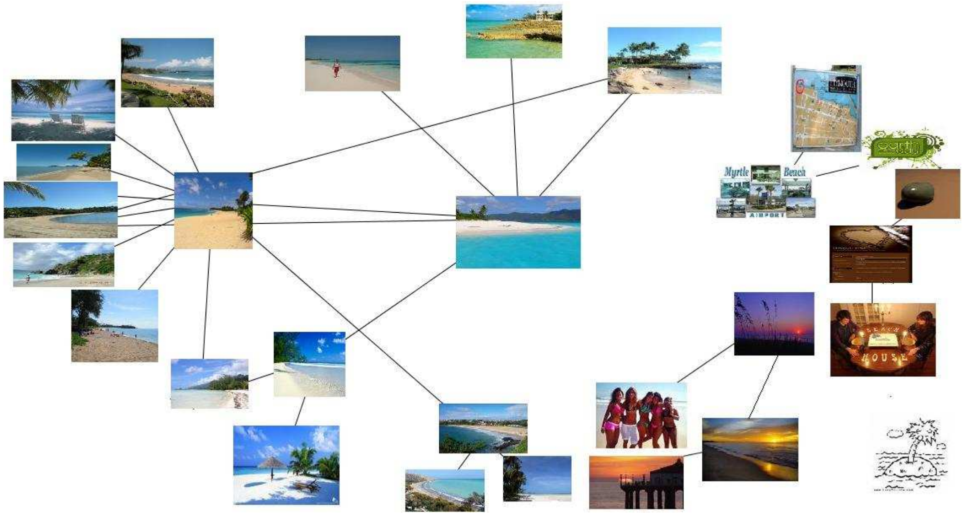


Figure 11: Image clusters and their inter-cluster correlations for the returned images of the keyword-based query “beach” from Google Images.

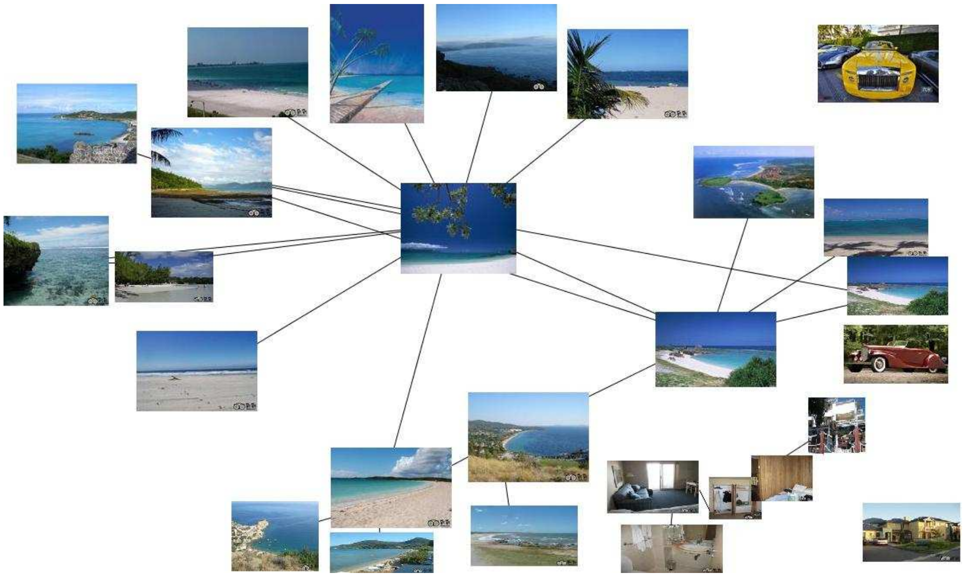


Figure 12: Image clusters and their inter-cluster correlations for the returned images of the keyword-based query “beach” from Baidu Images.

cisely by using a mixture of these three base image kernels (i.e., mixture-of-kernels) [175, 39].

$$\kappa(u, v) = \sum_{i=1}^{3} \beta_i \kappa_i(u, v), \qquad \sum_{i=1}^{3} \beta_i = 1 \tag{9}$$

where u and v are two images, $\beta_i \geq 0$ is the importance factor for the ith base kernel $\kappa_i(u, v)$.

Combining multiple base kernels can allow us to achieve more precise characterization of the diverse visual similarity contexts between the images. However, estimating the kernel weights in unsupervised learning (clustering) scenarios is a hard problem, due to the absence of class labels that would guide the search for the relevant information.

3.2 Image Clustering

The relevant images for the same keyword-based query, which are returned by different keyword-based image search engines, may have strong correlations on their visual properties. The image search results from one search engine may also has such properties as can be grouped into several related partitions. Thus image clustering is very attractive for improving the performance of the image retrieval systems [4, 74, 94, 119, 16]. The objective of image clustering is to provide clusters of images within each query set and the result clusters will be further evaluated with other clusters for their visual correlation. To achieve more effective image clustering and automatic kernel weight determination, a K-way min-max cut algorithm is developed, where the cumulative inter-cluster visual similarity contexts are minimized while the cumulative intra-cluster visual similarity contexts (summation of pair wise image

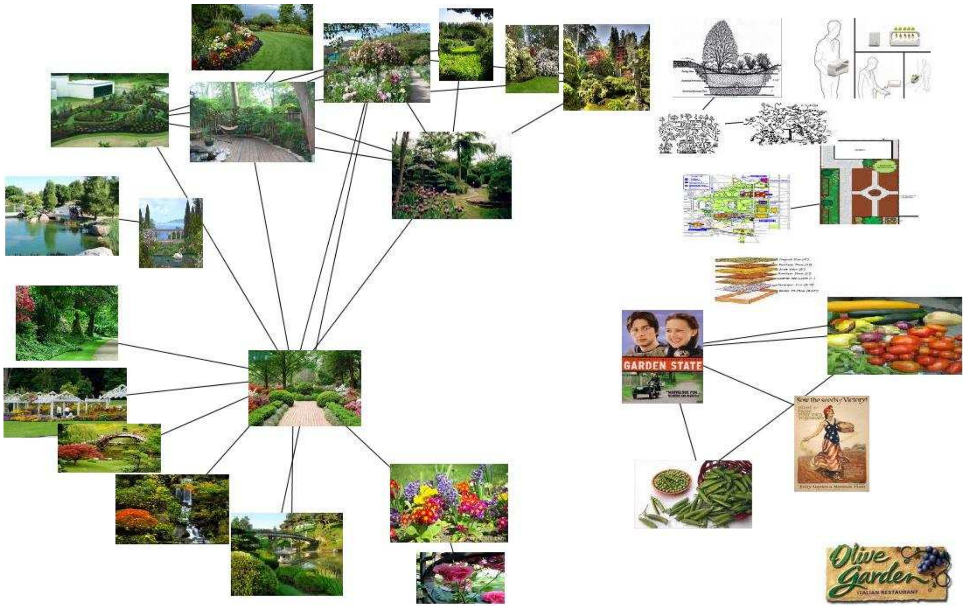


Figure 13: Image clusters and their inter-cluster correlations for the returned images of the keyword-based query "garden" from Google Images.

similarity contexts within a cluster) are maximized. These two criteria can be satisfied simultaneously with a simple K-way min-max cut function.

Our K-way min-max cut algorithm takes the following steps iteratively for image clustering and kernel weight determination:

(a) For a given keyword-based query C, a graph is first constructed to organize all its return images (which are obtained from one certain keyword-based image search engine) according to their visual similarity contexts [131, 28], where each node on the graph is one return image for the given keyword-based query C (from the same keyword-based image search engine) and an edge between two nodes is used to characterize the visual similarity contexts between two return images, $\kappa(\cdot,\cdot)$. An initial value for the number of image cluster is given as $K = 120$ and the kernel weights for three base kernels are set to be equal, e.g., $\beta_1 = \beta_2 = \beta_3 = 0.33$.

(b) All these returned images for the given keyword-based query C (from the same

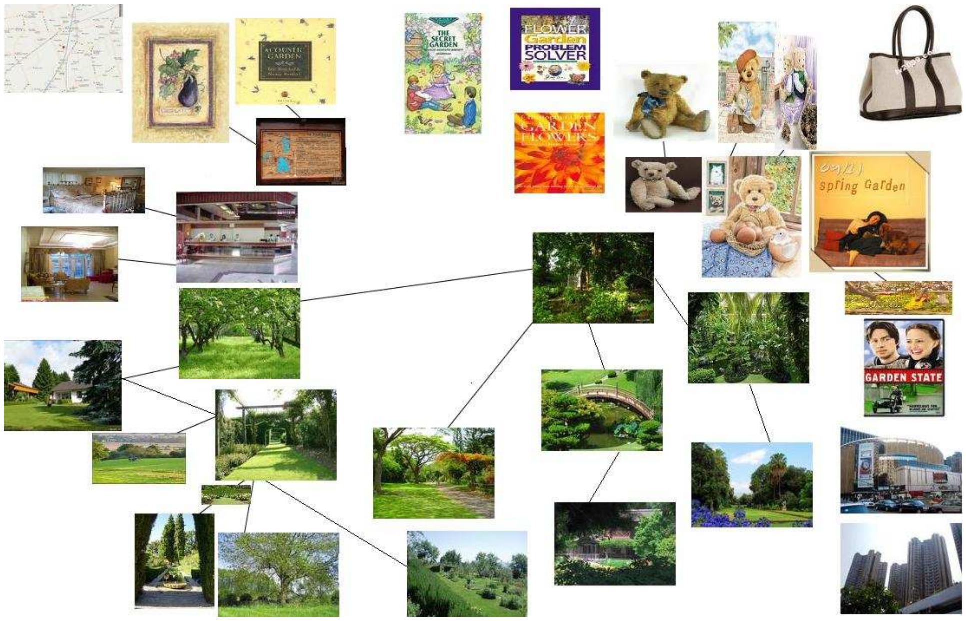


Figure 14: Image clusters and their inter-cluster correlations for the returned images of the keyword-based query “garden” from Baidu Images.

keyword-based image search engine) are partitioned into K clusters automatically by minimizing the following objective function:

$$min\left\{\Psi(C,K,\hat{\beta})=\sum_{i=1}^{K}\frac{s(G_i,G/G_i)}{s(G_i,G_i)}\right\} \tag{10}$$

where $G=\{G_i|i=1,\cdots,K\}$ is used to represent K image clusters for the given keyword-based query C (from the same keyword-based image search engine), G/G_i is used to represent other $K-1$ image clusters in G except G_i, K is the total number of image clusters, $\hat{\beta}$ is the set of the optimal kernel weights. The cumulative inter-cluster visual similarity context $s(G_i,G/G_i)$ is defined as:

$$s(G_i,G/G_i)=\sum_{u\in G_i}\sum_{v\in G/G_i}\kappa(u,v) \tag{11}$$

The cumulative intra-cluster visual similarity context $s(G_i, G_i)$ is defined as:

$$s(G_i, G_i) = \sum_{u \in G_i} \sum_{v \in G_i} \kappa(u, v) \tag{12}$$

We further define $X = [X_1, \cdots, X_l, \cdots, X_k] \in R^{n \times k}$ as the cluster indicators, and its component $X_l \in R^{n \times 1}$ is a binary indicator for the appearance of the lth cluster G_l,

$$X_l(u) = \begin{cases} 1, \ \ u \in G_l \\ \\ 0, otherwise \end{cases} \tag{13}$$

W is defined as an $n \times n$ symmetrical matrix (i.e., n is the total number of return images for the given keyword-based query C), and its component is defined as:

$$W_{u,v} = \kappa(u, v)$$

D is defined as an $n \times n$ diagonal matrix, and its diagonal components are defined as:

$$D_{u,u} = \sum_{v=1}^{n} W_{u,v} \tag{14}$$

Thus an optimal partition of large amounts of return images (i.e., image clustering) is achieved by:

$$min \left\{ \Psi(C, K, \hat{\beta}) = \sum_{l=1}^{K} \frac{X_l^T (D - W) X_l}{X_l^T W X_l} \right\} \tag{15}$$

Let $\overrightarrow{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, and $\overrightarrow{X_l} = \frac{D^{\frac{1}{2}} X_l}{\|D^{\frac{1}{2}} X_l\|}$, the objective function for our K-way min-max cut algorithm can further be refined as:

$$min \left\{ \Psi(C, K, \hat{\beta}) = \sum_{l=1}^{K} \frac{X_l^T D X_l}{X_l^T W X_l} - K = \sum_{l=1}^{K} \frac{1}{\overrightarrow{X_l}^T \cdot \overrightarrow{W} \cdot \overrightarrow{X_l}} - K \right\} \tag{16}$$

subject to: $\overrightarrow{X_l}^T \cdot \overrightarrow{X_l} = I, \ \ \overrightarrow{X_l}^T \cdot \overrightarrow{W} \cdot \overrightarrow{X_l} > 0, \ \ l \in [1, \cdots, K]$

The optimal solution for Eq. (7) is finally achieved by solving multiple eigenvalue equations:

$$\overrightarrow{W} \cdot \overrightarrow{X_l} = \lambda_l \overrightarrow{X_l}, \qquad l \in [1, \cdots, K] \tag{17}$$

(c) When an initial partition of the returned images (which are obtained from the same keyword-based image search engine for the given query C) is achieved, a post-process is further performed to estimate the optimal number of image clusters by either *splitting* the diverse image clusters or *merging* the similar image clusters. For a diverse image cluster G_l, it can be *split* into two homogeneous image clusters when its cumulative intra-cluster visual similarity context $s(G_l, G_l)$ is smaller than its cumulative inter-cluster visual similarity context $s(G_l, G_h)$ with any other image cluster G_h.

$$s(G_l, G_l) < s(G_l, G_h), \ \ h \in [1, \cdots, K], \ \ h \neq l$$

The *splitting* operation is conducted by performing another k-way min-max cut operation, with k equals to 2. $k(k-1)$ cluster pairs need to be evaluated for performing such *splitting* operation.

For two image clusters G_m and G_n, they can be *merged* as one single image cluster when their cumulative inter-cluster visual similarity contexts $s(G_m, G_n)$ is close to their average intra-cluster visual similarity context.

$$s(G_m, G_n) \approx \frac{s(G_m, G_m) + s(G_n, G_n)}{2}$$

The closeness is defined by a threshold value which is determined heuristically from the experiment results. $\frac{K(K-1)}{2}$ cluster pairs need to be evaluated for performing such *merging* operation.

(d) The objective function for kernel weight determination is to maximize the inter-cluster separability and the intra-cluster compactness. For one certain cluster G_l, its inter-cluster separability $\mu(G_l)$ and its intra-cluster compactness $\sigma(G_l)$ are defined as:

$$\mu(G_l) = X_l^T(D - W)X_l, \;\; \sigma(G_l) = X_l^T W X_l \tag{18}$$

For one certain cluster G_l, we can refine its cumulative intra-cluster pair-wise image similarity contexts $s(G_l, G_l)$ as $W(G_l)$:

$$W(G_l)_i = \sum_{u \in G_l} \sum_{v \in G_l} \kappa_i(u, v) = \beta^T \omega(G_l) \tag{19}$$

$$D(G_l)_i - W(G_l)_i = \beta^T(\epsilon(G_l) - \omega(G_l)) \tag{20}$$

where $\omega(G_l)$ and $\epsilon(G_l)$ are defined as:

$$\omega_i(G_l) = \sum_{u \in G_l} \sum_{v \in G_l} \kappa_i(u, v), \;\; \epsilon_i(G_l) = \sum_{v=1}^{n_l} \omega_i(G_v) \tag{21}$$

where $i \in [1, 2, 3]$, n_l is the total number of clusters.

The optimal weights $\vec{\beta} = [\hat{\beta}_1, \cdots, \hat{\beta}_3]$ for kernel combination are determined automatically by maximizing the inter-cluster separability and the intra-cluster compactness:

$$\arg\max_{\beta} \left\{ \frac{1}{K} \sum_{l=1}^{K} \frac{\sigma(G_l)}{\mu(G_l)} \right\} \tag{22}$$

subject to: $\sum_{i=1}^{3} \beta_i = 1, \quad \beta_i \geq 0$

The optimal kernel weights $\vec{\beta} = [\hat{\beta}_1, \cdots, \hat{\beta}_3]$ are determined automatically by solving the following quadratic programming problem:

$$\arg\min_{\beta}\left\{\frac{1}{2}\vec{\beta}^T\left(\sum_{l=1}^{K}\Omega(G_l)\Omega(G_l)^T\right)\vec{\beta}\right\} \tag{23}$$

subject to: $\sum_{i=1}^{3}\beta_i = 1, \quad \beta_i \geq 0$

$\Omega(G_l)$ is defined as:

$$\Omega(G_l) = \frac{\omega(G_l)}{\epsilon(G_l) - \omega(G_l)} \tag{24}$$

In summary, our K-way min-max cut algorithm takes the following steps iteratively for image clustering: (1) K is set as one large number and β is set equally for all these three feature subsets at the first run of iterations. It is worth noting that the optimal number of the image clusters K is determined automatically via splitting and merging operations and it does not depend on its initial value. Obviously, the setting of the initial value of K may affect the convergence rate of our K-way min-max cut algorithm. (2) Given the initial values of kernel weights and cluster number, our K-way min-max cut algorithm is performed to partition the images into K clusters according to their pair-wise visual similarity contexts. (3) Splitting and merging operations are performed on the image clusters for determining the optimal number of the image clusters (i.e., optimal K). (4) Given an initial partition of the images, our kernel weight determination algorithm is performed to estimate more suitable kernel weights, so that more precise characterization of the diverse visual similarity contexts can be achieved. (5) Go to step (b) and continue the loop iteratively until β and K are convergent or a maximum number of such iterations is reached.

Our image clustering results for two given keyword-based queries “beach” and “gar-

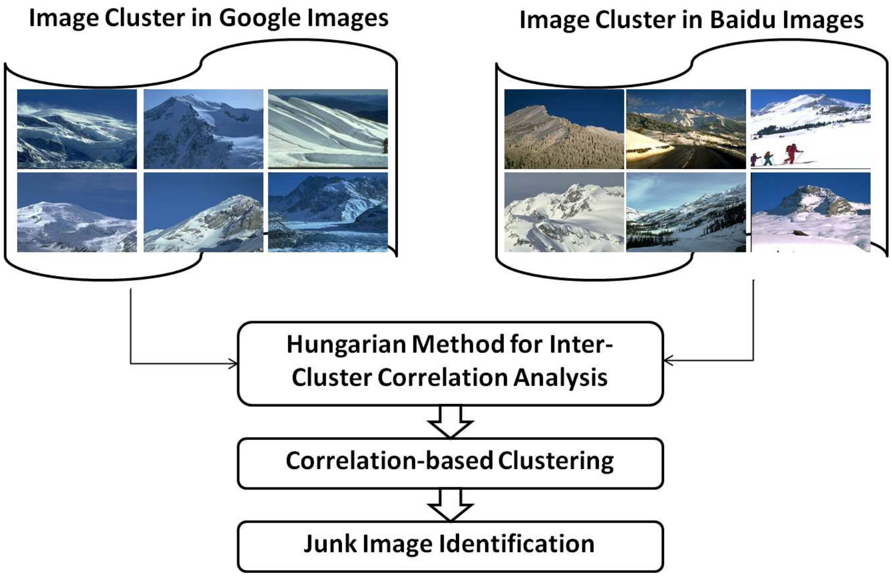


Image Cluster in Google Images
Image Cluster in Baidu Images
Hungarian Method for Inter-Cluster Correlation Analysis
Correlation-based Clustering
Junk Image Identification

Figure 15: Major steps for junk image filtering through inter-cluster correlation analysis

den", which are obtained from two image search engines (Google Images in English and Baidu Images in Chinese), are given in Figure 11, Figure 12, Figure 13, Figure 14. For each image cluster, one most representative image is selected for visualization. The visual correlations between the image clusters are represented as the edges among these most representative images. Even our image clustering algorithm can allow us to obtain more precise partitions of large amounts of returned images, it may fall short of additional information for automatically identifying the clusters of the junk images and the clusters of the relevant images.

3.3 Junk Image Filtering

Because the relevant images for the same keyword-based query (which is simultaneously performed on different keyword-based image search engines) may share some common or similar visual properties, it is very attractive to integrate the search results from multiple keyword-based image search engines in different languages to automati-

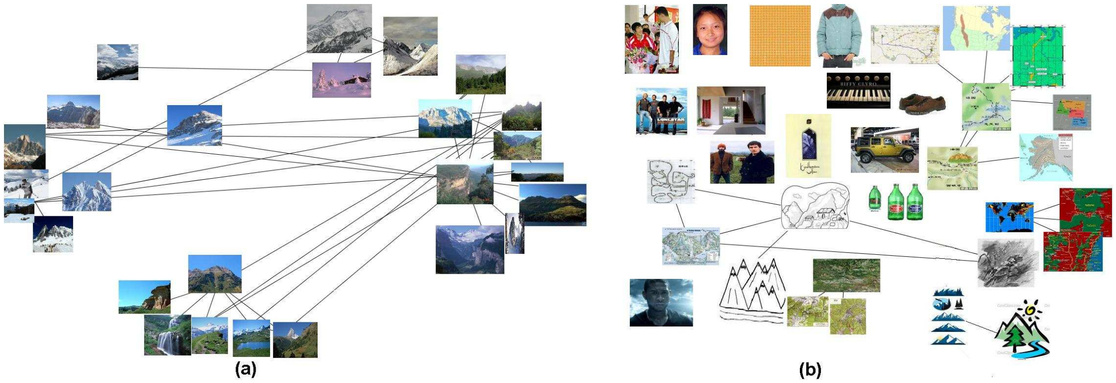


Figure 16: Inter-cluster correlation analysis for junk image filtering of the keyword-based query “mountain”: (a) bilingual image clusters for the relevant images; (b) bilingual image clusters for the junk images

cally identify the clusters for the junk images and the clusters for the relevant images. In this paper, we focus on integrating bilingual keyword-based image search results (Google Images in English and Baidu Images in Chinese) to automatically identify the clusters for the junk images and the clusters for the relevant images.

Because the query terms for image search could be short (i.e., keywords or short phrases), a dictionary-based approach is used for automatic bilingual query translation. In our current experiments, we focus on English to Chinese query translation by using an online English-Chinese bilingual dictionary. We are using the translate.google.com for English to Chinese translation. We had compare the translation results of several online dictionaries such as: Google Translate, Yahoo Babel Fish and Babylon Translate. The translation results are mostly the same because the query terms are mostly clearly-defined nouns. Even for those with different translation results, the image search results are similar.

To integrate bilingual image search results for junk image filtering, the Hungarian method [83] (as shown in Figure 15) is used to determine inter-cluster visual correlations between the image clusters for two keyword-based image search engines, Google Images in English and Baidu Images in Chinese. For two given image clusters G_i and G_j from Google Images and Baidu Images, their inter-cluster visual association $\gamma(G_i, G_j)$ is determined by using the Hungarian method [83] to optimize their bilingual pair-wise visual similarity contexts. First, one given image x for the image cluster G_i in Google Images is connected with one and only one image y for another image cluster G_j in Baidu Images, and their bilingual visual similarity context is defined as:

$$\delta(x, y) = \frac{1}{2}\left(\hat{\kappa}(x, y) + \bar{\kappa}(x, y)\right) \tag{25}$$

where $\hat{\kappa}(x, y)$ is the visual similarity context between two images x and y by using the kernel weights for the image cluster G_i, and $\bar{\kappa}(x, y)$ is the visual similarity context between two images x and y by using the kernel weights for the image cluster G_j. Second, the bilingual visual similarity context $\gamma(G_i, G_j)$ between the image clusters G_i and G_j is then determined by finding a maximum value of the sum of these bilingual pair-wise visual similarity contexts $\delta(\cdot, \cdot)$:

$$\gamma(G_i, G_j) = \max_{\delta} \frac{1}{N} \sum_{x \in G_i} \sum_{y \in G_j} \delta(x, y) \tag{26}$$

where N is the total number of such bilingual pairwise similarity contexts $\delta(\cdot, \cdot)$. The bilingual inter-cluster visual similarity contexts $\gamma(\cdot, \cdot)$ are further normalized to $[0, 1]$.

For a given keyword-based query C, all its image clusters from Google Images and Baidu Images can further be partitioned into two groups according to their bilingual inter-cluster visual similarity contexts $\gamma(\cdot, \cdot)$: positive group *versus* negative group. The returned images in the positive group have strong correlations on their visual

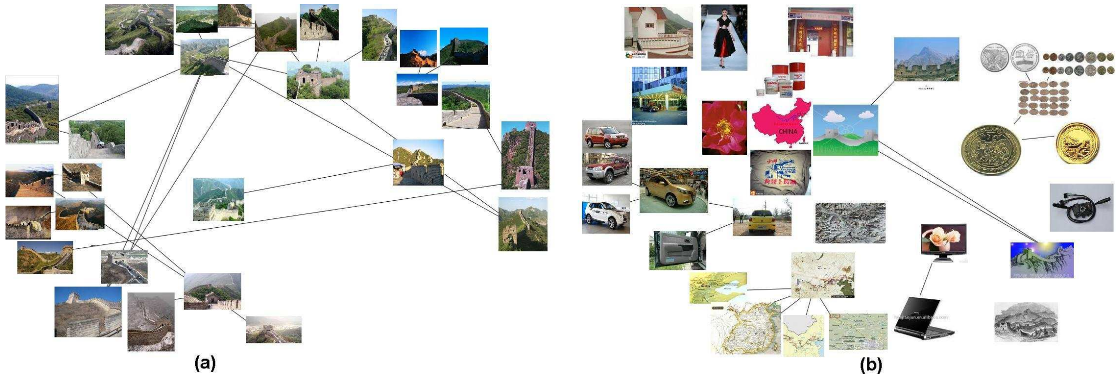


Figure 17: Inter-cluster correlation analysis for junk image filtering of the keyword-based query “great wall”: (a) bilingual image clusters for the relevant images; (b) bilingual image clusters for the junk images

properties (i.e., with larger values of the bilingual inter-cluster visual similarity contexts), thus they are strongly correlated on their visual properties and can be treated as the relevant images for the given keyword-based query C. On the other hand, the returned images in the negative group, which are weakly correlated on their visual properties and differ significantly from the images in the positive group on their visual properties (i.e., with small values of the bilingual inter-cluster visual similarity contexts), can be treated as the junk images for the given keyword-based query C and be filtered out automatically. There is a threshold value that determines whether or not two clusters are strongly correlated on their visual properties. The threshold value is gained from experiment results heuristically.

The negative clusters for the junk images may have small sizes because the junk images from different keyword-based image search engines in different languages may not have strong correlations on their visual properties, but there may have a large number of such small-size clusters for the junk images. Some experimental results for junk image filtering are shown in Figure 16,Figure 17 and Figure 18, one can observe

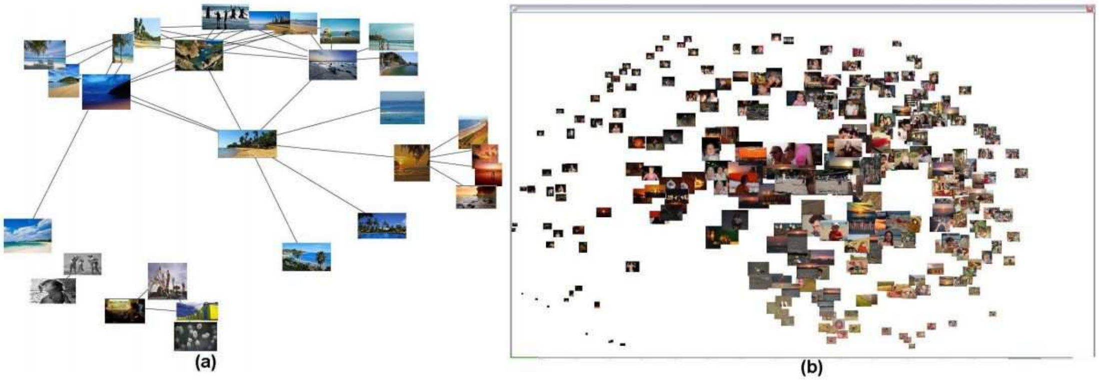


Figure 18: Inter-cluster correlation analysis for junk image filtering of the keyword-based query "beach": (a) Image cluster and their inter-cluster correlations for the returned images from Google Images; (b) bilingual image clusters for the junk images.

that our bilingual inter-cluster correlation analysis algorithm can filter out the junk images effectively.

3.4 Evaluation and Discussion

Our experiments on algorithm evaluation are performed on 5000 keyword-based bilingual queries which are simultaneously performed on Google Images and Baidu Images. The query terms are collected from multiple sources, including public image collection, dictionary and image sharing web sites. More than 4000 images are crawled for each query. To assess the effectiveness of our proposed algorithms, our algorithm evaluation work focuses on: (1) comparing the performance (i.e, the query accuracy rates) of the keyword-based image search engines before and after performing junk image filtering; (2) comparing the performance differences between various approaches for image clustering (i.e., our K-way min-max cut algorithm *versus* normalized cuts approach [131]) (because image clustering play an important role in search result partition and bilingual inter-cluster correlation determination).

It is worth noting that our junk image filtering system directly crawl the images which are returned by the keyword-based image search engines (Google Images and Baidu Images), thus our junk image filtering system will have same recall rates as that of Google Images and Baidu Images. By filtering out the junk images, our junk image filtering system can significantly improve the precision rates as compared with Google Images and Baidu Images. Based on these observations, the accuracy rate η is used to measure the performance of our junk image filtering system. The accuracy rate η is defined as:

$$\eta = \frac{\sum_{i=1}^{N} \delta}{N} \tag{27}$$

where N is the total number of returned images, δ is a delta function,

$$\delta = \begin{cases} 1, & \textit{relevant images}, \\ \\ 0, & \textit{junk images} \end{cases} \tag{28}$$

It is hard to manually annotate large-scale benchmark image set for our algorithm evaluation task and the interpretations of the junk images are largely user-dependent. Thus an image navigation system is designed to allow users to interactively provide their assessments of the relevance between the returned images and their real query intentions. The user labeling process start by receiving a query word from the user and return the result images organized in a graph structure as shown in Figure 16 and Figure 17. The user may open each individual image and label yes or no to indicate whether or not the image is relevant to the user's query intention. Multiple users are involved to assess the performance of our junk image filtering system for the same keyword-based query and their accuracy rates η are averaged as the final performance.

As shown in Figure 20, one can observe that our algorithm can improve the accuracy rates significantly, thus our algorithm can filter out the junk images effectively.

The accuracy rate ρ is used to measure the performance of our image clustering algorithm. The accuracy rate ρ is defined as:

$$\rho = \frac{\sum_{i=1}^{n} \delta(L_i, R_i)}{n} \tag{29}$$

where n is the total number of images, L_i is the cluster label for the ith image which is obtained by our K-way min-max cut algorithm, R_i is the label for the ith image which is given by a benchmark image set. $\delta(x, y)$ is a delta function,

$$\delta(x, y) = \begin{cases} 1, \quad x = y, \\ \\ 0, \quad otherwise \end{cases} \tag{30}$$

By using the same set of feature subsets for image content representation, we have compared the performance differences between two approaches for image clustering: (a) our K-way min-max cut algorithm; (b) normalized cuts [131]. As shown in Figure 20, one can observe that our K-way min-max cut algorithm can achieve higher accuracy rates for image clustering. The improvement on the clustering accuracy rate benefits from our better definitions of the inter-cluster similarity contexts and the intra-cluster similarity contexts. It is important to note that the objective function for the K-way normalized cut algorithm can be refined as:

$$Ncut(C, K) = \sum_{l=1}^{K} \frac{s(G_l, G/G_l)}{d_l} = \sum_{l=1}^{K} \frac{s(G_l, G/G_l)}{s(G_l) + s(G_l, G/G_l)} \tag{31}$$

Where d_l is the total connection from node l to all other nodes in the graph or

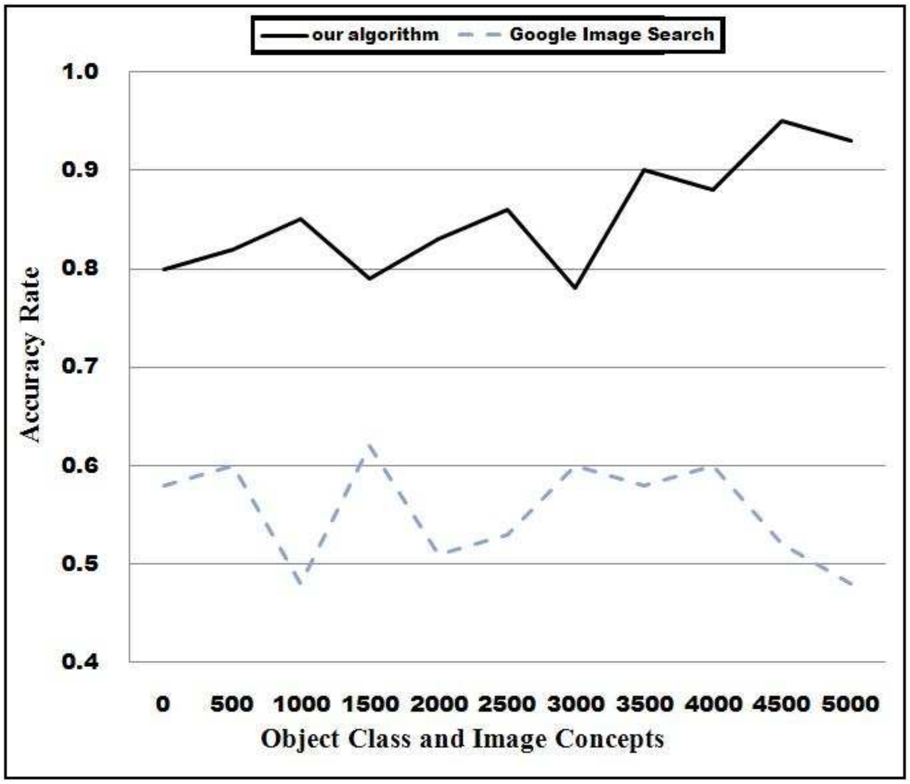


Figure 19: The comparison results on accuracy rates η for 5000 keyword-based queries before and after performing junk image filtering.

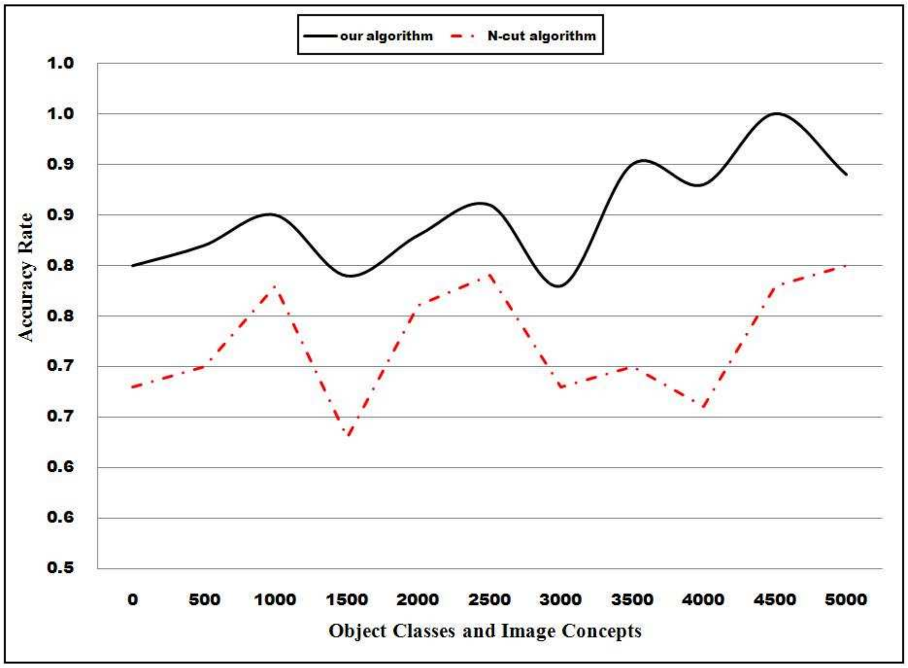


Figure 20: The comparison results on the accuracy rates ρ between our algorithm and the normalized cuts approach.

collection. G_l is defined similar as in Section 4. Because $s(G_l, G/G_l)$ may dominate to produce a smaller value of $Ncut(C, K)$, the normalized cuts algorithm may always cut out a subgraph (cluster) with a very small weight, e.g., a skewed cut. On the other hand, our K-way min-max cut algorithm has shown to be able to lead to more balanced cuts than the normalized cut [28].

CHAPTER 4: SPEED UP NEAR DUPLICATE IMAGE DETECTION

The search result should be further refined by eliminating near duplications within the relevant image clusters (such as the image pairs shown in Figure 18). We will introduce a coarse-to-fine structure near duplicate detection framework for general image collection in this Chapter and discuss its advantage in maintaining the balance between detection accuracy and computation efficiency.

4.1 Near Duplicate Detection and Removing

The task of collection refinement primarily requires efficient and accurate near duplicate detection before eliminating the redundancies. Traditional methods often require $O(n^2)$ pair-wise comparisons to detect duplicate image pairs in a group of n images. When we are dealing with large scale image set, often with n at a minimum of several hundreds, the traditional methods could be very time consuming. An intuitive idea is to use computationally cheap image features to conduct the comparison, meanwhile, we do not want to sacrifice the statistical correctness to gain computation speed up. As a trade-off, we conducted image clustering algorithm based on cheap visual features, such as global features, which can roughly partition the group of images without separating the duplicate pairs. Then the relatively expensive local features can be used for near duplicate detection on a pair-wise fashion within each cluster. An illustration of the proposed coarse-to-fine structure is shown in Figure 21.

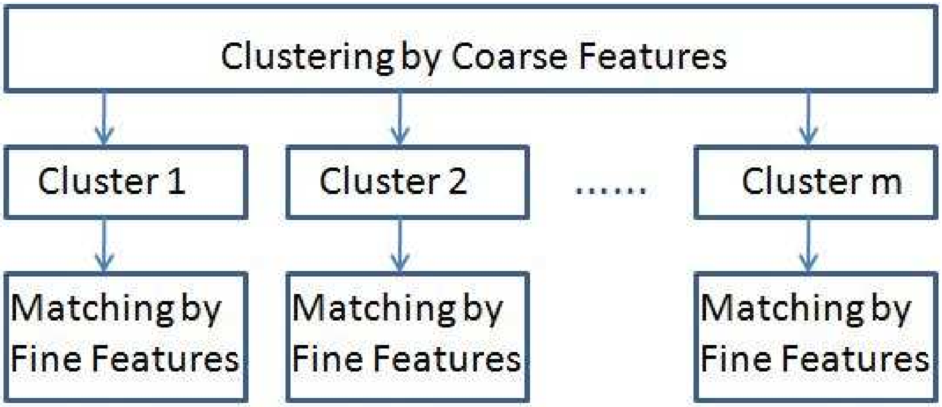

Clustering by Coarse Features
Cluster 1
Cluster 2
......
Cluster m
Matching by Fine Features
Matching by Fine Features
Matching by Fine Features

Figure 21: Coarse to fine cluster-base near duplicate detection framework.

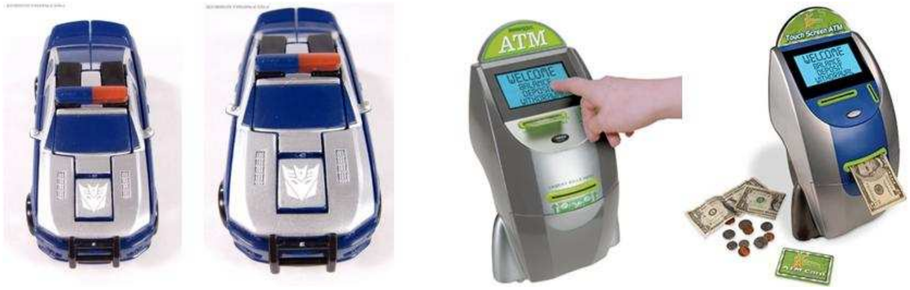


Figure 22: Example near-duplicate image pairs.

We first clarify the difference between the "duplicate" and "near duplicate" images.

- Duplicate: duplicate images are defined as image pairs that are only different on scale, color scheme or storage format.

- Near Duplicate: By relaxing the above restriction, we can define near duplicate images as duplicate pairs further varied by contrast, luminance, rotation, translation, a slight change of the layout or background.

Some examples for near-duplicate images are shown in Figure 22. From the definition above, we can learn that duplicate images are specialized near duplicate images, so that the duplicate image pairs should be within the detection results of near duplicate image pairs. The relaxation of the restriction of the near duplicate definition made the traditional Hash based method, which has been successfully applied for copy detection, inapplicable. Considering both computation efficiency and detection accuracy, a coarse-to-fine model was designed by integrating image clustering with pair-wise image matching.

As we have mentioned above, the proposed similarity detection model is sequentially partitioned into two steps: the first step is to cluster the images based on coarse features, which are usually cheap features in terms of computation cost; the second step is to conduct more complex features so as to more accurately detect duplicate image pairs within the clusters. The purpose for the first step is to roughly partition the image group while maintaining their duplication bonds within the clusters. In the second step, comparisons are conducted between image pairs in each cluster for near duplicate detection. We need more accurate object-oriented features, or in other words the local features, for the accurate detection step.

The global features of HSV color space is used in a similar way as introduced in the previous Chapter for the clustering step. Then we can partition the target image set into clusters with any computation efficient clustering algorithms, such as affinity propagation. After image set clustering, we conduct intra-cluster pair-wise image matching with local feature such as SIFT BoW model and CPAM BoW model. We design integration algorithm to combine the near-duplicate detection result from both models and propose some criterion to identify the duplicate image pairs and all the detected near duplications will be removed.

4.2 Global Approach

Object localization is very important for near duplicate detection. To utilize the global feature in near duplicate detection, the object has to be localized as accurate as possible. Meanwhile, to avoid the time consuming operation of accurate object localization, we just roughly partition the image into 5 segments and extract global color histogram feature from each of them and then concatenate the features to form our global feature.

Specifically, the features are extracted on a partition-based framework, rather than image-based or segment-based framework, as shown in Figure 23 (from left to right: image-based, partition-based, segment-based frameworks). We choose the partition-based framework based on the following consideration: a) image-based framework is not accurate enough for representing object images; b) segment-based framework is too computationally expensive and may sometimes fall into the over segmentation trouble; c) partition-based framework is a trade-off between accuracy and speed. The images are partitioned into 5 equal-sized parts, with 4 parts on the corner and 1 part at the center. We had the assumption that the object should either fill up the whole image or should lie in either one of the 5 partitions. The similarity measurement of two images will be represented as follows:

$$Similarity_{color}(X,Y) = \max_{x_i \in X, y_j \in Y} (-||x_i - y_j||^2) \tag{32}$$

$$||\alpha_i||_0 = 1 \tag{33}$$

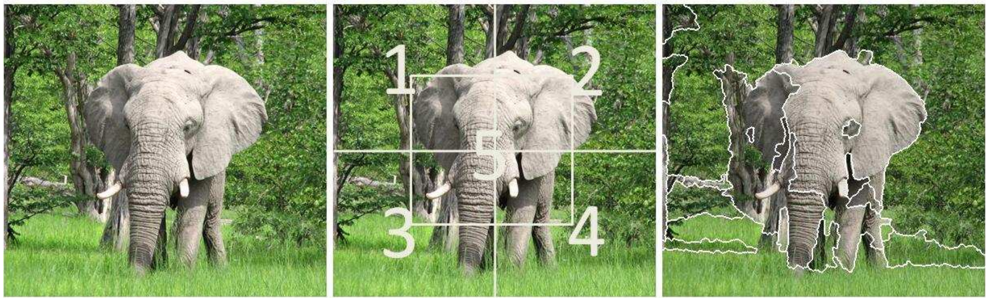


Figure 23: Global feature extraction framework: from left to right: image-based, partition-based and segment-based frameworks.

where i, j are from the partition set of X, which is composed by 5 regional partitions and one entire image. By calculating the similarity score for each of the partition pairs, the maximum score is taken as the similarity score between the two images.

Color histogram: A color histogram was used as the global feature in this model and performed on the partition-based framework. We performed on the HSV color space to extract the histogram vectors. HSV color space outperforms the RGB color space by its invariance to the change of illumination. We conducted the histogram extraction on the Hue component and formed a bin of 10 dimensions.

The data set is then clustered with Affinity Propagation algorithm into several clusters. Affinity propagation treats the data points equally as the "exemplar", then the exemplars will merge through the message-passing procedure. The merging procedure is realized by iteratively updating the responsibility and availability of the image node i to a exemplar k. The maximization of the following objective function determines the positioning of image node i:

$$c(i) = \max_{k \in X} a(i, k) + r(i, k) \tag{34}$$

where $r(i, k)$ is the responsibility of the image node i to the exemplar k, which reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i. $a(i, k)$ is the availability of the exemplar k to image node i, which reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar.

$$r(i,k) \leftarrow s(i,k) - \max_{k' s.t. k' \neq k} \{a(i,k') + s(i,k')\} \tag{35}$$

$$a(i,k) \leftarrow \min 0, r(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} \max\{0, r(i',k)\} \tag{36}$$

The availabilities are initialized to zeros: $a(i, k) = 0$, then the $r(i, k)$ and $a(i, k)$ are updated in turn until the termination criteria is satisfied. The number of final existing clusters are determined by the preference number which is could be related to the input similarities.

Affinity propagation has the advantages that it can automatically determine the number of clusters, treats each image node equally and has a relatively fast merging speed. For the next step, a more accurate pair-wise comparison of near-duplicates will be conducted within each of the clusters with local features.

4.3 Local Approach

We will use the Bag-of-Word model to represent the local visual features. The images are fine partitioned into blocks or represented by interest points; then CPAM descriptor and SIFT descriptor are applied respectively to represent each block or the neighborhood of an interest point. We will introduce these two models in detail as shown below:

There is evidence to prove the existence of different visual pathways to process color and pattern in the human visual system. Qiu et. al. [117] proposed the CPAM feature to represent color and pattern visual representations on YC_bC_r color space which gives state-of-the-art performance on content-based image retrieval applications. CPAM feature captures statistically representative chromatic and achromatic spatial image patterns and use the distributions of these patterns within an image to characterize the image's visual content. The two channels of pattern and color representation can be characterized as follows:

Let $Y = \{y(i,j), i,j = 0,1,2,3\}$ be the 4×4 Y image block. The stimulus strength of the block is calculated as

$$S = \frac{1}{16} \sum_{i=0}^{3} \sum_{j=0}^{3} y(i,j) \tag{37}$$

Then the pattern vector, $P = \{p(i,j), i,j = 0,1,2,3\}$ of the block, is formed as

$$p(i,j) = \frac{y(i,j)}{S} \tag{38}$$

A 16-dimensional vector quantizer, Q_p, is then designed for P channel using many training samples.

The color component of the model is formed by sub-sampling the two chromatic channels, C_b and C_r to form a single vector, which is also normalized by S. Let $C_b = \{C_b(i,j), i,j = 0,1,2,3\}$ and $C_r = \{C_r(i,j), i,j = 0,1,2,3\}$ be the two corresponding chromatic blocks of the C_b and C_r channels, then the sub-sampled C_b signal, $SC_b = \{SC_b(m,n), m,n = 0,1\}$ is obtained as follows

$$SC_b(m,n) = \frac{1}{4S} \sum_{i=0}^{1} \sum_{j=0}^{1} C_b(2m+i, 2n+j) \tag{39}$$

SC_r is obtained similarly. The color vector, $C = \{c(k), k = 0, 1, ..., 7\}$ is formed by concatenating SC_b and SC_r to form a 8-dimensional vector quantizer Q_c. With the above two types of vector quantizer, we build a 256-dimensional codebook for each of the two channels from large collection of training data. Each image is encoded with vector quantization technique into a 256-dimensional feature vector. The two types of codebook is concatenated into one 512-dimensional codebook, so the corresponding combined feature vector is a 512-dimension vector, with 256-dimension for pattern channel (P) and 256-dimension for color channel (C).

We also consider another well know local feature descriptor, the Bag-of-Words model with SIFT descriptor, to represent the local visual patterns of the image. For each image, the interest points are extracted with difference of Gaussian and represented with a 128-dimensional SIFT descriptor. Specifically, the descriptor is formed from a vector containing the values of all the orientation histogram entries. We have followed David Lowes's implementation [95] with a 4 by 4 array of histogram with 8 orientations in each bin. Therefore, the dimension of the feature vector for each interest point is $4 \times 4 \times 8 = 128$. A universal code book is then constructed by collecting all the available interest points from the data set. One critical issue for code book construction is to determine the size of the code book. A size too large or too small may both defect the content representation ability for the images. We have used the grid search method to browse through different size of code book and choose the best code book size in terms of the near duplicate detection performance. In our experiment, we choose a codebook size of 3000 and use the vector quantization technique to encode the images into a 3000-dimensional feature vector.

4.3.1 Indexing Scheme

Considering the size of the codebook (3000 code words for SIFT BoW model) and the scale of available images in the database (from several hundreds to several thousands), it is usually infeasible to realize real time response to query with direct indexing scheme. The inverted file indexing scheme is suitable for large scale data indexing. An inverted file table contains all the code words and an inverted list that stores a list of pointers to all occurrance of that codeword in the database images. When investigating an interest point in the query image, we will find its nearest neighbor in the dictionary table and add this point into the corresponding bin. The occurrence of all the retrieved database images in the corresponding linked list will be added by one. After we counted all the interest points in the query image, the index table will filled only with the database images which share identical interest points with the query image.

Considering the size of the codebook could be as large as several thousands (3000 in this implementation), we have designed the hierarchical k-means inverted file indexing table, as illustrated in Figure 24, which can tremendously reduce the operation time of nearest neighbor search in vector quantization step. The nearest neighbor searched is an approximate nearest neighbor, however, it is accurate enough to satisfy the requirement of our application. The out degree of each node in the hierarchical tree is 10, which means a k-means clustering with k equals to 10 is operated at each level. Specifically, the original codebook is clustered into 10 groups, and each group is further partitioned into 10 groups until the result partition is smaller than a threshold size. The code words are located on the leaf nodes only. The number of steps used for nearest neighbor search will be reduced to the logarithmic of the size of the dictionary. We need to notice that the hierarchical tree may not necessarily be balanced.

For interest point matching similarity measurement, we purely count the number of matches that falls into the same bin. The similarity between two images is measured by the number of matches between these two images. This similarity score need to be normalized by dividing the number of interest points in both participating images as in Equation 40. However, only the number of the images in the database images will count, because the number of interest points in the query image will always be the same for all the database images. The similarity between two images with SIFT BoW model is defined as follows:

$$Similarity_{sift}(i,j) = \frac{M_{ij}}{N_j} \tag{40}$$

where M_{ij} is the number of matched interest points in query image i and database image j; N_j is the number of interest points in database image j.

For CPAM feature matching similarity measurement, we will use Euclidean distance to measure the similarity of two 512-dimension CPAM BoW feature vector directly, and retrieve with direct pair-wise comparison scheme. The similarity between two images with CPAM BoW feature is defined as follows:

$$Similarity_{cpam}(i,j) = -||x_i - x_j||^2 \tag{41}$$

where x_i, x_j are the corresponding feature vector for the two images. We will use

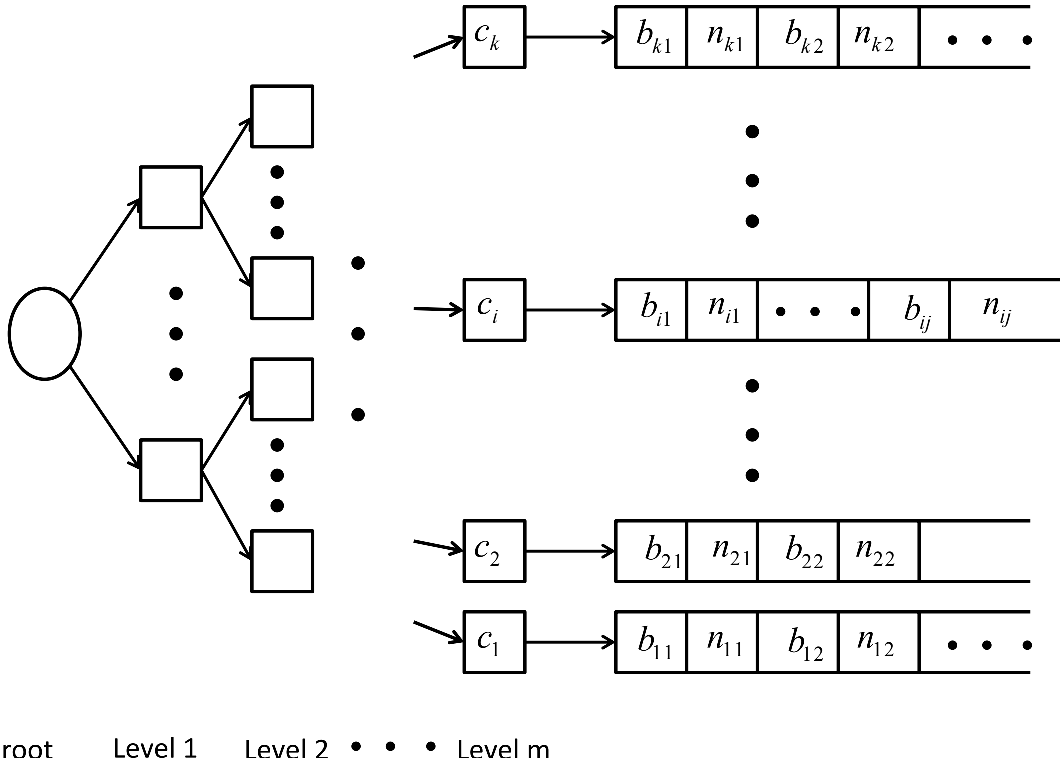

c_k
b_{k1} n_{k1} b_{k2} n_{k2} • • •
c_i
b_{i1} n_{i1} • • • b_{ij} n_{ij}
c_2
b_{21} n_{21} b_{22} n_{22}
c_1
b_{11} n_{11} b_{12} n_{12} • • •
root
Level 1
Level 2 • • • Level m

Figure 24: Hierarchical k-means indexing table and inverted file indexing scheme. c_i is the ith codeword; b_{ij} is the jth database image falling into the ith bin; n_{ij} is the number of occurrence of the key points from query image in the ith bin. The actual hierarchical tree does not have to be balanced as shown in this illustration figure.

direct indexing scheme to index the query for CPAM BoW model.

4.3.2 Multimodal Detection Integration

The CPAM feature and SIFT feature map the input images into different high-dimensional feature spaces respectively. The similarity for the feature points in these two different feature spaces are characterized by different visual aspects of the images, such as color visual pattern and texture visual pattern. As a result, each feature could be used as a strong supplement to the other in nearest neighbor search step, or the near-duplicate detection task. Before we could fusing the detection results from two different perspective, we need to firstly separate the correct matches from the entire return ranking list.

The near-duplicate detection task is intrinsically an image reranking task. The database images are ranked based on their similarity to the query image. The result ranking list will show all the similarities to the query image no matter the near duplicate matches exist or not in the database. Therefore, another research issue would be to determine if there are true matches and which returned images are true matches. Traditionally, the thresholding methods are used in this scenario for duplicates extraction where similarity value smaller than a threshold are determined as a positive match. However, it is not necessarily true since a universal threshold value could not fit for all the similarity measurements. For example, the similarity is measured by the normalized number of matched interest point pairs in the SIFT BoW model. The truth is the number of interest points in an image may vary in a large range in regard to the complexity of the visual content. Thus, for simple images, even matched image pairs does not share a large number of match interest points which means a universal threshold value would not fit for local descriptor models. In this case, we design a general method to distinguish the true matches and false matches from the top ranked return images. We will discuss the similarity distribution of these two types of features respectively as follows.

For CPAM feature, given the existence of a large number of negative matches, the similarity between the query image and the negative matches are distributed uniformly in a certain range. If there exist positive matches, the similarity between query image and the matched images should be out of the bound of the above range. For a true positive near-duplicate match, the query image and the matched image should be very close to each other on the feature space, while all the negative images are significantly far away from the matched pairs and uniformly distributed in the feature space. If we draw the distance diagram with respect to the returned ranking list, then the negative distances will form a linear function in the larger range, and the top few true matches, if exist, are outliers of this linear function. This assumption ignites us to use linear regression to reconstruct a linear function to reproduce the distribution of the distances for all the images to the query image, and the top few outlier, if exist, of this reconstructed linear function should be the true positive matches. As shown in Figure 25, there is one true positive match in the database for the given query image (the first indexed point). The majority of the non-matched distance score can be perfectly modeled by a linear regression function, and the first indexed distance score is left as an outlier to this regression function, which results in a true positive detection of near-duplicate. In actual implementation, we will randomly

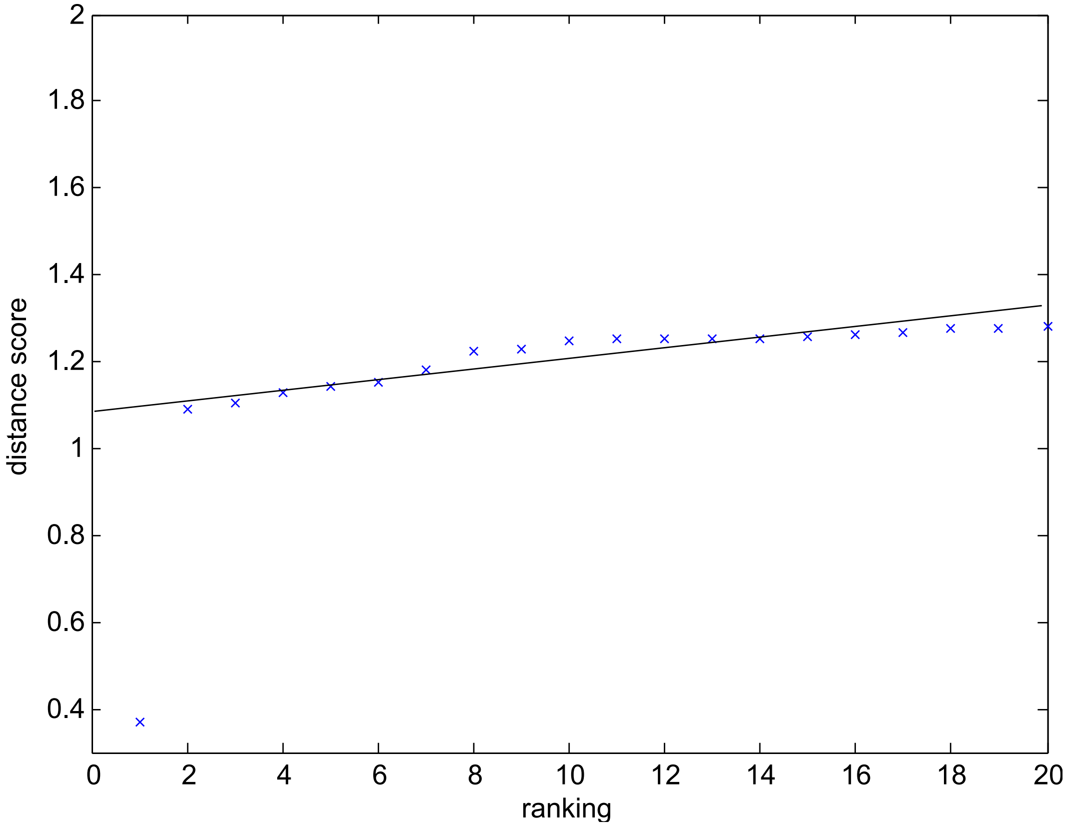


Figure 25: True positive match determination by linear regression on the CPAM feature. x-axis represents the returned ranking index for the query image in Figure 26 ; y-axis represents the corresponding distance values to the query image.

sample the retrieval results from large range for linear regression and repeat the linear regression operation for several times, to make sure the correct distribution of false matches is discovered and the true matches are left as outliers. The normalized number of matched interest points with SIFT feature as shown in Figure 26 (b) reveals similar pattern as the CPAM feature as shown in Figure 25.

The above linear regression framework models the distribution of the distances to the query image and detect the outliers with respect to the generated linear function with a predefined threshold. The duplicates are extracted as the outliers discovered with the above regression model.

For SIFT BoW model, even true matches does not distinguish itself from all the others by having a significant larger normalized number of matched interest points. We can observe from Figure 27 that there exists some images which are visually complex and tend to hamper the matching results. The given two example noisy images in Figure 27 are usually found as negative matches in many near duplicate detection queries. The number of interest points in these two images are very large and the images also show versatile visual properties. The reason for this is due to the limited representation ability of the Bag-of-Word model: non-matched points may also fall into the same bin of the codebook. A successful example with SIFT BoW model can be found in Figure 26. In order to strictly detect true duplicate pairs, we have calculate the similarity value of the query image with itself, and compared this value with the returned values from the ranking list. We set a threshold heuristically to strictly enforce that only the true duplicates are detected. Specifically, only the similarity values that is close enough to the similarity value of the query image to itself will be accepted as duplication, which can be defined below:

$$ratio = \frac{Similarity_{sift}(i,j)}{N_j \times Similarity_{sift}(i,i)} > threshold, i \neq j \tag{42}$$

where i is the query image and j is the database image.

Finally, the duplicate extraction results with CPAM BoW model is merged with the SIFT BoW model to form the final result, which realizes the multimodal integration of the two different features. Specifically, for each query, we will retrieve with both CPAM and SIFT BoW models; afterwards, we extract the duplicates from the CPAM

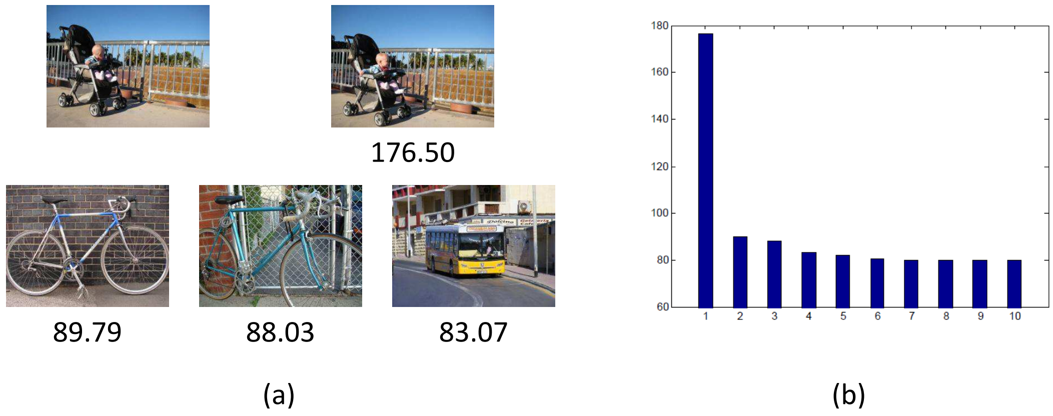


Figure 26: Near duplicate detection result with SIFT feature. (a) shows the detection results and the corresponding similarity score for a query image: the top-left image is the query image, the top-right image is the top 1 returned detection result and also true positive match; the bottom 3 images are the following 3 returned images. (b) shows the actual similarity score, in terms of normalized number of matched interest points.

Figure 27: Two example noisy images with the SIFT BoW model.

BoW model results with linear regression model, and from SIFT BoW model results with self comparison, and then combine the two extraction results to get the final duplications. All the detected duplications will be eliminated from the data collection.

4.4 Evaluation and Discussion

We have evaluated near duplicate detection performance with two similar evaluation metrics, which are *precision/recall* and *average precision*. For *precision/recall* evaluation, we only investigate the first return image from the ranking list, which is the top one detection result. If it is a true positive match, then we say the match for the query image is successfully found; if not, then the match is not detected for this query image. As a result, the precision and recall can be defined as follows:

$$precision = \frac{|\{positive\_matches\}| \cap |\{retrieved\_images\}|}{|\{retrieved\_images\}|} \tag{43}$$

$$recall = \frac{|\{positive\_matches\}| \cap |\{retrieved\_images\}|}{|\{positive\_matches\}|} \tag{44}$$

For a ranked list of return results, we can also use the *mean average precision* (mean AP) to evaluate the performance, which is more accurate than only evaluating the first returned result. For each query, we consider the average precision up to the top 10 return results, which has the discrete form of definition as follows:

$$average\_precision = \sum_{i=1}^{10} p(i)\Delta r(i) \tag{45}$$

where $p(i)$ is the probability for true positive detection in the first i return results, $\Delta r(i)$ is the change in the recall from $i-1$ to i. Mean AP equals to the mean of

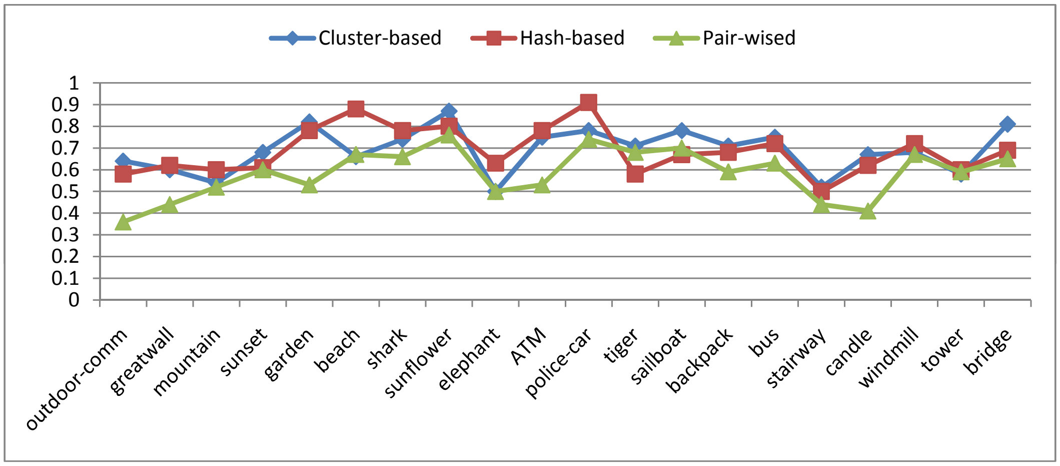


Figure 28: Precision performance comparison for the proposed and baseline detection frameworks

the average precision values from all the queries. We will evaluate the near duplicate detection results with both types of evaluation metrics.

For the near duplicate detection and elimination, we evaluate our proposed framework with other two baseline algorithms by comparing their performance on both detection accuracy (precision/recall) and computation cost. The first baseline algorithm, Hash-based algorithm, proposed in [152], partitioned the image into blocks, applied a hash function, took the binary string as the feature vector and then grouped the matches based on their Hamming distance. The second baseline algorithm, pair wise-based algorithm, used the CPAM and SIFT BoW model matching algorithms directly without applying the clustering step. We manually labeled 20 clusters from 20 different categories for duplicate pair detection as shown in Figure 17 and Figure 18. We have the following observations: a) The three models have comparable detection precision. The cluster-based model and hash-based model perform similarly and they both outperform the pair wise-based model as shown in Figure 28. b) The

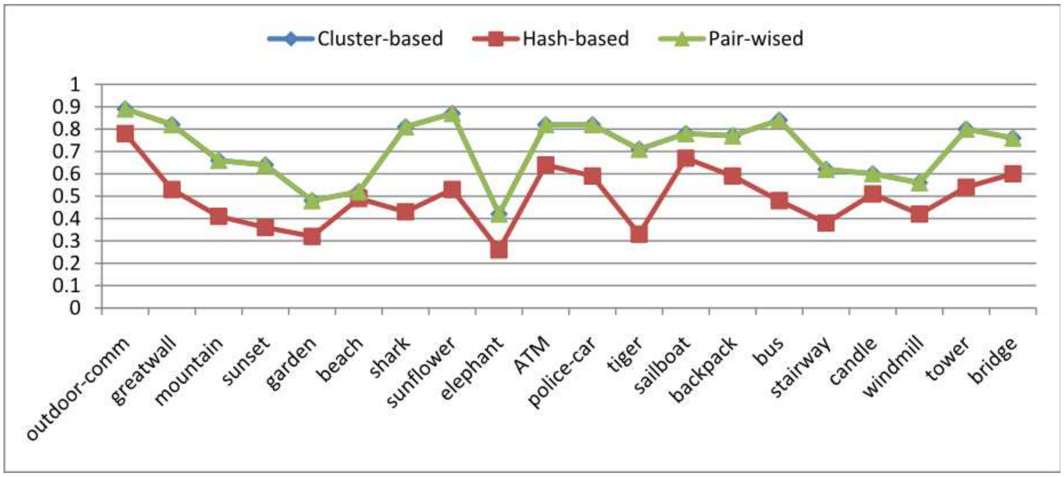


Figure 29: Recall performance comparison for the proposed and baseline detection frameworks (the line for cluster-based and hash-base framework coincide with each other)

hash-based model has a low recall score compared to the other two which means a large false positive rate. The reason is that hash-based method can successfully detect all the duplicate image pairs but miss most of the near-duplicate pairs which varies slightly on the background. On the other hand, the cluster-based model successfully detect most of the near-duplicate pairs. For example in the “outdoor commercial” set, cluster-based model successfully detected all the 7 near duplicate pairs while hash-based model missed 5 of them. So the cluster-based model has exactly the same recall performance as the pair-wised model as shown in Figure 29. The average performance for the 20 categories can be found in Table 3 and we have the conclusion that cluster-based model achieved the best average performance among the three models.

In order to evaluate the computation efficiency of the proposed framework, we counted the number of comparisons and recorded the actual runtime for each of the models on “outdoor commercial” set as appeared in Table 4. Experiment ran on a Intel Duo3.0G PC. We observed that, without considering the detection power,

	Cluster-based	Hash-based	Pair-wised
average (20 categories)	**0.72/0.71**	0.68/0.50	0.64/**0.71**

Table 3: Evaluation of the average duplicate detection ability among three frameworks (precision/recall)

	# of Comparisons	Actual Runtime(min)
Cluster-based	22136	22
Hash-based	400	1
Pair-wised	79600	69

Table 4: Evaluation of the computation efficiency among three frameworks for category "outdoor-comm"

hash-based algorithm ran much faster than the algorithms based on local features. If the detection of near-duplicate was a must, the cluster-based model outperformed the pair wise-based model dramatically by saving more than 2/3 of the computation cost. Specifically, the evaluation data set was partitioned into 7 clusters with each clustering containing 56, 66, 26, 80, 174, 4, 16 images respectively (with a total of 422 images). The computation burden for global feature clustering was insignificant, which ran at almost real time (2 sec) compared to local feature step. Furthermore, the cost saving was even remarkable as the scale of the data set increased. As shown in Figure 30, The growth pattern under the pair-wise based model satisfies the quadratic curve. The cluster based model will reduce the number of comparisons to less than 1/3 as a result.

In the following parts of this section, we will evaluate the performance of CPAM and SIFT BoW model on near duplicate detection task; our design of using linear regression to extract true positive near duplicates; and whether or not the multimodal integration structure will improve the performance when compared with using single feature model only. The near duplicate detection techniques are designed for general

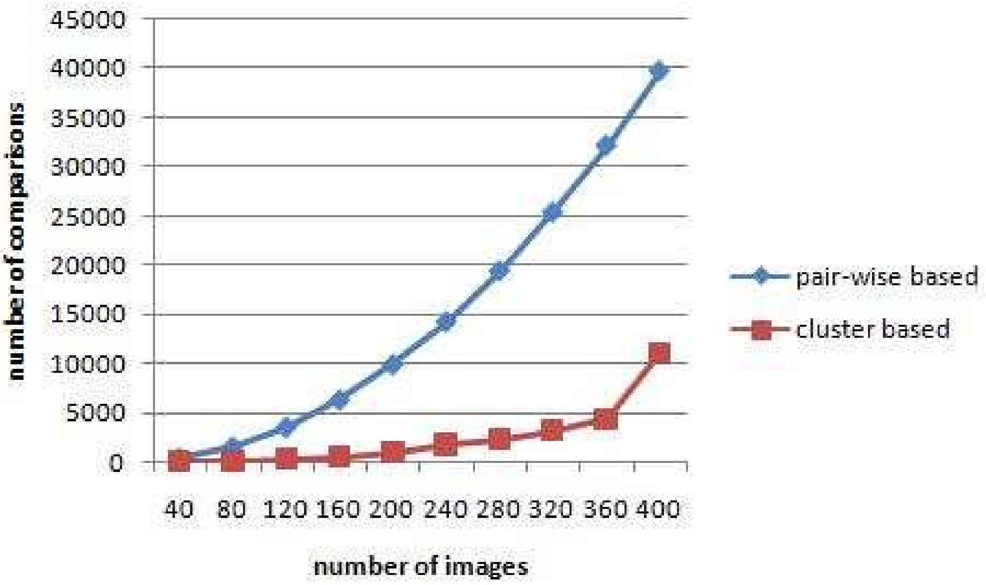


Figure 30: Computation cost with the growth of evaluation set of “outdoor-comm”

image collections. In order to make more clear comparison, we will evaluate the proposed techniques and frameworks on a more challenging data set, which is composed by 15000 images with both scene and object images. We manually extracted 190 query images. For each image, there is at least one near duplicate can be found in the data set. Some example near duplicate image pairs in this data set can be found in Figure 31.

We have compared the effectiveness of our proposed true positive detection extraction framework. The evaluation results are reported in Table 5: The false removal rate equals to 0.1537, which means the percentage of true positive detections that are falsely removed by the proposed extraction framework. We can observe that a very small percentage of true positives are removed by the proposed framework. Moreover, the removed true positive detection will be recovered by the SIFT BoW model, which is major benefit of our multimodal integration framework. The Recall value equals

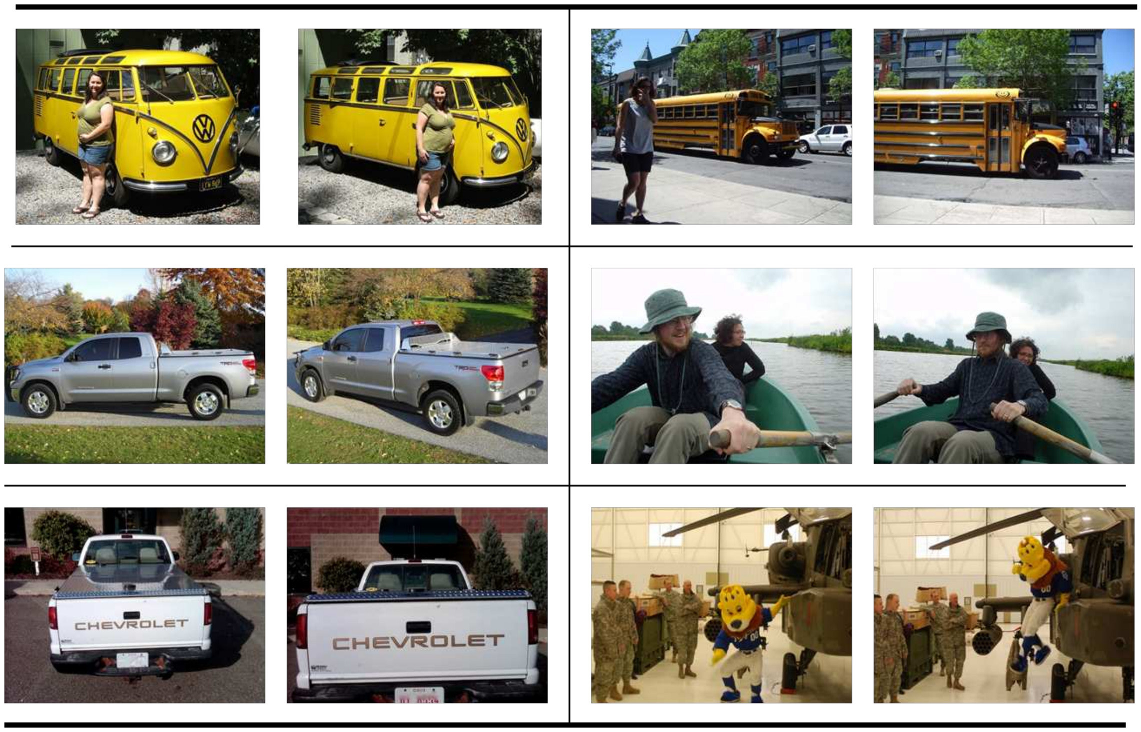


Figure 31: Example Near Duplicate Pairs: 6 example near duplicate image pairs, taken same object or scene from variant angle and with significant appearance difference.

	mean AP	Precision	Recall	False removal rate
True positive extraction	0.7694	0.7751	0.9007	0.1537

Table 5: Performance evaluation of the true positive detection extraction with CPAM BoW model. The method use linear regression to detect outliers from the returned ranking list.

to 0.9007, which means a very little percentage of false positive detections will be retained in the final detection result. The mean AP and Precision measurement are defined similarly as before. From the recall value and false removal rate, we have the conclusion that the proposed near duplicate detection framework is effective in terms of maintaining the true positive detections, as well as eliminating false positive detections.

For accurate detection with local features, we have evaluate the performance of the CPAM and SIFT BoW model in comparison with the "simply-designed" feature, such as global color histogram. The detection result can be found in Table 6. We can see from the result that the proposed "CPAM + SIFT" model performs the best, especially when compared with using single model of CPAM or SIFT. If using only single feature model, CPAM model performs better than SIFT model on average, however, we have observed both cases that, some queries work better with CPAM models, while some others work better with SIFT model, such as the examples shown in Figure 32. The top image pair in Figure 32 can be successfully detected with CPAM model while not be able to detected by SIFT model; the bottom image pair in Figure 32, on the other hand, works with SIFT model rather than CPAM model. As a result, successful combination of the detection ability of both models will inevitably increase the detection performance. Some more detection result with the proposed "CPAM

	RGB Color	SIFT	CPAM	CPAM + SIFT
mean AP	0.3540	0.5650	0.8424	0.8836

Table 6: Near duplicate detection model evaluation: global feature model, single local feature model and integrated feature model.

+ SIFT" design can be found in Figure 33. The top 3 rows in Figure 33 (a) show successful detection, with the near-duplicate images bounded by a red box; the 4th row in Figure 33 (b) shows a negative detection result, where the near-duplicate pair is not detected. The evaluation result of multimodal integration framework compared with single feature model and global feature model can be found in Table 6. From this table, we can observed that local feature models perform significantly better than global features on near duplicate detection task, by at least 50% of performance improvement in terms of mean AP. The proposed the CPAM and SIFT integration model performs the best, followed by using CPAM model alone.

For near duplicate image elimination, we will take all the images in the given image set as the query image; then conduct the near duplicate detection with the above introduced "CPAM + SIFT" framework. The detected duplicates are considered as redundancy and will be removed from the data collection.

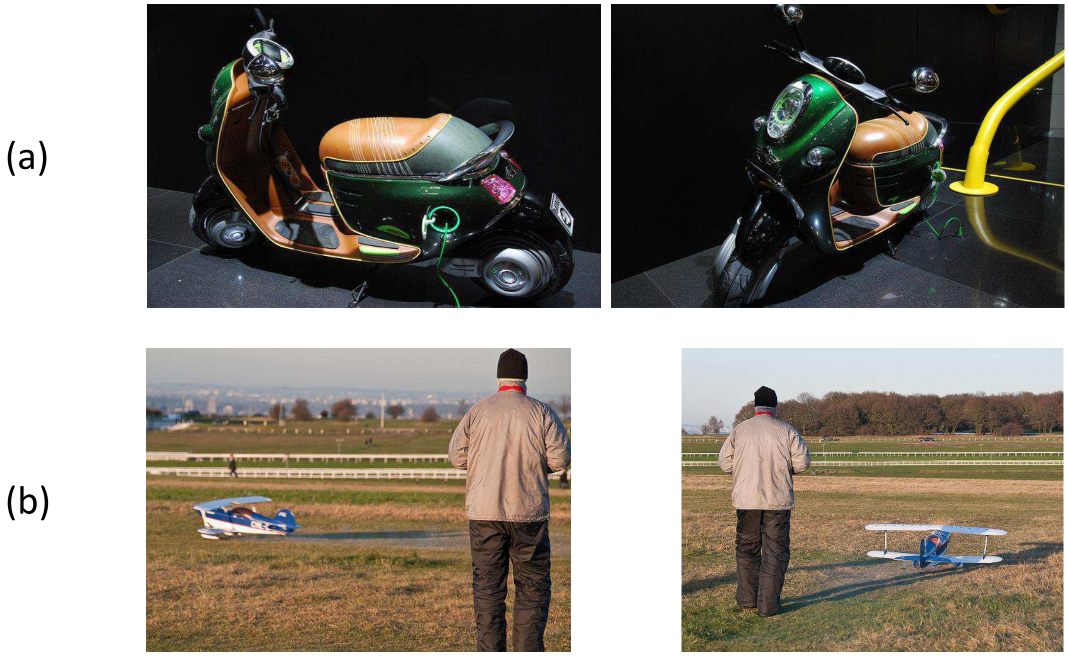


Figure 32: Comparison between CPAM model and SIFT model: (a) CPAM model works for near duplicate detection on given image pair while SIFT model do not work; (b) SIFT model works for near duplicate detection on given image pair while CPAM model do not work.

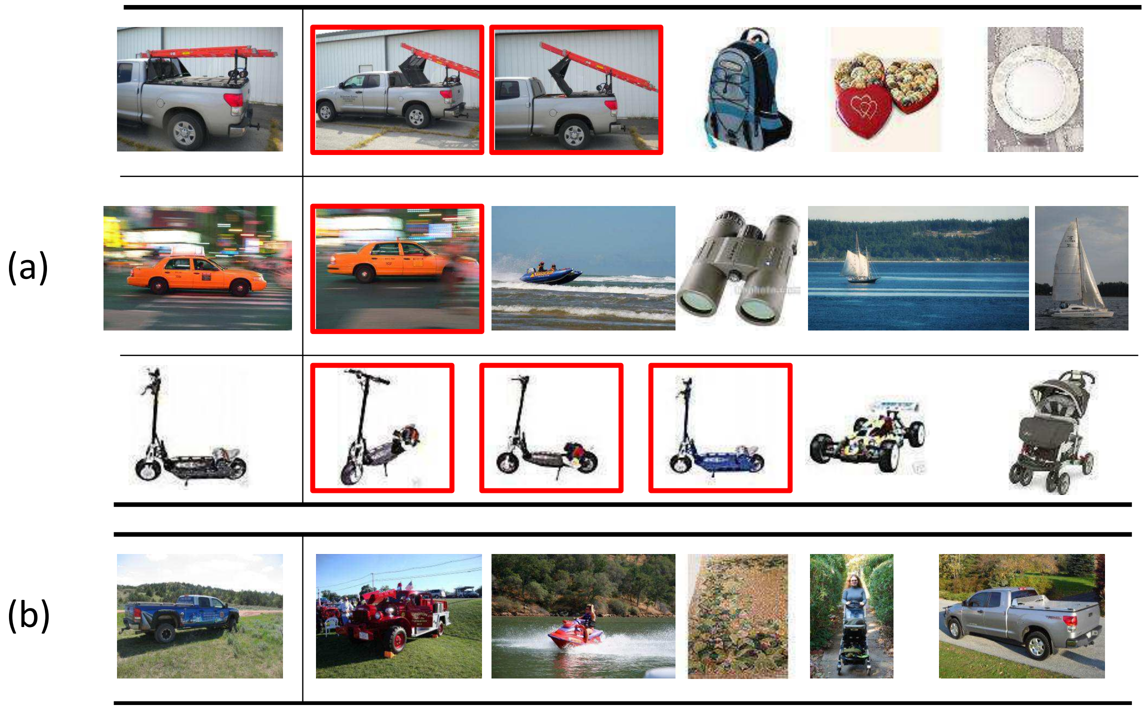


Figure 33: Example near duplicate detection result with the “CPAM + SIFT” design. The left most image is the query image, the image with a red bounding box is a positive near duplicate pair. (a) shows the result with successful detection, (b) shows unsuccessful detection

CHAPTER 5: INTERACTIVE IMAGE VISUALIZATION & EXPLORATION

After efficient refinement of the image data collection, e.g. junk image filtering and near duplicate elimination, we have achieved a more accurate category based data collection. In this Chapter, we will discuss about a novel organization of the category based image data collection: concept network.

5.1 Feature Extraction and Image Similarity Characterization

We will start this chapter with the introduction of feature extraction and similarity characterization. For image retrieval application, the underlying framework for image content representation and feature extraction should be able to: (a) characterize the image contents effectively and efficiently; (b) reduce the computational cost for feature extraction and image similarity characterization significantly. Based on these observations, we have incorporated two frameworks for image content representation and feature extraction: (a) image-based as shown in left-hand sub-figure of Figure 23; and (b) grid-based as shown in center sub-figure of Figure 23. In the image-based approach, we have extracted both the global visual features and the local visual features from whole images [95]. In the grid-based approach, we have extracted the grid-based local visual features from a set of image grids [163].

The global visual features such as color histogram can provide the global image statistics and the perceptual properties of entire images, but they may not be able to

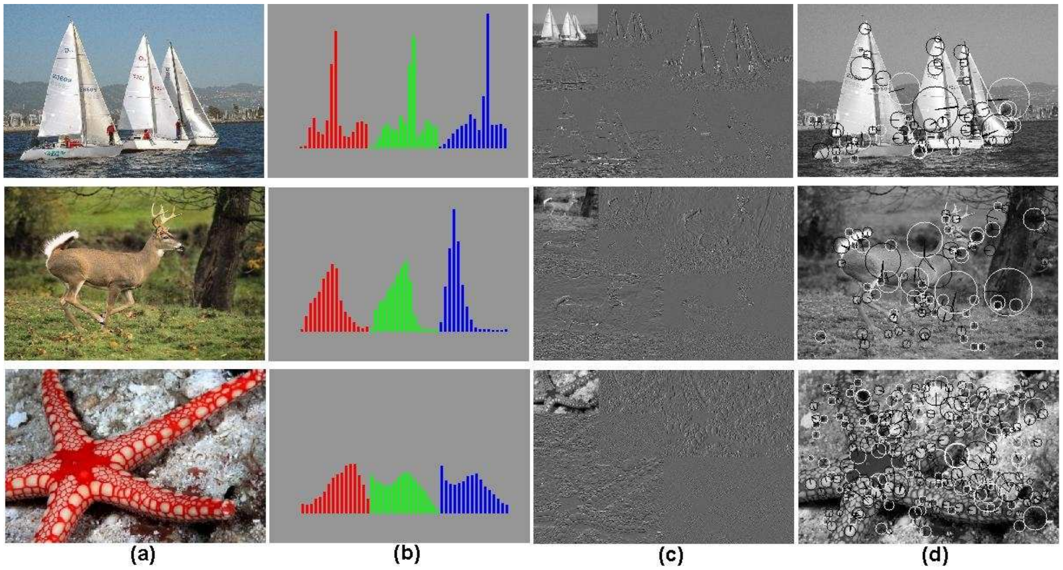


Figure 34: Image feature extraction for similarity characterization: (a) original images; (b) RGB color histograms; (c) wavelet transformation; (d) interest points and SIFT features.

capture the object information within the images [95, 8]. On the other hand, the local visual features such as SIFT (scale invariant feature transform) features and the grid-based visual features can allow object recognition against the cluttered backgrounds [95, 8]. In our current implementations, the global visual features consist of 72-bin RGB color histograms and 48-dimensional texture features from Gabor filter banks. The local visual features consist of a number of interest points and their SIFT features and a location-preserving union of grid-based visual features. As shown in Figure 34, one can observe that our feature extraction operators can effectively characterize the principal visual properties for the images.

5.2 Kernel Design and Combination

By using multi-modal visual features for image content representation, it is able for us to characterize the diverse visual properties of the images more sufficiently. On the other hand, the statistical properties of the images in such high-dimensional multi-

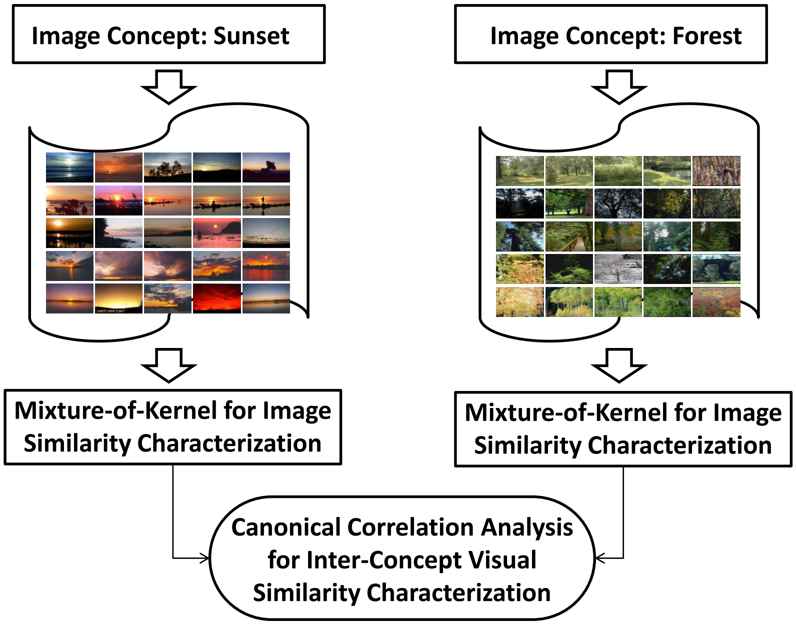


Image Concept: Sunset
Image Concept: Forest
Mixture-of-Kernel for Image Similarity Characterization
Mixture-of-Kernel for Image Similarity Characterization
Canonical Correlation Analysis for Inter-Concept Visual Similarity Characterization

Figure 35: Major components for inter-concept visual similarity determination

modal feature space may be heterogeneous, thus it is impossible for us to use only one type of kernel to characterize the diverse visual similarity relationships between the images precisely.

Based on these observations, the high-dimensional multi-modal visual features are first partitioned into multiple feature subsets and each feature subset is used to characterize one certain type of visual properties of the images, and the underlying visual similarity relationships between the images are more homogeneous and can be approximated more precisely by using one particular type of kernel. In our experiments, the high-dimensional multi-modal visual features are partitioned into three feature subsets: (a) color histograms; (b) wavelet textural features; and (c) SIFT features.

We have also studied the statistical property of the images under each feature subset, and the gained knowledge has been used to design the basic image kernel for each feature subset. Finally, multiple types of basic image kernels are integrated to characterize the diverse visual similarity relationships between the images more precisely.

5.2.1 Color Histogram Kernel

Given the histogram-based color representation, a kernel function is designed to construct the kernel matrix for characterizing the image similarity according to their color principles. We adopt the χ^2 kernel function and it is a Mercer kernel. Given two color histograms with equal length (72 bins), their statistical similarity is defined as:

$$\chi^2(u, v) = \frac{1}{2} \sum_{i=1}^{72} \frac{(u_i - v_i)^2}{u_i + v_i} \tag{46}$$

where u_i and v_i are the corresponding color bins from two color histogram u and v. The color histogram kernel function K_c(u,v) is defined as:

$$K_c(u, v) = e^{-\chi^2(u,v)/\sigma_c} \tag{47}$$

where σ_c is set to be the mean value of the χ^2 distances between all the image pairs in our experiments.

5.2.2 Gabor Wavelet Kernel

To capture the image texture, we apply a bank of wavelet filters on each images, where the image textures are represented by histogramming the outputs of the filtered channels. Consider two images u and v, and let u and v represent the corresponding feature vectors. Then the distance between the two patterns in the feature space is defined to be

$$d(u,v) = \sum_m \sum_n d_{mn}(u,v) \tag{48}$$

where

$$d_{mn}(u,v) = \left| \frac{\mu^u_{mn} - \mu^v_{mn}}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma^u_{mn} - \sigma^v_{mn}}{\alpha(\sigma_{mn})} \right| \tag{49}$$

where $\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the standard deviation of the respective features over the entire data set, and are used to normalize the individual feature components. The Gabor texture kernel function K_t(u,v) is defined as

$$K_t(u,v) = e^{-d(u,v)/\sigma_t} \tag{50}$$

where σ_t is set to be the mean value of the distances between all the image pairs in our experiments.

5.2.3 Points Matching Kernel

By matching the distance of the interesting points between images, we got the similarity value between each image pairs. For two images, their interest point set Q and P may have different numbers of interest points: M_Q and N_P. Base on this observation, the Earth mover's distance (EMD) between these two interest point sets is then defined as

$$D(Q,P) = \frac{\sum_{i=1}^{M_Q} \sum_{j=1}^{N_P} \omega_{ij} d(q_i, p_j)}{\sum_{i=1}^{M_Q} \sum_{j=1}^{N_P} \omega_{ij}} \tag{51}$$

where w_{ij} is an importance factor that can be determined automatically by solving a linear programming problem, and d(q_i,p_j) is the ground distance function between two random interesting pints q_i and p_j from Q and P. To incorporate the EMD distance into our kernel-based image similarity characterization framework, our interesting point matching kernel K_s is defined as

$$K_s(u, v) = e^{-D(Q,P)/\sigma_s} \tag{52}$$

where σ_s is set to be the mean value of the distances between all the image pairs in our experiments.

5.2.4 Multiple Kernel Combination

Because different basic image kernels may play different roles on characterizing the diverse visual similarity relationships between the images, and the optimal kernel for diverse image similarity characterization can be approximated more accurately by using a linear combination of these basic image kernels with different importance. Based on these observation, we have developed a multiple kernel learning algorithm for SVM image classifier training. For a give atomic image concept C_j, its SVM classifier can be learned by using a mixture of these basic image kernels (i.e., mixture-of-kernels).

$$\hat{K}(u, v) = \sum_{i=1}^{\tau} \alpha_i K_i(u, v), \qquad \sum_{i=1}^{\tau} \alpha_i = 1 \tag{53}$$

where τ is the number of feature subsets(i.e., the number of basic image kernels), $\alpha_i \geq 0$ is the importance factor for the ith basic image kernel K_i(u,v).

Obviously, combining different kernels can allow us to obtain better performance for image classification. On the other hand, the weights for all these three kernels may be different for different image classification tasks and could be essential to the final performance of the image classifiers. Ideally, for different image concepts, we should be able to identify different sets of these weights for kernel combination. Given

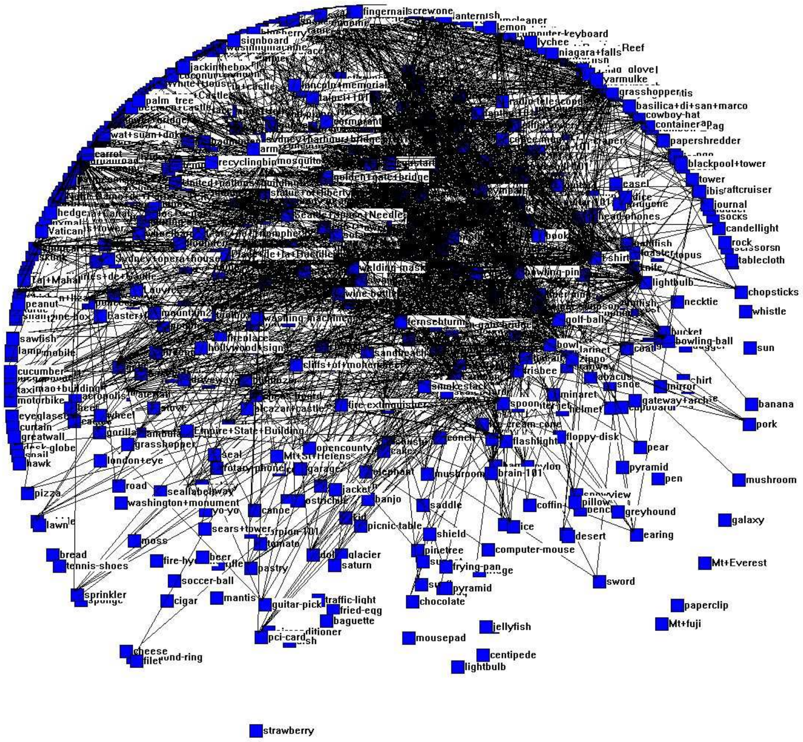


Figure 36: Visual concept network for our 600 image concepts and objects.

a set of labeled training images, the weights for these three kernels are determined automatically by searching from a given set of all the potential weights and their combinations.

5.3 Inter-Concept Visual Similarity Determination

After the image concepts and their most relevant images are available, we can use these images to determine the inter-concept visual similarity contexts for automatic visual concept network generation as shown in Figure 36. The inter-concept visual similarity context $\gamma(C_i, C_j)$ between the image concepts C_i and C_j (the linkage between two concept nodes in Figure 36) can be determined by performing kernel canonical correlation analysis (KCCA) [58] on their image sets S_i and S_j:

$$\gamma(C_i, C_j) = \begin{matrix} max \\ \\ \theta, \vartheta \end{matrix} \frac{\theta^T \kappa(S_i)\kappa(S_j)\vartheta}{\sqrt{\theta^T \kappa^2(S_i)\theta \cdot \vartheta^T \kappa^2(S_j)\vartheta}} \tag{54}$$

where θ and ϑ are the parameters for determining the optimal projection directions to maximize the correlations between two image sets S_i and S_j for the image concepts C_i and C_j, $\kappa(S_i)$ and $\kappa(S_j)$ are the cumulative kernel functions for characterizing the visual correlations between the images in the same image sets S_i and S_j.

$$\kappa(S_i) = \sum_{x_l, x_m \in S_i} \kappa(x_l, x_m), \;\; \kappa(S_j) = \sum_{x_h, x_k \in S_j} \kappa(x_h, x_k) \tag{55}$$

where the visual correlation between the images is defined as their kernel-based visual similarity $\kappa(\cdot, \cdot)$ in Equation 53.

The parameters θ and ϑ for determining the optimal projection directions are obtained automatically by solving the following eigenvalue equations:

$$\kappa(S_i)\kappa(S_i)\theta - \lambda_\theta^2 \kappa(S_i)\kappa(S_i)\theta = 0 \tag{56}$$

$$\kappa(S_j)\kappa(S_j)\vartheta - \lambda_\vartheta^2 \kappa(S_j)\kappa(S_j)\vartheta = 0 \tag{57}$$

where the eigen values λ_θ and λ_ϑ follow the additional constraint $\lambda_\theta = \lambda_\vartheta$.

When large numbers of image concepts and their inter-concepts visual similarity contexts are available, they are used to construct a visual concept network. However, the strength of the inter-concept visual similarity contexts between some image concepts may be very weak, thus it is not necessary for each image concept to be linked with all the other image concepts on the visual concept network. Eliminating the weak inter-concept links can increase the visibility of the image concepts of interest dramatically, but also allow our visual concept network to concentrate on the most significant inter-concept visual similarity contexts. Based on this understanding, each image concept is automatically linked with the most relevant image concepts with larger values of the inter-concept visual similarity contexts $\gamma(\cdot, \cdot)$ (i.e., their values of $\gamma(\cdot, \cdot)$ are above a threshold $\delta = 0.65$ in a scale from 0 to 1).

Compared with Flickr distance [162], our algorithm for inter-concept visual similarity context determination have several advantages: (a) It can deal with the sparse distribution problem more effectively by using a mixture-of-kernels to achieve more precise characterization of diverse image similarity contexts in the high-dimensional multi-modal feature space; (b) By projecting the image sets for the image concepts into the same kernel space, our KCCA and Hungarian technique can achieve more precise characterization of the inter-concept visual similarity contexts.

5.4 Concept Network Visualization

For inter-concept exploration, to allow users to assess the coherence between the visual similarity contexts determined by our algorithm and their perceptions, it is very important to enable graphical representation and visualization of the visual concept network, so that users can obtain a good global overview of the visual similarity contexts between the image concepts at the first glance. It is also very attractive to enable interactive visual concept network navigation and exploration according to the inherent inter-concept visual similarity contexts, so that users can easily assess the coherence with their perceptions.

Based on these observations, our approach for visual concept network visualization exploited hyperbolic geometry [85]. The hyperbolic geometry is particularly well suited for achieving graph-based layout of the visual concept network and supporting interactive exploration. The essence of our approach is to project the visual concept network onto a hyperbolic plane according to the inter-concept visual similarity contexts, and layout the visual concept network by mapping the relevant image concept nodes onto a circular display region. Thus our visual concept network visualization scheme takes the following steps: (a) The image concept nodes on the visual concept network are projected onto a hyperbolic plane according to their inter-concept visual similarity contexts by performing multi-dimensional scaling (MDS) [24] (b) After such similarity-preserving projection of the image concept nodes is obtained, Poincare disk model [85] is used to map the image concept nodes on the hyperbolic plane onto a 2D display coordinate. Poincare disk model maps the entire hyperbolic space onto an open unit circle, and produces a non-uniform mapping of the image concept nodes to the 2D display coordinate.

The visualization results of our visual concept network are shown in Figure 39, where each image concept is linked with multiple relevant image concepts with larger values of $\gamma(\cdot, \cdot)$. By visualizing large numbers of image concepts according to their inter-concept visual similarity contexts, our visual concept network can allow users to navigate large amounts of image concepts interactively according to their visual similarity contexts.

5.5 System Evaluation

5.5.1 Feature Extraction Framework Evaluation

We will firstly evaluate the effectiveness of the proposed partition-based feature extraction framework and the kernel-combination scheme from the perspective of an image classification task. Specifically, our benchmark experiment focuses on three issues: (a) which image content representation approach (i.e., image-based approach and partition-based approach) has better performance; (b) which feature subset has higher discrimination power; and (c) what kind of kernel combination can have better performance.

In order to reveal the advantage in characterizing visual properties of image collection, we will compare the performance of classifier under proposed design with classifiers under baseline designs. The *benchmark metric* for classifier evaluation includes *precision* ϕ and *recall* φ. They are defined as:

$$\phi = \frac{\zeta}{\zeta + \psi}, \varphi = \frac{\zeta}{\zeta + \xi} \tag{58}$$

where ζ is the set of true positive images that are related to the corresponding image concept and are classified correctly, ψ is the set of true negative images that are irrelevant to the corresponding image concept or object and are classified incorrectly, and ξ is the set of false positive images that are related to the corresponding image concept or object but are misclassified.

The images used in our benchmark experiment are partly selected from Caltech-256 [55] and LabelMe [124] (20%) and partly collected by crawling from the Internet

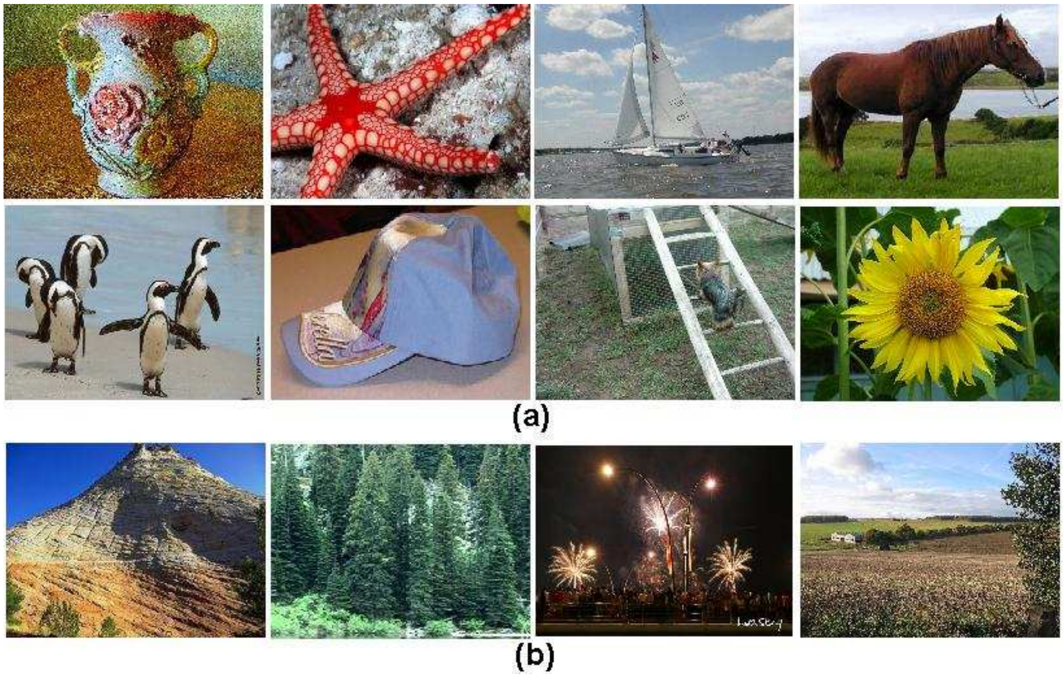


Figure 37: Sample images from 12 categories: (a) object-oriented images; (b) scene-oriented images.

(80%). The junk images from the internet are filtered with the previously introduced method. Then we generated a benchmark image set which consists of 12,000 images of 40 different image categories. The keywords for interpreting the image categories are extracted from the previously introduced concept taxonomy which have good match with human common sense and the visual properties of the images. Some of these image categories belong to scenery images (i.e., their semantics are interpreted according to their global visual properties of whole images), and some of these image categories belong to object images such as bottles and vase (i.e., their semantics are interpreted according to the underlying image objects) as in Figure 37;

Partitioning our image collections into scenery images and object images is very important for us to assess the discrimination power of various visual features and the effectiveness of various image content representation frameworks for different image

Category	Color	Texture	SIFT
airplane	**0.72**	0.57	0.50
bathtub	0.36	0.16	**0.42**
bus	**0.41**	**0.43**	0.16
building	0.31	0.20	**0.69**
elephant	0.53	0.43	**0.69**
face	0.87	0.74	0.70
firework	**0.82**	**0.68**	0.34
highway	**0.93**	**0.87**	0.27
horse	0.28	0.21	**0.86**
moon	**0.84**	**0.78**	0.33
penguin	0.27	0.23	0.18
shark	**0.52**	0.25	**0.50**
waterfall	**0.63**	0.36	0.22
desert	**0.62**	0.21	**0.66**
cloud	**0.55**	0.18	0.31
snow_view	**0.55**	0.21	0.13
farmland	**0.48**	0.22	0.6
coast	**0.63**	0.24	**0.77**
forest	**0.56**	**0.47**	0.24
mountain	0.25	0.14	0.36
backpack	0.57	0.42	0.41
baseballbat	0.70	0.71	0.49
baseballcap	0.34	0.30	**0.52**
boxingglove	**0.42**	**0.39**	0.22
calculator	0.43	0.28	0.36
coat	**0.45**	**0.46**	0.21
corridor	**0.41**	0.28	0.25
goblet	0.53	0.46	0.41
ladder	0.11	0.17	0.13
pictureframe	0.39	0.32	**0.52**
remocontroll	0.44	0.35	0.45
sailboat	0.50	0.44	0.47
saucer	0.66	0.69	0.51
starfish	0.37	0.31	0.36
sunflower	**0.81**	0.26	**0.82**
toaster	0.25	0.19	0.31
treadmill	**0.86**	**0.87**	0.14
umbrella	0.45	0.43	0.25
vase	0.19	0.13	**0.74**
wreath	**0.45**	0.24	0.17

Table 7: Classification performance in terms of precision: the features that outperform the other features are highlighted in bold.

classification tasks. Although the 40 categories are assigned into two groups named as scenery categories and object categories, the partition is not strictly done. Because there are much more object images than scenery images. 8000 images are selected randomly as training images and with each category consists of 200 images. The residue 4000 images are treated as test set.

Our experiment is done by using two PCs 2.0GHz Dual Core, 2GB RAM, and it takes 72 hours for learning the classifiers for all 40 image categories and comparing the differences of their performance under different feature subsets and different feature extraction frameworks.

From our benchmark experiments, we have observed the following interesting issues:

1. Scenes versus Objects: Under this given benchmark goal, we have compared the discrimination power of the global visual features and the local visual features for the scenery images and the object images. We have found that the global visual features such as color histograms and wavelet features are more effective for scenery image classification, on the other hand, the local visual features are more effective for object image classification as shown in Table 7. The feature subset in bold performs better than the other feature subsets with approximately 50% higher in precision rate. From these experimental results, one can observe that the SIFT feature subset tends to have better classification performance on the object categories, such as *bathtub, elephant, horse, baseball cap* and etc; the global visual features such as color and texture tend to have better classification performance for the scenery categories, such as *highway, waterfall, snow view, cloud*, etc.

2. Image-based versus Partition-based: Under this given benchmark goal, we have compared the classification performance for both the scenery images and the object images by using the image-based approach and the partition-based approach for feature extraction. As shown in Table 8, one can observe that the partition-based approach can outperform the image-based approach for most image categories which are used in our benchmark experiments. The reason for this is that the partition-based approach can characterize both the global and local visual properties of the images effectively. This observation has also told us that simple image partition may provide good performance on image classification.

3. Multiple Kernels versus Single Kernel: For the space limitation, we just list the partition-based approach for feature extraction to benchmark the performance of im-

Dataset Type	Color(%)	Texture(%)
image-based	39.675	34.15
partition-based	47.475	36.475

Table 8: Classification accuracy for the object categories.

age classifiers by using multiple kernels and single kernel. As shown in Figure 38, combining multiple kernels for diverse image similarity characterization can significantly outperform using single kernel for image classifier training. The reason behind this phenomenon is that the statistical properties of the images in the high-dimensional multi-modal feature space is heterogeneous and the diverse similarity relationships between the images cannot be approximated effectively by using one single kernel. On the other hand, our mixture-of-kernels algorithm can combine multiple kernels to approximate the diverse similarity relationships between the images more sufficiently and each basic image kernel can achieve more accurate approximation for the relevant feature subspace.

Through matching the interest points and their SIFT features, it is able to detect the objects from the images and determine the similarity between the images according to the similarity between the underlying image objects. Unfortunately, SIFT features may not have good performance for two types of images:

(a) Object images with cluttered background: This kind of images may have too many uninteresting points which come from the cluttered background. Sometimes, uninteresting points outnumber the interesting points which come from the objects. For example, the classification accuracy (precision) for the category of “treadmill” is lower than 20% if we use the SIFT features for image classifier training. On the other

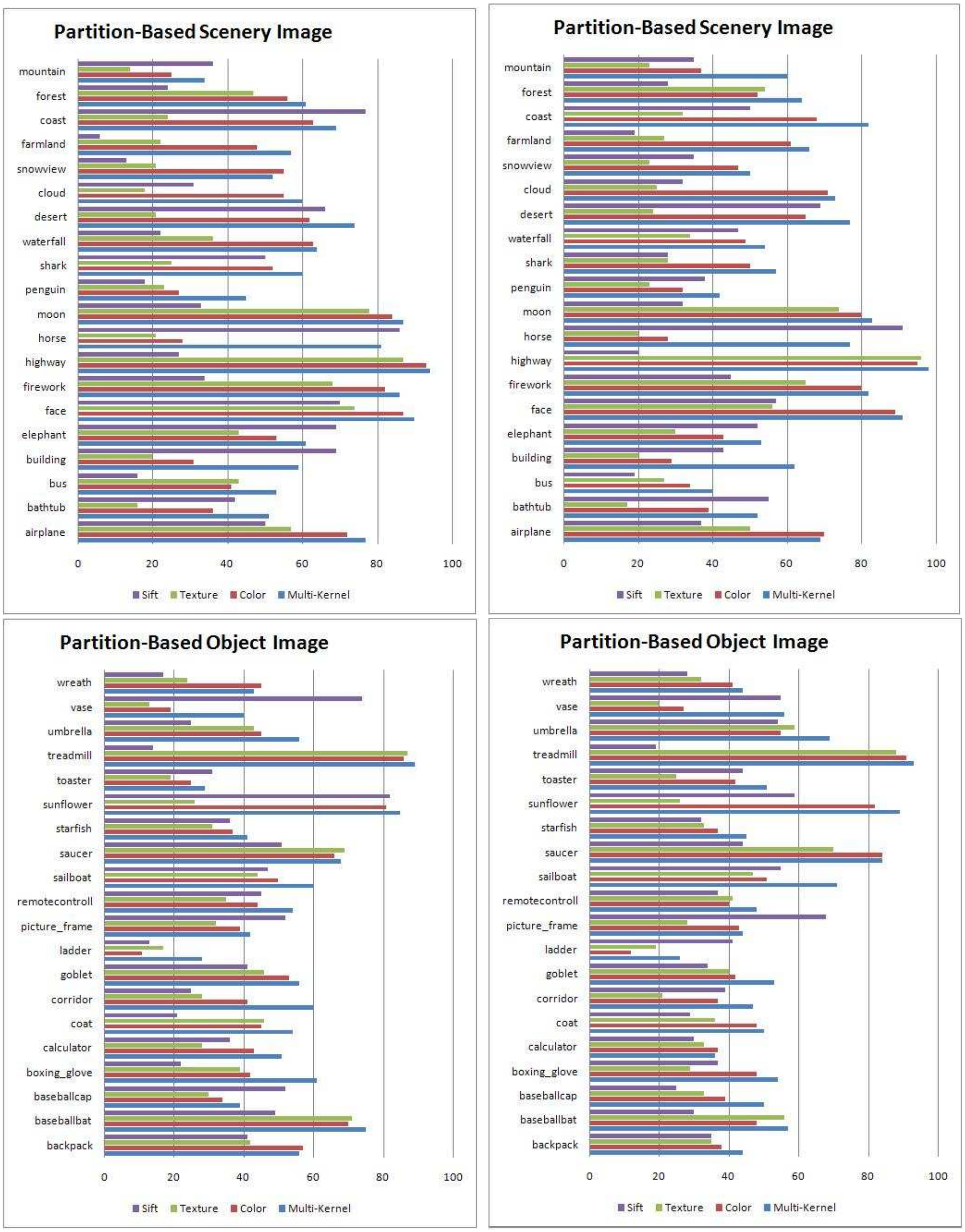


Figure 38: Classification accuracy (above/below: scenery categories/object categories; left/right: precision/recall)

hand, the accuracy (precision) for the same category “treadmill” is much higher if using the global features.

(b) Scenery images with uniform structure: For the scenery images with uniform structures like “desert”, their interesting points concentrate on the skyline with relatively similar orientation. As a result, image classification by using the SIFT features compared with using the global features may obtain very similar performance. On the other hand, for the scenery image category “forest”, image classification by using the global visual features such as color histograms may have better performance (i.e., higher precision) than using the local visual features.

From Table 7 and Table 8, one can observe that the global visual features such as color and textures are more effective for the scenery images, on the other hand, the local visual features such as SIFT are more effective for the object images. However, such correlation between the types of the visual features and the types of images is not coincidence because such correlation largely depends on the statistical properties of the images under the given feature subsets. Based on this observation, we can seamlessly integrate feature subset selection with our image use kernel combination scheme to integrate interpretation ability of multi-modal feature subset in characterizing different types of image categories.

5.5.2 Concept Network Evaluation

For algorithm evaluation used in concept network generation, we focus on assessing whether our visual similarity characterization techniques (i.e., mixture-of-kernels and KCCA) have good coherence with human perception. We have conducted both subjective and objective evaluations. For subjective evaluation, we have conducted a user study to evaluate the coherence between the inter-concept visual similarity contexts and their perceptions. For objective evaluation, we have integrated our visual concept network for exploring large-scale image collections and evaluating the benefits on using the visual concept network.

For subjective evaluation, users are involved to explore our visual concept network and assess the visual similarity contexts between the concept pairs. In such an interactive visual concept network exploration procedure, as shown in Figure 39, users can score the coherence between the inter-topic visual similarity contexts provided by our visual concept network and their perceptions. By clicking the node for each image concept, our hyperbolic concept network visualization technique can change the view into a star-schema view (as shown in Figure 39 right sub-figure), which can allow users to easily assess the coherence between their perceptions and the inter-concept visual similarity contexts determined by our algorithm. For the user study listed in Table 9, 21 sample concept pairs are selected equidistantly from the indexed sequence of concept pairs. The first one is sampled from the top and the following samples are derived every 20,000th in a sequence of 179,700 concept pairs. By averaging the scores from all these users, we get the final scores as shown in Table 9, one can observe that our visual concept network has a good coherence with human perception on the underlying inter-concept visual similarity contexts.

As shown in Table 9, we have also compared our KCCA-based approach with Flickr distance approach [162] on inter-concept visual similarity context determination. The normalized distance to human perception is 0.92 and 1.42 respectively in terms of

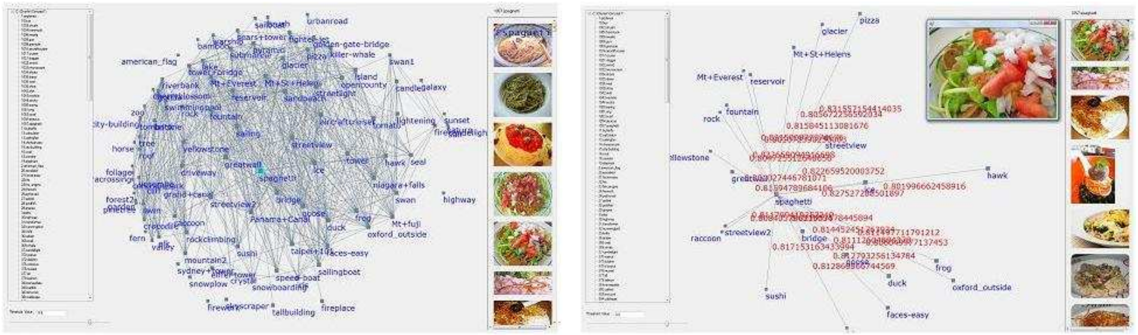


Figure 39: System User Interface: left: global visual concept network; right: cluster of the selected concept node

Euclidean distance, which means KCCA-base approach performs 54% better than Flickr distance on the random selected sample data.

We incorporate our inter-concept visual similarity contexts for concept clustering to reduce the size of the image knowledge. Because the image concepts and their inter-concept similarity contexts are indexed coherently by the visual concept network, a constraint-driven clustering algorithm is developed to achieve more accurate concept clustering. For two image concepts C_i and C_j on the visual concept network, their constrained inter-concept similarity context $\varphi(C_i, C_j)$ depends on two issues: (1) inter-concept similarity context $\gamma(C_i, C_j)$ (e.g., similar image concepts should have larger values of $\gamma(\cdot, \cdot)$); and (2) constraint and linkage relatedness on the visual concept network (e.g., similar image concepts should be closer on the visual concept network). The constrained inter-concept similarity context $\varphi(C_i, C_j)$ between two image concept C_i and C_j is defined as:

$$\varphi(C_i, C_j) = \gamma(C_i, C_j) \times \begin{cases} e^{-\frac{l^2(C_i, C_j)}{\sigma^2}}, & if \quad l(C_i, C_j) \leq \Delta \\ \\ 0, & otherwise \end{cases} \tag{59}$$

concept pair	user score	γ	Flickr Distance
urbanroad-streetview	0.76	0.99	0.0
cat-dog	0.78	0.81	1.0
frisbee-pizza	0.56	0.80	0.26
moped-bus	0.50	0.75	0.37
dolphin-cruiser	0.34	0.73	0.47
habor-outview	0.42	0.71	0.09
monkey-humanface	0.52	0.71	0.32
guitar-violin	0.72	0.71	0.54
lightbulb-firework	0.48	0.69	0.14
mango-broccoli	0.48	0.69	0.34
porcupine-lion	0.58	0.68	0.22
statue-building	0.72	0.68	0.32
sailboat-cruiser	0.70	0.66	0.23
doorway-street	0.54	0.65	0.58
windmill-bigben	0.40	0.63	0.85
helicopter-city	0.30	0.63	0.34
pylon-highway	0.34	0.61	0.06
tombstone-crab	0.22	0.42	0.40
stick-cupboard	0.28	0.29	0.51
fridge-vest	0.20	0.29	0.43
journal-grape	0.22	0.19	0.02

Table 9: Evaluation results of perception coherence for inter-concept visual similarity context determination: KCCA and Flickr distances.

where the first part $\gamma(C_i, C_j)$ denotes the inter-topic visual similarity context between C_i and C_j, the second part indicates the constraint and linkage relatedness between C_i and C_j on the visual concept network, $l(C_i, C_j)$ is the distance between the physical locations for the image concepts C_i and C_j on the visual concept network, σ is the variance of their physical location distances, and Δ is a pre-defined threshold which largely depends on the size of the nearest neighbors to be considered. In this paper, the first-order nearest neighbors is considered, $\Delta = 1$.

Our concept clustering results are given in Table 10. Because our KCCA-based measurement can characterize the inter-concept visual similarity contexts more precisely, our constraint-driven concept clustering algorithm can effectively generate the concept clusters, which may significantly reduce the cognitive load for human coherence assessment on the underlying inter-concept visual similarity contexts. By clustering the similar image concepts into the same concept cluster, it is able for us to deal with the issue of synonymous concepts effectively, e.g., multiple image concepts may share the same meaning for object and scene interpretation. Because only the inter-concept visual similarity contexts are used for concept clustering, one can observe that some of them may not semantic to human beings, thus it is very attractive to integrate both the inter-concept visual similarity contexts and their inter-concept semantic similarity contexts for concept clustering.

group 1	urban-road, street-view, city-building, fire-engine, moped, brandenberg-gate, buildings
group 2	knife, humming-bird, cruiser, spaghetti, sushi, grapes, escalator, chimpanzee
group 3	electric-guitar, suv-car, fresco, crocodile, horse, billboard, waterfall, golf-cart
group 4	bus, earing, t-shirt, school-bus, screwdriver, hammock, abacus, light-bulb, mosquito

Table 10: Image concept clustering results

CHAPTER 6: LARGE-SCALE IMAGE COLLECTION SUMMARIZATION

Image collection summarization is another important research issue for image collection exploration and image recommendation. The summarization process can be seen as an extreme form of collection refinement, which extracts a small group of most representative images from the given data set. The size of the summary is usually much smaller than the size of the original set. We have proposed a novel understanding of this problem from the perspective of dictionary learning and sparse coding.

6.1 Automatic Image Summarization

In this Chapter, we first define the criterion for assessing the quality of an image summary (i.e., whether the most representative images (image summary) are good enough to effectively reconstruct all the images in the original image set), where the problem of automatic image summarization is reformulated as the issue of dictionary learning under sparsity and diversity constraints, e.g., finding a small set of the most representative images to reconstruct all the images in the original image set in large size. We then point out the significant differences between our reformulation of dictionary learning for automatic image summarization with traditional formulation of dictionary learning for sparse coding.

6.1.1 Problem Reformulation

The BoW (Bag-of-Word) model serves as a basic tool for visual analytic tasks, such as object categorization. The summarization problem, which tries to generalize the major visual components that appear in a collection, will therefore, utilize the BoW model very well. The choice of local descriptor in BoW model is application dependent: the use of both texton descriptors [120, 6] and SIFT (Scale Invariant Feature Transform) [95] descriptors [135, 25] are widely observed. Considering the fact that texton descriptors are suitable for scene image categorization, and SIFT descriptor has a much wider range of usage, we have chosen the SIFT descriptor as the feature to construct BoW model.

Each image, in a given set, is represented with BoW model. The "visual words" in BoW model are iconic image patches or fragments which are learned by clustering methods, and therefore represents prominent visual perspectives of the entire collection. The feature vector is represented in a histogram fashion, with each bin value represents the frequency of the corresponding visual word occurrence. We can presume that the major visual contents of an image will be reflected by a large value on the corresponding bins of the feature vector; while other bins will have close-to zero values, which implies non-existence of the corresponding "visual words" in the image. Therefore, the BoW vector of an image can be understood as the distribution of the occurrence probability of the visual words or visual patterns. If we assume the visual patterns appear independently in the images and we will observe the additivity property of the BoW model, which is, one feature vector and be represented by the weighted summation of several other vectors; or the accumulated probability of the appearance of visual patterns. One visual pattern should either present or not present in an image, which implies positive and zero weights respectively. A negative weight for a vector does not have practical meaning in illustrating the additivity property of the BoW model. Therefore, sparse coefficients applied on the dictionary should be nonnegative. Such restriction is unique for summarization problem and BoW model. We also observe similar design in face recognition applications [161], which allows negative coefficients but without providing a practical explanation.

By treating the problem of automatic image summarization as the issue of dictionary learning, each image in the original image set can be approximately reconstructed by a nonnegative weighted linear combination of the summary images,or in other words, represented by accumulated probability of the appearances of various visual words (visual patterns) as shown in Figure 40. The summary images “beach” and “palmtree” will jointly reconstruct the image which has both two visual objects, and such linear correlation is reflected by the corresponding feature histograms. The above linear reconstruction model illustrates the foundation of how each image can be reconstructed by the exemplars or bases. Also from Figure 40, one can observe that the richness of the visual content in an image is limited, thus one image can only be “sparsely” represented by the bases of a dictionary. The definition of sparsity in this work is different from the dictionary selection model such as in [23], our proposed approach for automatic image summarization considers the dictionary to “sparsely” represent the images in the original image set. Based on our new definition of the reconstruction function, automatic image summarization is achieved by minimizing

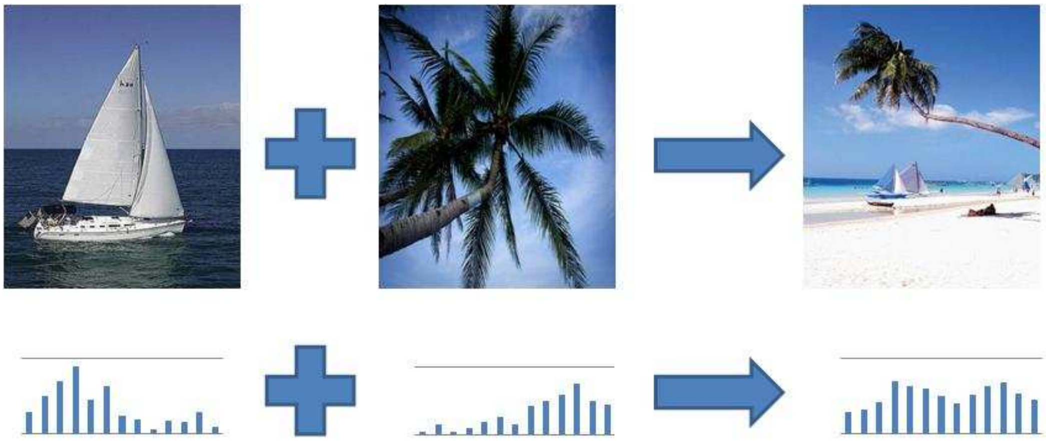


Figure 40: Demonstration for the additivity property of Bag-of-Visual-Words feature.

the overall reconstruction error in L2-norm:

$$\min \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{k} d_j \alpha_{ji}||_2^2 \tag{60}$$

where $x_i, d_j \in \mathbb{R}^d$, $\alpha_{ij} \in \mathbb{R}_0^+$. x_i and d_j are data items from the original collection; α_{ij} is the nonnegative weight for the corresponding d_j.

For the problem of automatic image summarization, $\{d_j\}$ is the set of the most representative images that we want to learn, and $\{d_j\}$ should come from the original image set. The size of $\{d_j\}$ (summary) is a trade-off between concise summarization of the original image set and accurate reinterpretation of the original image set: a small size of $\{d_j\}$ means more concise summarization of the original image set but its reinterpretation power for the original image set may reduce; on the other hand, a large size of $\{d_j\}$ guarantees a better reinterpretation power but the summarization could be verbose.

The idea of this proposed reconstruction model (for automatic image summarization) is similar to nonnegative matrix factorization which learns the prominent objects or major components of an image set. In our problem for automatic image summarization, the summary (which is learned in this manner) is inclined to be composed by the salient visual components of the original image set. If we heavily penalize on the sparsity term α (such as $||\alpha||_0 = 1$) which is used for determining the number of bases for reinterpretation, our proposed model for automatic image summarization can be reduced to k-medoids (the discrete form of k-means). The k-medoids algorithm is well known as one of the effective methods for collection summarization [168]. Thus, our proposed approach for automatic image summarization via dictionary learning for sparse representation can be treated as an extension of the k-medoids. Consequently, considering that the richness of the visual content of an image is limited, it is necessary to bring in the sparsity constraint to the objective function for guaranteeing that only a limited number of bases may take effect in the reconstruction. Hence, only the bases with non-zero coefficients are used to reconstruct the images in the original image set. Meanwhile, the bases should be diverse; each basis represents one type of principal visual patterns and all these bases should be different from each other. Thus the diversity constraint should be included in the objective function for dictionary learning. We rewrite Equation 60 as follows by adding both the sparsity constraint and the diversity constraint.

$$\min_{D,A} \sum_i ||x_i - D\alpha_i||_2^2 + \lambda \sum_i ||\alpha_i||_0 + \beta \max_{j \neq k} corr(d_j, d_k) \tag{61}$$

The problem of automatic image summarization is reformulated as the optimization problem in Equation 61, which can be jointly optimized with respect to the dictionary

D (a small set of most representative images) and the nonnegative coefficient matrix $A = [a_1^T, ..., a_n^T]^T, a_i \in R^{1 \times k}$. The diversity constraint is determined by the maximized correlation score rather than the average correlation, or the mean distance to the centroids [132]. Because the diversity (quality of the bases set) is determined by the least different bases pairs; while the mean value measurements do not guarantee the member of any pair differs from each other to some degree.

There are two different aspects between our formulation of sparse coding for automatic image summarization and traditional formulations of dictionary learning for sparse representation: 1) the coefficients $\{\alpha_{ji}\}$ have to be non-negative; 2) the dictionary D is selected from a group of given candidates (original images) X rather than their combinations. This can be explained briefly: Firstly, from our description of the accumulated appearance probability of various visual patterns, we know that each image may contain certain types of visual patterns (positive coefficients) or do not contain these visual patterns (zero coefficients). It does not make sense that any type of visual patterns contributes negatively (negative coefficients) to an image in the original image set. Thus Equation 61 has to satisfy the constraint that α has non-negative elements. Secondly, the purpose for automatic image summarization via dictionary learning is to get a small set of the most representative images from the original image set, thus the dictionary for automatic image summarization should be selected from the original image set rather than learning analytically (such as the combination or variation of the original images).

6.1.2 Dictionary Learning and Sparse Coding

The optimization problem defined in Equation 61 is NP-hard [77](i.e. the search space is discrete and can be transformed to k-medoids problem which is know NP-hard), and most existing algorithms are inevitable to fall into the traps of the local optimums. In contrast, the simulated annealing algorithm is suitable for solving the global optimization problem, which can locate a good approximation of the global optimum of a given function in a large search space.

The basic idea of exploiting the simulated annealing algorithm for dictionary learning is to avoid the local optimum by efficiently searching the solution space to obtain the global optimum solution. It is well known that the greedy algorithms seek for the local optimal solution and the final results of the AP and k-medoids algorithms largely depend on the initial inputs. During each iteration, the simulated annealing algorithm searches the neighborhood space for all the possible candidates, which is based on the *Metropolis criterion* and can effectively avoid the local traps, e.g., the candidate that does not decrease the objective function still has a chance to be accepted for the next iteration. The current global best solution will be recorded for future reference. When enough search iterations are performed, the region for the global minimum can be found with a high probability. We follow the idea of simulated annealing to design our algorithm by introducing the major components as below:

Cooling schedule: The cooling schedule is used to decide when the searching process will stop. Traditional cooling schedule is set in a simple way as $T_{k+1} = \alpha T_k$ with

$\alpha \in (0, 1)$. The canonical annealing schedules is defined as below:

$$T_k = \frac{T_0}{\log(k_0 + k)} \tag{62}$$

where k is the iteration index. The temperature T_k decreases faster during the initial steps and slower during the later steps. This can reduce the computation cost because the search space and the number of candidates are much larger in the initial steps. The temperature can be used to determine the search range and the acceptance probability, the temperature decreases monotonically to make sure that the search will terminate in a limited number of iterations.

Acceptance probability density function: The improvement of reconstruction ability is measured by the difference of the objective function, as defined in Equation 61, between two consecutive selections of the bases of the dictionary (i.e., the most representative images in the summary). The scale of the measurement decreases with the temperature and it is compared with a random threshold as below:

$$\exp(-\frac{R(D_{k+1}) - R(D_k)}{\alpha T_k}) > U \tag{63}$$

where $R(\cdot)$ is the reconstruction function as defined in Equation 61. T_k is the current temperature in the kth iteration. $U \in [0, 1)$ is randomly chosen as the acceptance threshold at each test, and new selection is accepted when the above inequity holds. The candidates, that decrease the objective function, are definitely accepted while the other candidates are accepted with a probability proportional to the current temperature.

Basis update stage: We iteratively update each basis by searching from its neighborhood in the similarity matrix S. The similarity is defined as

$$s_{ij} = \exp\,(-\frac{||x_i - x_j||^2}{2\sigma^2}) \tag{64}$$

Then we sort the columns of the similarity matrix in decreasing order. For each new basis, we randomly search in its neighborhood in terms of similarity as defined above. The search range is restricted by $\exp(\frac{T_k - T_0}{T_0}) \cdot |X|$ which defines the maximum index that can be searched in the sorted column. During the basis update stage, each of these K bases is updated in parallel according to the above criterion. A total number of $MaxTries$ dictionaries are selected in this stage and can be filtered by the acceptance function as defined in Equation 63. The accepted dictionaries can form a candidate set and be used as the input for next iteration.

Sparse coding stage: Every time we have found a set of candidate dictionaries with the above operation, we will need to calculate a set of coefficients that can minimize the optimization function. As we have discussed before, the coefficient matrix satisfies L0-norm constraint. Given the tractability of L1-norm problem (P1) and the general intractability of the L0-norm problem (P0), it has been proved that the solutions for P1 dictionaries are the same as the solutions for P0 dictionaries when they are sufficiently sparse [30]. As discussed above about the highly sparsity of our proposed model for automatic image summarization, we can replace the L0-norm by L1-norm and seek for analytical solution. Furthermore, during the sparse coding stage, the dictionary is fixed, hence, we can reduce the objective function to the following form which overlooks the diversity constraint $\beta \max_{j \neq k} corr(d_j, d_k)$.

$$f: \quad \min_A \quad \sum_i ||x_i - D\alpha_i||_2^2 + \lambda \sum_i ||\alpha_i||_1 + C$$
$$\forall i \in [1..N], \alpha_i \geq 0$$

The above formulation is similar to the nonnegative matrix factorization and nonnegative sparse coding, so we can make use of the multiplicative algorithm [62] to solve the above convex optimization problem. The objective function is non-increasing under the update rule:

$$A^{t+1} = A^t . * (D^T X)./(D^T D A^t + \lambda 1) \tag{65}$$

where $.*$ and $./$ denote element-wise multiplication and division (respectively). A is updated by simply multiplying nonnegative factors during the update stage, so that the elements of A are guaranteed to be nonnegative under this update rule. As long as the initial values of A are chosen strictly positive ($1/k$ in our case), the iteration is guaranteed to reach the global minimum.

Diversity function: The diversity metric is measured by the correlation between two distributions rather than their Euclidean distance or cosine distance. Because the correlation of two variables is known to be both scale invariant and shift invariant when compared to Euclidean distance and cosine distance. Thus, it is more appropriate for the additive appearance property of the bag-of-visual-words model. The correlation between two images is calculated as follows:

$$corr(d_i, d_j) = \frac{(d_i - \bar{d}_i)(d_j - \bar{d}_j)}{\sigma_i \sigma_j} \tag{66}$$

where $\bar{d}$ is the mean value of the vector and σ is the standard deviation.

The dictionary and the coefficients are updated in turns. In practical implementation, the current optimal combination is always saved as (A_{opti}, D_{opti}) which keeps $R(A_{opti}, D_{opti})$ in the current minimum. The annealing process stops when the temperature reaches T_{stop} or the R_{opti} is not being updated for $MaxConseRej^2$ times of iterations. Then, we go to the iterative basis selection stage, which strictly decreases the reconstruction function until convergence.

Iterative basis selection stage: In this stage, the basis is updated iteratively and the reconstruction function is strictly decreased during each iteration. Suppose we are updating the basis b_j for those i whose corresponding coefficient α_{ij} is not zero, we fix all the other $k-1$ bases and calculate the residue as:

$$E_i = \sum ||x_i - \sum_{p \neq j} d_p \alpha_{ip}||^2 \tag{67}$$

Then a new d_j^*, which can maximally approximate the current residue $\sum E_i d_j^*$, is found and it is equivalent to

$$d_j^* = \arg\min < \sum E_i, d_j^* > \tag{68}$$

which means d_j^* is the closest point to the center of all the nonzero E_i. Then we check whether d_j^* decreases the objective function or not. After all the K bases are updated, we calculate the coefficient matrix by using the method which is introduced in the sparse coding stage and repeat the updating process until convergence. The algorithm stops when no basis is being updated. The purpose for this stage is to make

[2] *MaxConseRej* stands for maximum consecutive rejection

sure that our the proposed algorithm can converge to some points. The algorithm is summarized as Algorithm 1.

6.2 Evaluation and Discussion

In this section, we report our experimental setup and algorithm evaluation results. The experiments are designed to acquire both the objective performance and the subjective performance of our proposed algorithm as compared with other 6 baseline algorithms such as SDS (sparsifying dictionary selection) [81], K-medoids [168], AP (Affinity Propagation) [3], Greedy (Canonical View) [133], ARW (Absorbing Random Walk) [153] and K-SVD [1].

6.2.1 Experiment Setup

Image Sets: The image sets used in this work are collected from ImageNet [40], NUS-WIDE-SCENE [17] and Event image set [89]. ImageNet is an image collection which is organized according to the WordNet hierarchy. The majority of the meaningful concepts in WordNet are nouns (80,000+) which are called "synset". There are more than 20,000 such synsets/subcategories in ImageNet and we have downloaded only partial of this large-scale image set and reported our summarization results on 13 object categories of *bakery, banana, bridge, church, cinema, garage, library, monitor, mug, pajama, school bus, skyscraper*, and *mix*.

The algorithms for automatic image summarization should work on image collections with various sizes and visual variety, so we have integrated the images from ImageNet to construct a new image category called *mix* by mixing the images from multiple object categories to strengthen the visual diversity and enlarge the size of

Algorithm 1 Proposed Dictionary Learning

```
Input: Original image set $X \in \mathbb{R}^{d \times n}$.

Output: Optimized dictionary $D_{opti} \in \mathbb{R}^{d \times k}, k \ll n, D_{opti} \in X$.

Initialization: Initial dictionary is appointed by random selection of $k$ bases from $X$.

Basis Update:

while $T^k > T_{stop}$ and $Rej < MaxConseRej$ do
    $T^{k+1} = Update\_T(T^k)$
    for each $d$ in $D^k$ do
        $d' = Update\_D(d)$
        if $accept(d', T^k)$ then
            $D^{k+1} = D^{k+1} \cup d'$
        end if
    end for
    $A = Sparse\_Coding(X, D^k, T^k)$
    if $R(X, A, D^k) < R_{opti}$ then
        $R_{opti} = R(X, A, D^K)$
        $D_{opti} = D^k$
    else
        $Rej = Rej + 1$
    end if
end while
Iterative Selection:
while not converge do
    for $i = 1$ to $k$ do
        $Update(d_i)$
    end for
    $A = Sparse\_Coding(X, D^k, T^k)$
    $R_{opti} = R(X, A, D^K)$
    $D_{opti} = D^k$
end while
```

	bakery	banan	bridge	church	cinema	garag	libra	monitor	mug	pajama	schoolbus	skyscraper	mix
number of images	1214	1409	1598	1329	1392	1291	1305	1399	1573	900	1303	1546	1759
size of summary	10	26	33	34	25	20	16	31	27	18	22	37	31

Table 11: ImageNet Data Collection Statistics: 13 object categories with different size of summary.

image category. For each of these 13 categories used in our experiments, the number of images ranges from 900 to 1800 and the predefined size of image summary is reported in Table 11.

The NUS-WIDE database consists of 269,648 images which are collected from Flickr. We focused on a subset called NUS-WIDE-SCENE which covers 33 scene concepts with 34,926 images in total. We have collected 11 scene concepts which are *beach* (449 images), *building* (451), *clouds* (317 images), *hillside* (466 images), *lakes* (383 images), *plaza* (425 images), *running* (302 images), *skyline* (147 images), *sunrise* (111 images), *weather* (225 images) and *zoos* (448 images).

The Event image set contains 8 sport event categories: *rowing* (250 images), *badminton* (200 images), *polo* (182 images), *bocce* (137 images), *snowboarding* (190 images), *croquet* (236 images), *sailing* (190 images) and *rock climbing* (194 images). The images in the Event image set are closer to personal photo album which focuses on the presence of people or ongoing activities.

Each of these three image sets covers different visual aspects: ImageNet focuses on object categories, NUS-WIDE-SCENE focuses on natural scene categories, and Event image set focuses on event categories.

Experimental Specification: We extract interest points and calculate their SIFT descriptors for image representation. A universal codebook with 1000 visual words is constructed, where the k-means algorithm is performed on 10 million interest points as introduced in Chapter 6.1.2 for codebook (dictionary) learning. We have investigated how the size of dictionaries will affect the reconstruction performance. The affection of the dictionary size is evaluated on the reconstruction performance on the mixture data set and we observed that too small size (less than 500) or too large size (larger than 100000) dictionary will all reduce the reconstruction performance, as shown in Figure 41, so that we choose size 1000 for the purpose of computation efficiency. The image representation (1000-dimensional histogram of code words) is obtained by quantifying all the interest points in the images into the codeword dictionary. In our experiments, we have found that our 1000-dimensional codebook can produce good representations of the images. In the following, without special indication, we denote the number of images in the given category by N and the number of code words by K.

Baseline algorithms: We have selected 6 baseline algorithms for comparison.

The k-medoids algorithm [168] is a typical clustering-based image summarization algorithm, k is the number of clusters or the size of the dictionary and the medoid of each cluster is selected as one basis. The clustering algorithm aims to partition the original image set into k clusters which can minimize the within-cluster sum of the square errors:

$$\min_{S} \sum_{i=1}^{K} \sum_{x_j \in S_i} ||x_j - d_i||^2$$

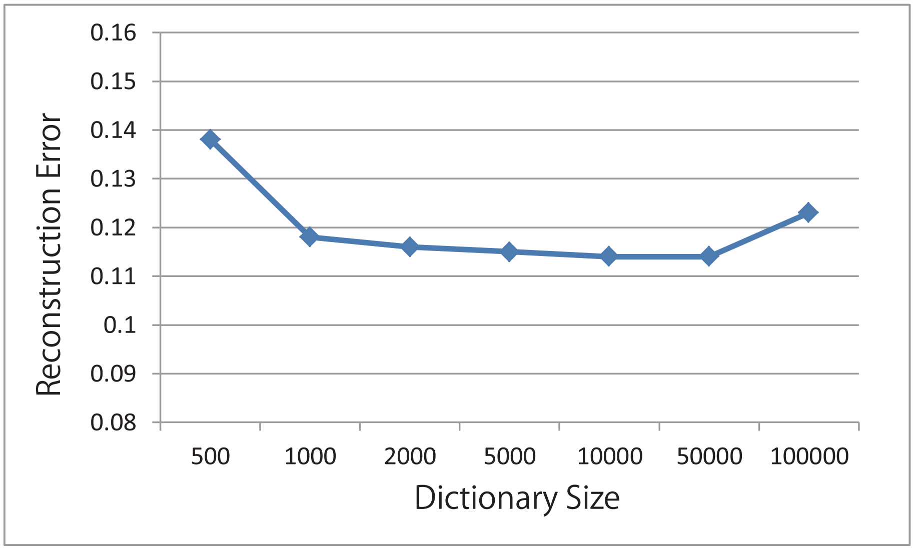


Figure 41: MSE performance in terms of different dictionary size on a mixture data set with summary size equals to 9.

The SDS algorithm [81] represents a series of greedy algorithms which iteratively select the current best basis. Krause *et al.* suggested in [81] that the local optimal derived by the greedy algorithm is a near-optimal solution when the data collection satisfy the submodular condition. The greedy algorithm starts with an empty dictionary D, and at every iteration i adds a new element (basis) via

$$d_i = \arg \min_{d \in X \setminus D} F(D_{i-1} \cup d)$$

where F is the evaluation function. The SDS algorithm is modified to satisfy our positive coefficient constraint.

The Affinity Propagation algorithm [3] updates the availability function and the responsibility function in turns. For any data point i, during affinity propagation, the value of k, which maximizes $a(i,k) + r(i,k)$, indicates that the data point i can be selected as an exemplar (basis) when $k = i$. The responsibility and availability are defined as

$$r(i,k) \leftarrow s(i,k) - \max_{k' s.t. k' \neq k} \{a(i,k') + s(i,k')\}$$

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} \max\{0, r(i',k)\}\}$$

where $s(i,k)$ is the similarity between two data points. The number of exemplars is determined by the value of the preference which is usually set to be median of the data similarities. The algorithms like AP and Greedy does not require a preset number of bases (number of clusters). If this number is required, we can obtain it by tuning the value of the preference. Instead, we can also fix the value of the preference to generate a set of bases with AP, and then we can make sure other algorithms generate the same number of bases for the same image category as shown in Table 11.

The Greedy algorithm [133] follows Simon's definition of the quality function as written below. The image, which maximally increases the quality function at each iteration, is added to the basis set D. The algorithm terminates when the quality function reduces below zero or the preset number of bases is reached. We tune the penalty weight α to ensure the required number of bases can be selected automatically.

$$Q(D) = \sum_{x_i \in X} (x_i \cdot D_{d(i)}) - \alpha |D| - \beta \sum_{d_i \in D} \sum_{d_{j>i} \in D} (d_i \cdot d_j)$$

The ARW algorithm [153] turns the selected items to the absorbing state by setting the transition probability into 0 (from the current item to other items), and the transition probability is set to 1 when it transits to itself. The selected items are arranged and the transition matrix is rewritten as

$$T_D = \begin{pmatrix} I_D & 0 \\ R & Q \end{pmatrix}$$

The item, which has the largest expected number of visits in the current iteration, is selected. The average expected number v is calculated as follows, and N is the so-called fundamental matrix

$$\begin{aligned} v &= \frac{N^T e}{n - |D|} \\ N &= (I - Q)^{-1} \end{aligned}$$

The K-SVD algorithm [1] is flexible, and works in conjugation with any sparse coding algorithms. In order to incorporate the K-SVD algorithm into the proposed framework, we learn the sparse coefficient matrix under the non-negative constraint. In the dictionary update stage, we follow the same SVD decomposition operation and update the basis iteratively. After the dictionary learned from K-SVD, we will assign each basis in the dictionary to its nearest neighbor in the original set and construct the final summarization.

For automatic image summarization, our proposed algorithm is compared with all these 6 baseline algorithms objectively and subjectively. We compare all these algorithms (our proposed algorithms and 6 baseline algorithms) on their reconstruction abilities under the sparsity and diversity constraints as defined in Equation 61, specifically, in terms of mean square error (MSE). Smaller MSE value indicates better reconstruction ability.

6.2.2 Experimental Results and Observations

MSE performance on ImageNet: The MSE value is calculated for all these six algorithms (our proposed algorithm and 6 baseline algorithms) on 13 object categories where the size of image summary is predefined as shown in Table 11. We have observed that: (a) Our proposed algorithm has the best performance in terms of the reconstruction ability on all these 13 object categories. The results are reported in Table 12 and Figure 42. For our proposed algorithm, its improvement on the reconstruction ability is insignificant when compared with K-SVD, but is significant as compared with other 6 baseline algorithms. (b) The simultaneous summarization algorithms like AP and k-medoids performed slightly better than the iterative summarization algorithms like Greedy, SDS and ARW. (c) The performance improvement on the *mix* category is especially significant, which implies that the proposed algorithm has better summarization ability on more visually diverse data collections.

The improvement comes from two aspect: (1) our proposed algorithm considers both the sparsity constraint and the diversity constraint while other baseline algorithms do not have such complete consideration of a good summary; (2) the simulated annealing algorithm is adopted to seek for the global optimum solution while all the other five algorithms seek the local optimum solutions. When the same size of image summary is used, we have also compared their performance in terms of MSE values as shown in Table 13 and Figure 43. The performance is similar to the predefined size of summary experiment.

MSE performance on NUS-WIDE-SCENE: The MSE value is calculated for all

	bakery	banana	bridge	church	cinema	garage	library
SDS	0.13	0.179	0.180	0.171	0.176	0.187	0.163
K-med	0.162	0.169	0.181	0.160	0.162	0.171	0.144
AP	0.100	0.170	0.162	0.165	0.168	0.174	0.143
Greedy	0.167	0.172	0.180	0.175	0.176	0.178	0.173
ARW	0.128	0.175	0.185	0.176	0.175	0.186	0.162
Proposed	**0.083**	**0.112**	**0.125**	**0.118**	**0.11**	**0.113**	**0.103**
K-SVD	0.117	0.123	0.125	0.119	0.123	0.128	0.112
Size	10	26	33	34	25	20	16

	monitor	mug	pajama	sch-bus	skyscrap	mix	Avg.
SDS	0.192	0.181	0.184	0.134	0.191	0.182	0.173
K-med	0.175	0.169	0.174	0.158	0.173	0.169	0.167
AP	0.175	0.168	0.175	0.162	0.177	0.168	0.162
Greedy	0.179	0.171	0.184	0.180	0.193	0.172	0.177
ARW	0.191	0.181	0.184	0.158	0.189	0.178	0.174
Proposed	**0.125**	**0.108**	**0.106**	**0.105**	**0.15**	**0.118**	**0.114**
K-SVD	0.129	0.122	0.127	0.106	**0.131**	0.121	0.121
Size	31	27	18	22	37	31	N/A

Table 12: Performance comparison of the proposed algorithm with 6 other baseline algorithms in terms of reconstruction error; 13 object categories selected from ImageNet with different summary sizes.

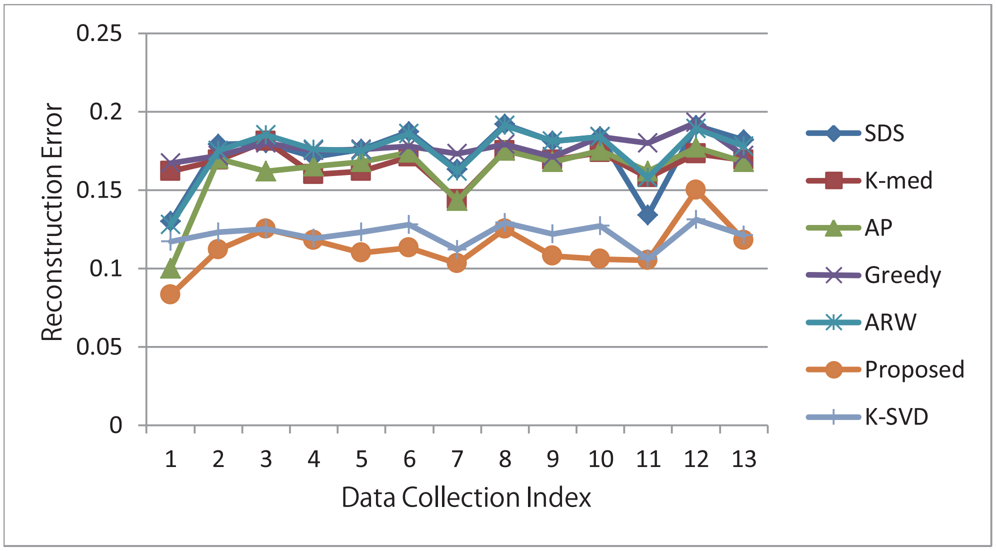


Figure 42: MSE comparison among all these 6 algorithms on ImageNet with different summary sizes.

	bakery	banana	bridge	church	cinema	garage	library	monitor	mug	pajama	sch-bus	skyscrap	mix	Avg.
SDS	0.13	0.185	0.177	0.180	0.180	0.186	0.163	0.192	0.181	0.184	0.134	0.191	0.182	0.173
K-med	0.115	0.129	0.110	0.109	0.125	0.129	0.096	0.132	0.125	0.127	0.113	0.136	0.123	0.121
AP	0.083	0.128	0.116	0.105	0.121	0.129	0.107	0.134	0.125	0.129	0.114	0.128	0.123	0.119
Greedy	0.096	0.128	0.136	0.127	0.127	0.130	0.117	0.142	0.127	0.132	0.104	0.149	0.126	0.126
ARW	0.097	0.133	0.142	0.133	0.134	0.137	0.117	0.142	0.131	0.134	0.120	0.153	0.132	0.131
Proposed	**0.0.077**	**0.088**	**0.102**	**0.094**	**0.091**	**0.092**	**0.086**	**0.099**	**0.086**	**0.089**	**0.085**	**0.121**	**0.087**	**0.092**
K-SVD	0.118	0.130	0.137	0.121	0.130	0.133	0.108	0.137	0.129	0.131	0.124	0.153	0.123	0.129

Table 13: Performance comparison of the proposed algorithm with 6 other baseline algorithms in terms of reconstruction error; 13 object categories selected from ImageNet image set with equal summary size of 9.

these 6 algorithms on 11 scene categories in NUS-WIDE-SCENE image set when the size of image summary is fixed. Similar performance is obtained as what we have got in ImageNet, however, the performance improvement for the proposed algorithm, and also among all these 6 algorithms is not as significant as we have observed in ImageNet data set, and the proposed algorithm is outperformed by other algorithms on two categories as shown in Table 14 and Figure 44. The absolute MSE value and the difference among the baseline algorithms are also smaller as compared with the object categories in ImageNet. The result demonstrates that the images in the scene categories are more evenly distributed and our proposed algorithm does not have as distinguish performance as we have obtained in the object categories.

MSE performance on Event image set: The MSE value is calculated for all these six algorithms on 8 categories in the Event image set with equal summary size of

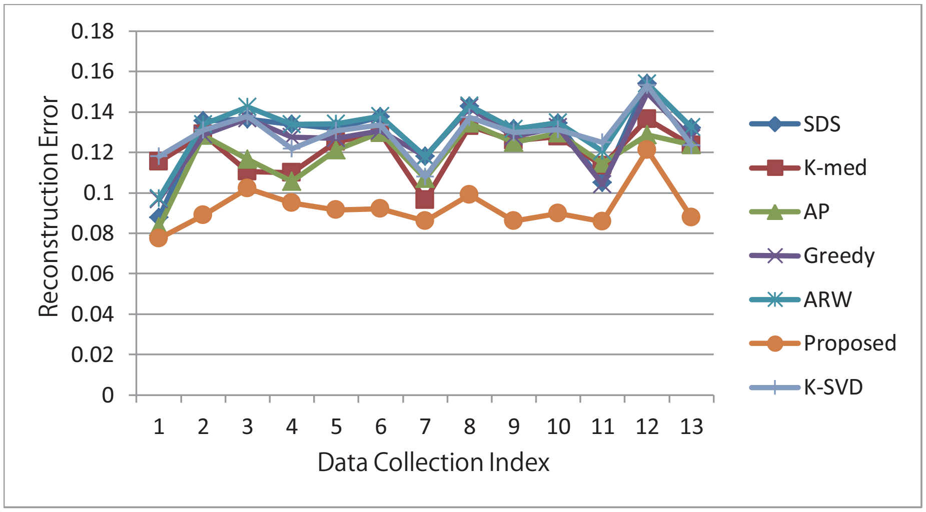


Figure 43: MSE comparison among the algorithms on ImageNet with equal summary size of 9.

9. We have observed that the MSE curves are more consistent as compared with the MSE curves for ImageNet and NUS-WIDE-SCENE and the difference is very consistent and relatively small as shown in Table 15 and Figure 45. The reason is that the images for the Event image set is organized much better and more consistent on visual content as compared with ImageNet and NUS-WIDE-SCENE.

Discussion: We will discuss how the major components and parameters will affect the performance of the proposed algorithm.

The spatial information is believed to be discarded with the proposed SIFT BoW model, which is one of the major drawbacks for BoW model. However, for the image collection summarization applications, the spatial distribution or organization of objects within a certain image is not critical. The critical property is the existence of an object or visual component in an image, and the distribution of the occurrence probability of the visual words within an image. Under such interpretation, the MSE measure should be enough to serve for image collection summarization task

	beach	building	clouds	hillside	lakes	plaza	running	skyline	sunrise	weather	zoos	Avg.
SDS	0.134	0.127	0.125	0.124	0.123	0.114	0.125	0.125	0.130	0.148	0.107	0.127
K-med	0.124	0.121	**0.105**	0.135	0.116	0.111	0.121	0.110	0.126	0.131	0.097	0.120
AP	0.125	0.116	0.115	0.109	0.119	0.103	0.113	0.108	0.118	0.130	0.099	0.116
Greedy	0.123	0.117	0.122	0.121	0.123	0.110	0.118	0.110	**0.109**	0.130	0.110	0.119
ARW	0.140	0.123	0.121	0.135	0.127	0.120	0.130	0.123	0.127	0.137	0.116	0.128
Proposed	**0.119**	**0.106**	0.106	**0.107**	**0.108**	**0.097**	**0.104**	**0.102**	0.110	**0.127**	**0.090**	**0.109**
K-SVD	0.131	0.122	0.107	0.135	0.109	0.112	0.123	0.107	0.119	**0.124**	0.108	0.118

Table 14: Performance comparison of the proposed algorithm with 6 other baseline algorithms in terms of reconstruction error; 11 scene categories selected from NUS-WIDE-SCENE with equal summary size of 9.

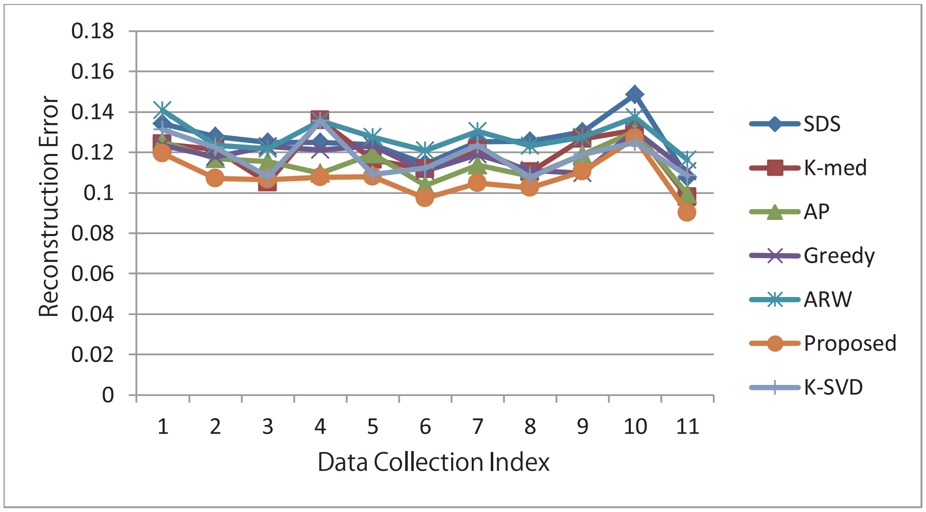


Figure 44: MSE comparison among the algorithms on NUS-WIDE-SCENE with equal summary size of 9.

	rockclimb	badminton	bocce	croquet	polo
SDS	0.088	0.119	0.099	0.102	0.103
K-med	0.086	0.108	0.103	0.097	0.092
AP	0.086	0.115	0.103	0.096	0.098
Greedy	0.084	0.106	0.088	0.099	0.092
ARW	0.093	0.121	0.104	0.101	0.102
Proposed	**0.074**	**0.102**	**0.08**	**0.09**	**0.086**
K-SVD	0.085	0.112	0.102	0.096	0.099

	rowing	sailing	snowboard	Avg.
SDS	0.111	0.133	0.129	0.111
K-med	0.11	0.117	0.118	0.104
AP	0.109	0.124	0.117	0.106
Greedy	0.104	0.129	0.126	0.104
ARW	0.110	0.133	0.132	0.112
Proposed	**0.095**	**0.109**	**0.109**	**0.093**
K-SVD	0.115	0.129	0.126	0.108

Table 15: Performance comparison of the proposed algorithm with 6 other baseline algorithms in terms of reconstruction error; 8 event categories selected from Event image set with equal summary size of 9.

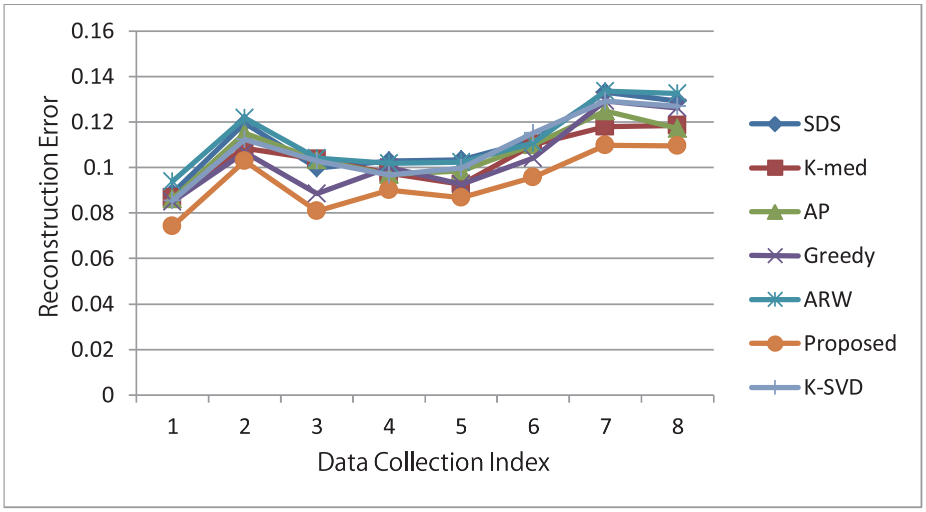


Figure 45: MSE comparison among the algorithms on Event image set with equal summary size of 9.

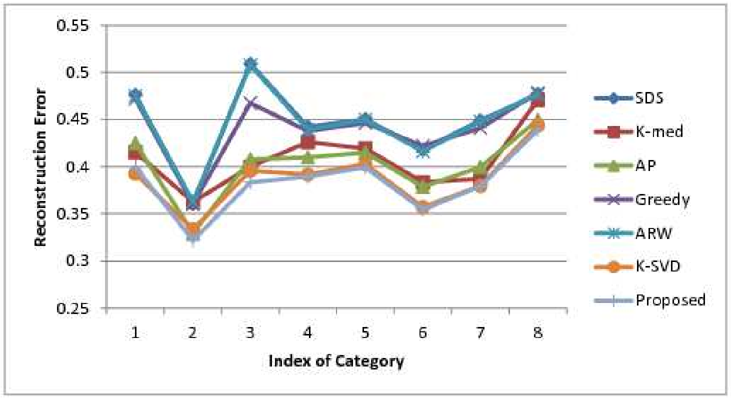


Figure 46: MSE comparison among all the 6 algorithms on the Torralba-8 dataset with GIST feature.

evaluation, compared to measuring metric such as SSIM [156].

As for the choice of the feature and the size of the image, we further conduct another experiment on the 8 scene categories of Torralba dataset [111], whose images have the same size (256 by 256). Although GIST feature [111] does not have a straightforward interpretation ability as SIFT BoW feature, we still consider it appropriate for scene image representation and use this feature for summarization task. The complete result is reported in Figure 46, and can be briefly concluded as: the average MSE value for the proposed algorithm is 0.3833, with K-SVD followed closely by 0.3871. The other 5 algorithms performs relatively poor in the range between 0.4 to 0.44. Similar to NUS-WIDE-SCENE and Event image data set, the Torralba 8 scene category data set is also consistent on visual content, the performance improvement of the proposed algorithm is not as significant as with object data sets. For this data collection, both K-SVD and the proposed algorithm can achieve close to optimal results. In conclusion, the consistency of visual content is the critical factor for summarization task, rather than the spatial layout or size of the image.

The initial choice of k random bases does not affect the final reconstruction performance. Our proposed algorithm has consistent reconstruction value with different inputs. We have used the clustering results from AP or k-medoids as the initial inputs and no significant difference is observed as compared with random initial inputs.

We also observed that the L1-norm sparse coding scheme can be used to replace the L0-norm sparse coding scheme. The coefficients are very sparse, and the majority of the weights concentrate on a few number of bases (2 or 3 bases in general; extremely small as compared with the size of the dictionary), which coincide with our assumption.

The sparsity penalty weight α and the diversity weight β may also affect the reconstruction value. We have tuned these two parameters, so that two constraint terms can contribute equally to the reconstruction function. We have tuned these two parameters under the following rules: a) the sparsity penalty weight α is determined first to make sure that each image is represented sparsely enough by the dictionary; b) we tune the diversity weight of β, so that the MSE curve decreases when the summary size is increased. The MSE curves under different β values are shown in Figure 48. The value of $\beta = 0.05$ (the middle curve in Figure 48) produces a balanced diversity term while other β values lead to unbalanced diversity terms. We also observed that the MSE curve (y-axis in Figure 48) decreases when the summary size increases (x-axis in Figure 48). This observation coincides with our assumption in Chapter 6.1.2 that the reconstruction ability will increase as the size of the summary increases. We also observed that most of the results strictly decrease the objective as the size of the dictionary increases, but there are still some outliers that do not fit the curve well. The reason is that the simulated annealing algorithm does not guarantee that the

Number of iteration	40
Diversity weight β	0.05
Number of different initials	3
MaxConseRej	20
MaxTries	40
Temperature decrease rate	0.9

global optimum is found every time (although it is close to the global optimum). If we can sacrifice the efficiency and repeat the learning process with more iterations, we can have a much higher probability to achieve the global optimum. In other words, the curve in Figure 48 can prove that our proposed algorithm finds the close-to-global optimum solution with high probability.

We will discuss about the convergence of the simulated annealing algorithm in this task. The use of annealing schedule is to make it possible to avoid local optima, and terminates the basis update stage in a limited number of steps. The newly accepted updates do not critically decrease the reconstruction error; the solution, which does not decrease the optimization function, still has a chance to be accepted, which makes it possible to jump out of a local minimum neighborhood. We have compared the proposed algorithm with greedy algorithm in terms of reconstruction error on a given data set with GIST feature. The result can be found in Figure 47. We observed that after the greedy algorithm converges at a local optimum position (blue dot), the SA algorithm (green dot) could still jump out of the local optimal neighborhood and find a better optimal solution. The reconstruction error curve is not smooth because the solution space is not continuous. Some important factors such as iteration number, number of attempts with different initials, cooling schedule, would all affect the convergence result. The optimal parameters are given as below:

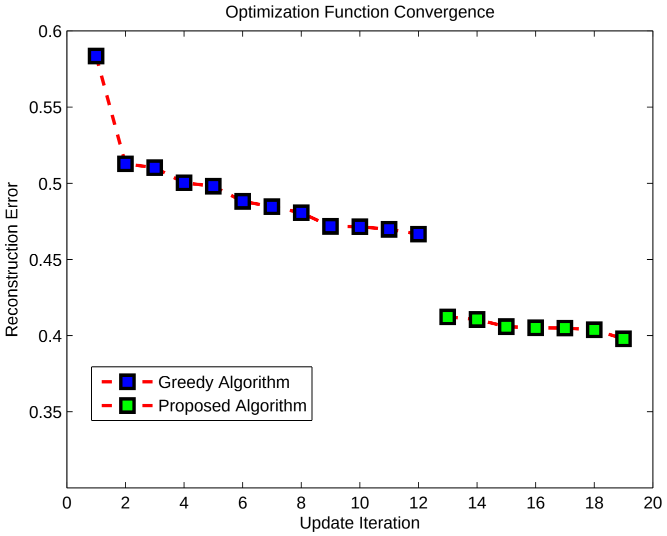


Figure 47: Optimization comparison in terms of reconstruction error: the blue dot is the greedy algorithm; the red dot is the SA algorithm. Only the updated steps are shown in this figure, thus the curve is not smooth.

We further tested how the number of iterations may affect our proposed algorithm and reported the summarization result for the category of "clouds" as in Figure 49. We have repeated the algorithm for 90 times and each time with different number of iterations. We have observed that the optimization function can find close to optimum solution when a certain amount of iteration is guaranteed.

Our proposed approach treated each image in the image collection equally for getting the summary, so there does not exist so-called "outliers" (a group of similar images that are far different from the rest of the data set). As a result, the summarization result may not coincide with the human perception of that image categories. For example, the "bakery" category in ImageNet contains a bunch of blank images which maybe the result of a broken download link. So the summarization results for our proposed approach can always include a blank image which are usually eliminated by some other algorithms.

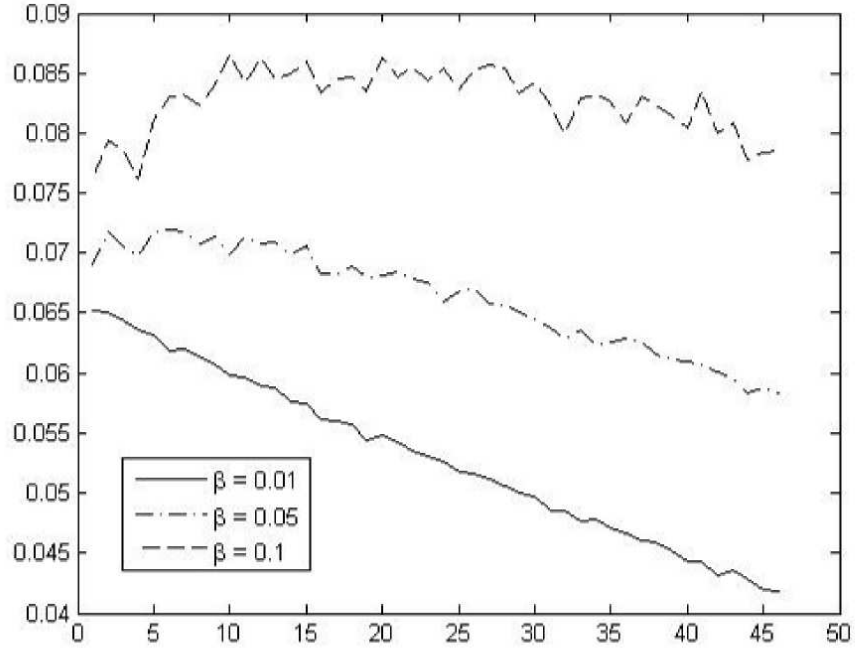


Figure 48: The MSE curve under different β value; x-axis represents the size of the summary, y-axis represents the MSE value.

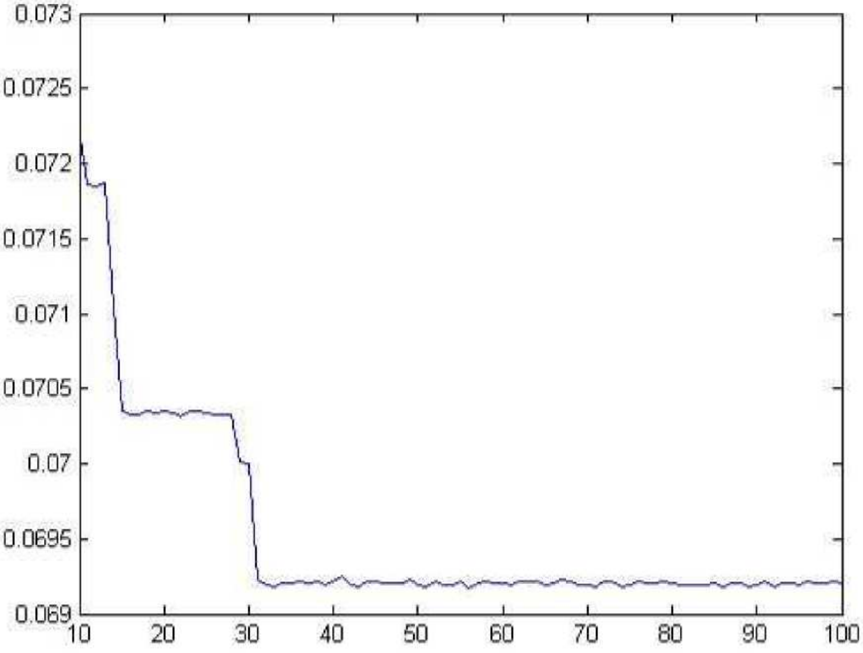
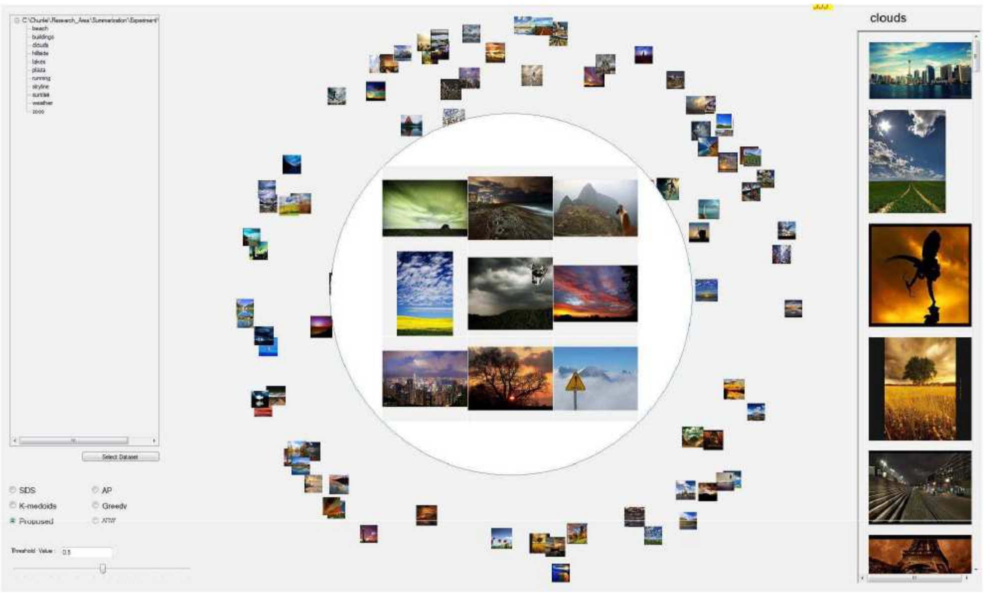


Figure 49: The MSE curve under different number of iterations; x-axis is the number of iterations of the proposed algorithm, y-axis represents the MSE value.

Computation efficiency: The computation cost of our proposed algorithm is largely affected by the annealing schedule which is used to determine the number of iterations. During each iteration, the most time consuming operation is to learn the non-negative sparse coefficients. In practical implementation, the simulated annealing stage terminates after 30 to 40 iterations and the overall computation time is around 2 to 3 minutes for each image set (with around 1400 images). By contrast, the simultaneous summarization learning algorithms such as AP and k-medoids take around 30 to 40 seconds. The ARW, K-SVD and Greedy algorithms has similar computation cost as compared with our proposed algorithm. The SDS algorithm runs slowest because it needs to examine the reconstruction performance for every image in the image set during each iteration. All these experiments are carried out in a 2.6G CPU and 4G memory computation environment.

Subjective evaluation: Image summarization is often task-specific, so the subjective results from user study are meaningful and also inevitable. We have performed user study to evaluate the effectiveness of our proposed approach and compared with other baseline approaches. The evaluation metric is measured by the users' feedback on how well the summarization results can recover the overall visual aspects for the original image set[3]. Our survey consists the following components : (1) 30 users (graduate students) are involved in this survey to investigate the summarization results for 3 image sets. (b) The system interface is shown in Figure 50. The users should be able to explore the image category list (left: treeview), the image set (right: panel), and

[3] The score ranges from 0 to 10, with 10 represents that all the visual aspects can be discovered by the summarization result. Visual aspects usually means salient objects or major scenes that are reflected in the original image set.



● Proposed

Threshold Value : 0.5

Figure 50: Screen shot of the system interface of category "clouds"; the algorithm and category names are not hidden in this case.

summarization results as given in the middle blob (summary size may vary according to user's demand) for all six algorithms (our proposed algorithm and other 6 baseline algorithms). (c) In actual survey, the category names are hidden from users because we do not want to distract users' judgment by involving their semantic understanding of that image category. The judgment should rely only on the visual aspects of the images. The algorithm names are also hidden from users to avoid biased opinion. (d) The average scores are reported in Table 16 – table 19. The results indicate that our proposed approach (via dictionary learning) has higher average appropriateness score as compared with other baseline algorithms, which coincides with the objective evaluation results.

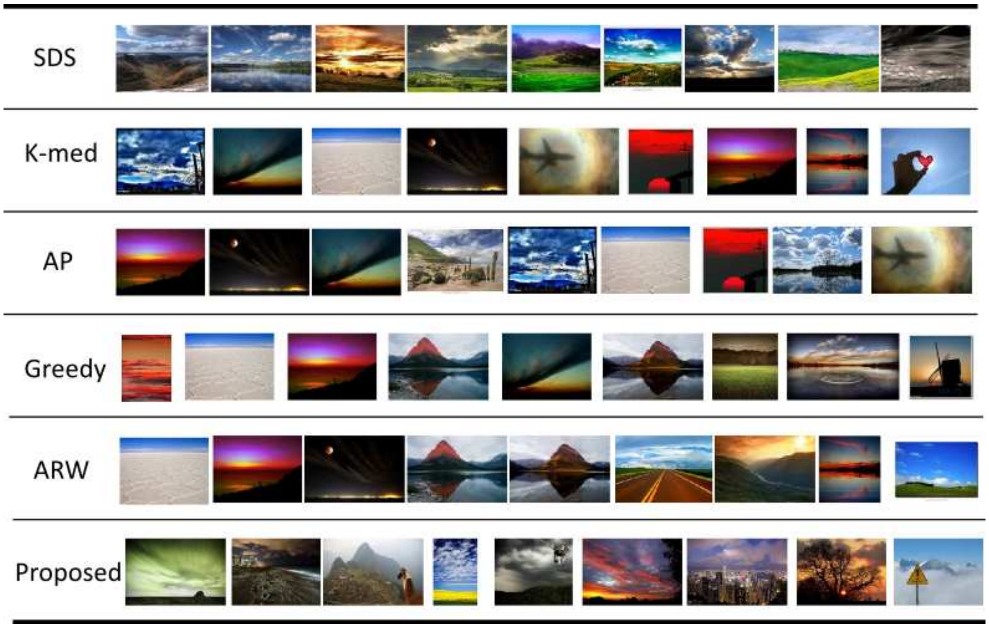


Figure 51: Summarization results for category "clouds" for the proposed algorithm and 6 other baseline algorithms (without K-SVD); A tile-view illustration of Figure 50.

	bakery	banana	bridge	church	cinema	garage	library	monitor	mug	pajama	schoolbus	skyscraper	mix	Avg.
SDS	6	6.9	7.5	7.3	7	6.9	6.3	7.1	6.7	6.4	**7.7**	7.9	7	7
K-med	6.2	7.6	7.4	7.2	7.4	7.5	6.9	6.9	6.6	7.3	7.1	8.4	7.2	7.2
AP	**6.7**	7.5	7.8	7.1	7.6	7.5	7.1	7.2	7.2	7.2	6.9	8.4	7.2	7.3
Greedy	5.8	6.3	6.9	6.6	6.7	7.1	5.9	6.8	6.5	6.8	7	8	7	6.7
ARW	5.7	6.3	7.3	6.9	7.4	7.8	6	6.1	7.1	6.6	6.6	7.7	6.9	6.8
Proposed	6.6	**7.8**	**8.1**	**7.9**	**8**	**8.2**	**7.7**	**7.4**	**7.7**	**7.8**	7.5	**8.5**	**7.3**	**7.7**
K-SVD	6.6	6.7	7.1	7.1	7.2	7.5	6.9	6.5	6.6	7.2	7.3	7.2	7	7
Size	10	26	33	34	25	20	16	31	27	18	22	37	31	N/A

Table 16: Subjective evaluation of the proposed algorithm with 6 other baseline algorithms in terms of user grading on ImageNet with different summary size

	bakery	banana	bridge	church	cinema	garage	library	monitor	mug	pajama	schoolbus	skyscraper	mix	Avg.
SDS	6.4	5.8	5.9	6.7	6	5.8	5.3	6.9	7	6.4	5.3	5.4	**8.5**	6.0
K-med	7.2	5.9	6.7	**8.9**	6	6.2	5.1	6.8	7.4	8.5	7.7	8.1	5.7	7.0
AP	7.3	5.7	**8.6**	5.4	**7.8**	7	7.9	7.1	6.5	8.1	**7.8**	**8.6**	7.9	7.3
Greedy	**8.7**	6.2	5.6	5.7	6.6	5.4	6	7.3	5.9	**8.9**	6	7.6	6.3	6.6
ARW	8.5	6.8	5.9	8.6	7.3	6.1	8.2	5.9	6.9	5.1	6.3	6.9	7.7	6.8
Proposed	7.4	**8.3**	6.2	6.7	7.4	**7.6**	**8.7**	**8.8**	**7.7**	8.6	5.5	7.8	5.1	**7.5**
K-SVD	7.3	7.7	5.9	6.1	6.6	6.7	7.2	8.2	7.2	5.9	6.3	6.2	6.1	6.7

Table 17: Subjective evaluation of the proposed algorithm with 6 other baseline algorithms in terms of user grading on ImageNet with equal summary size of 9.

	beach	building	clouds	hillside	lakes	plaza	running	skyline	sunrise	weather	zoos	Avg.
SDS	7	8.2	6.9	5.2	**8.2**	7.8	5.3	5.2	6.1	6.4	5.2	6.5
K-med	6.9	7.3	5.6	**7.7**	**8.2**	7.5	5.5	6.5	6.7	5.7	7.9	6.8
AP	7.4	8.5	7	7	7.6	6.7	**8.3**	7.6	5.6	7.8	7.1	7.3
Greedy	**8.6**	5.7	**8.9**	5.1	7.8	8.2	5.6	7.1	5	6.9	6	6.8
ARW	7.4	5.9	7.8	5.2	5.5	6.8	6.5	6.6	6.9	6.3	6.6	6.5
Proposed	8.4	**8.6**	6.8	5.3	7	**8.3**	8.2	**8.5**	**7.4**	**8.6**	**8.7**	**7.8**
K-SVD	7.4	8.2	7.1	7.5	6.7	7.7	5.2	5.9	6.8	6.6	6.4	6.8

Table 18: Subjective evaluation of the proposed algorithm with 6 other baseline algorithms in terms of user grading on NUS-WIDE-SCENE with equal summary size of 9.

	rockc	badm	bocce	croq	polo	rowi	saili	snowb	Avg.
SDS	6.2	5.7	**8.5**	5.6	6.9	6	7.9	8.2	6.8
K-med	7.8	5.5	7.6	8.4	5.4	6.1	6.3	6	6.6
AP	6.6	7.7	5.1	6.8	5.7	6.5	8.2	**8.6**	6.9
Greedy	7.6	**8.9**	5.7	7.5	**7.3**	**7.4**	7.3	7.3	7.3
ARW	7.1	5.6	6.4	6.5	5.9	6	5.4	5	5.9
Proposed	**8.9**	7.6	7.2	**8.9**	6.7	7.3	**8.5**	8.5	**7.9**
K-SVD	8.4	7.7	5.6	6.7	7.2	6.9	8.1	6.8	7.2

Table 19: Subjective evaluation of the proposed algorithm with 6 other baseline algorithms in terms of user grading on Event image set with equal summary size of 9.

CHAPTER 7: A PRACTICE: GRAFFITI IMAGE RETRIEVAL

In this Chapter, we will introduce a novel information retrieval system of graffiti image retrieval. The proposed graffiti retrieval system comprises two major components: character detection, and string recognition/retrieval. The string recognition/retrieval component is further broken down by the image-wise retrieval and semantic-wise retrieval. The work-flow of the entire system is shown in Figure 52. In this section, we will describe the design detail of the steps, as shown in the framework diagram. We will use the top left image in Figure 8 as an example input throughout this Chapter.

7.0.1 Image Preprocessing

We have some basic requirement for the quality of the images. The character components should be contrasting from the background and the background is not extremely cluttered or colorful. Otherwise, the graffiti lost its meaning to pass on messages. For preprocessing of the images, we conduct a series of sequential operations, including image resizing, grayscaling, and smoothing. The sizes of the collected graffiti images are usually large (larger than 2000 by 1500), which is difficult to display and inefficient to process. Therefore, we keep the aspect ratio and resize the image to make sure its largest dimension is smaller than 800 pixels. An image of this size shows clear graffiti characters and is small enough for efficient processing. The

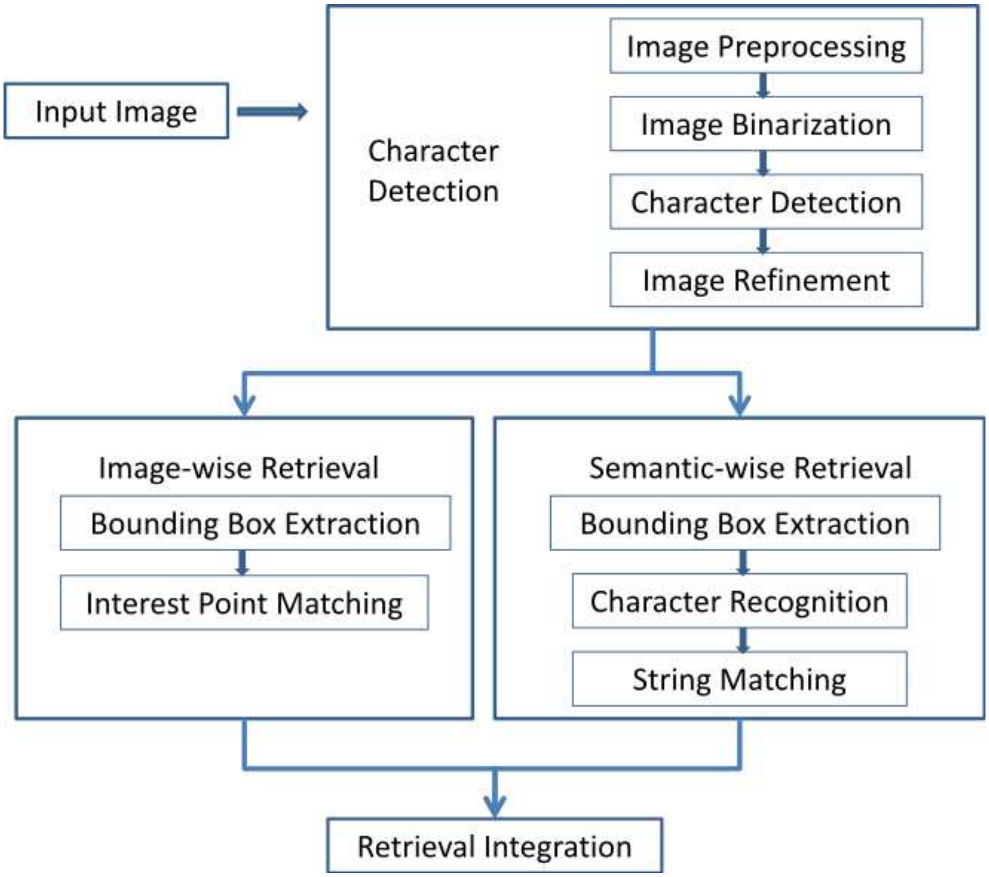


Figure 52: Graffiti retrieval framework

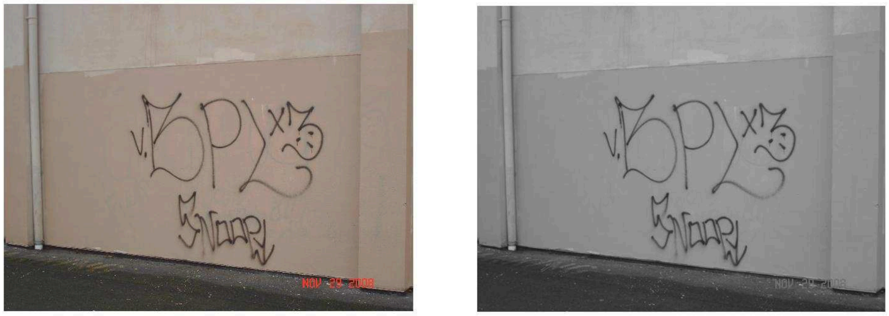


Figure 53: Image preprocessing. Left: original image; right: image after preprocessing

resized image is then changed to gray-scale[4] and smoothed with a 5 by 5 Gaussian filter. The smoothing operation dramatically reduces large amounts of unnecessary background noise. The image after pre-processing is shown as in Figure 53.

7.0.2 Image Binarization

The grayscale image has pixel values ranging from 0 to 255. For the purpose of character detection, we need to partition the image into the potential object area (the character area) and the background area, which is a binarization process. Image binarization is realized by the global thresholding algorithms such as Niblack [179]. The intensity of the pixel of the input image is compared with a threshold of T; the value above the threshold is set to white (1; the potential object area pixel), otherwise

[4] We have also tested using the H component from the HSV color space, which is known to be a more robust visual attribute to pixel intensity variation, hence the lighting variation; and we observed very similar results compared to grayscale framework on RGB color space.

black (0; the obvious background pixel). The Niblack's algorithm calculates a pixel-wise threshold by sliding a square window on the grayscale image. The size of the window is determined by the size of the image, which is based on the fact that the character components are visible and thus occupy a certain proportion of the image area. The threshold T is calculated with the mean m and standard deviation s on each of the sliding windows. The pixel intensity is compared with threshold T, calculated as:

$$T = m + k * s \tag{69}$$

where k is a positive number between 0 and 1, if we are detecting white characters on black background or k is a negative number between -1 and 0, if we are detecting black characters on white background. In the scenario of graffiti detection, we have observed cases of dark ink characters on a light colored surface and vice versa, so we are actually conducting pixel-wise comparisons with both thresholds:

$$T_{1,2} = m + k_{1,2} * s \tag{70}$$

where $k_1 \in [0, 1]$ and $k_2 \in [-1, 0]$. The parameters, such as the size of the sliding window and the value of two k, will affect the binarization results. Because of the various visual representations of the large number of images in the data set, we may predict that there is no global configuration that can fit all the data. As a result, we determine the parameters by specific input images; for example, the size of the sliding window is based on the size of the input image and the value of k is based on the entropy of the image, specifically, linearly correlated with the entropy value. We can

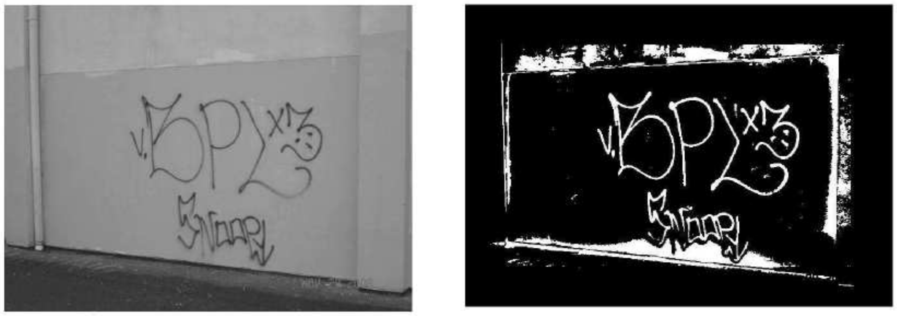


Figure 54: Image binarization: Left: image after preprocessing; right: image after binarization

see from Figure 54 that the Niblack algorithm will delete a large area of background patches that have a smooth visual appearance and keep the object areas that always appear with a high standard deviation of intensity.

7.0.3 Character Detection

The binary image is organized by the connected components that are recognized as candidate objects. These candidates could be either the actual graffiti characters or the noisy background patches that cannot be deleted from the previous steps. The goal of the object detection task is to delete all these distracters and retain as many positive candidates as possible. We find that several visual attributes of character objects differ from the background objects, and the most important attribute is the edge contrast, with the idea derived from [179]. Edge contrast is defined as follows:

$$T_{edge\_contrast} = \frac{\{border\_pixels\} \cap \{edge\_detection\}}{\{border\_pixels\}} \tag{71}$$

The above threshold is defined based on the observation that character objects' border pixels have a large portion of overlapping with the edge detection result from the original image, while the borders of background objects do not overlap much with

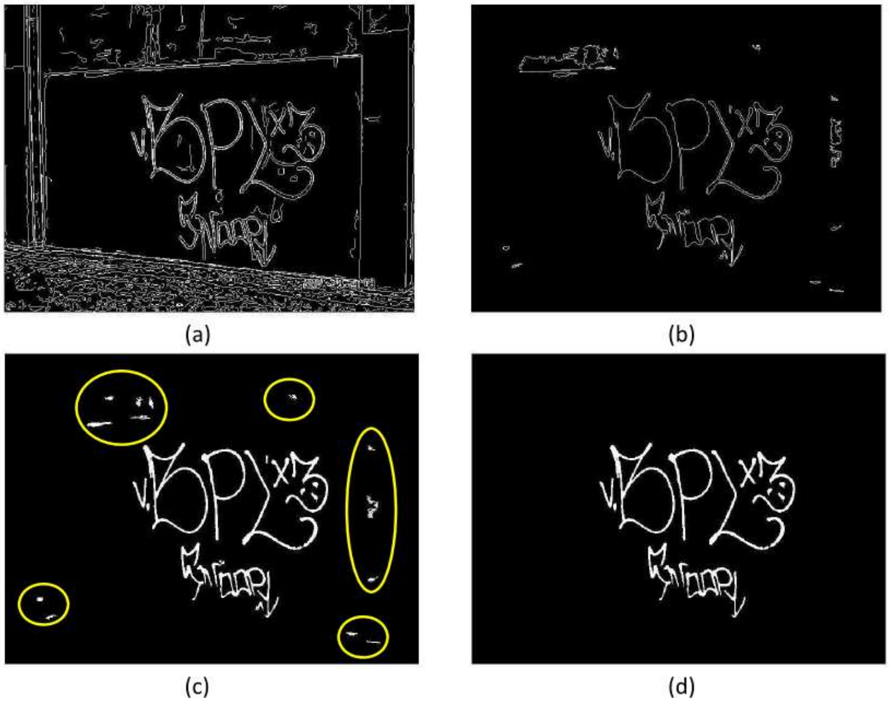


Figure 55: Edge contrast. (a) edge detection; (b) border detection; (c) noisy patches (in yellow circle) before comparing edge contrast; (d) after elimination.

the edge detection result. We can easily observe this property in Figure 55. Figure 55, (a) and (b) show the edge detection result and border detection result respectively. We can also see that the character components coincide with each other in (a) and (b) while the background components do not. Therefore, we will delete all the connected components whose edge contrast value is smaller than this threshold $T_{edge\_contrast}$ and the image is further refined as shown in Figure 55 (d).

Other attributes, such as the aspect ratio, length ratio, size ratio, border ratio, number of holes, smooth ratio, skeleton distance, and component position may also differentiate the positive objects from the noisy objects. Below we briefly introduce the functionality of the above criteria:

1) Aspect ratio: The printed characters usually have the aspect ratio of 7:5 or

other ratios close to this value, which relies on the font or style of the character. The graffiti characters, with no exception, will follow this approximate aspect ratio. So we can exclude those connected components with much larger or smaller aspect ratios, because they are very unlikely to be characters.

2) Length ratio, size ratio, and position: Several background objects in the graffiti images can be deleted based on their extreme length or size. The character components in the graffiti images usually appear in formal shape and will locate as the focus of the photo. Extremely large or long components and components on the border of the images are usually noisy components. These noisy components often are windows or door frames.

3) Number of holes: Image patches from the background may have very rough textures or crude surfaces. The components derived from these areas have an irregular pattern with lots of holes or loops. The components derived from characters, on the other hand, have a more stable and consistent pattern with a limited number of holes. We will empirically define a threshold T_{holes} to exclude components with too many inner loops.

4) Smooth ratio: Graffiti characters are painted with oil or ink, and the oil paint itself is rough regardless of what kind of surface it is painted on. On the other hand, the background patches could be part of a very smooth surface. We define the smoothness of a connected component by its standard deviation value. The graffiti components show a moderate level of smoothness as indicated by a modest standard deviation; while some background components show perfect smoothness indicated by a near-zero standard deviation, which means the intensity values throughout the

components are almost the same. We thus can exclude those components with a very small standard deviation value, because they are very unlikely to be a graffiti component.

5) Border ratio: The refinement criteria of border ratio are derived directly from the field of traditional character recognition. The characters, whether they are handwriting or graffiti, are composed of strikes, and the shape of the strikes is different from random patches. If the border ratio is defined as the proportion of border pixel to the total pixel, the components of strikes should have a much larger border ratio than random patches. Therefore, we will exclude the components with a small value of border ratio because they are very likely to be background components.

6) Skeleton distance: The notion of skeleton distance is also derived from the traditional character recognition field. We first conduct the inside loop filling operation as introduced in the following subsection, then extract the skeleton of the components, and further calculate the distance for each of the skeleton pixels. The distance of a skeleton pixel is defined as the minimum distance of the skeleton pixel to a pixel that is not in this component. Next, we gather the mean and deviation statistics of all the skeleton distances. If the component is a character, then the mean and deviation of the skeleton distance should both be small because of the consistent thickness of the strikes. Otherwise, it is more likely to be a noisy component.

The above background exclusion criteria are used sequentially and lead to a joint result that excludes all of the background components and retains as many character components as possible. For the threshold value used in each detection criteria, we conservatively select the one that keeps all the positive components and eliminates

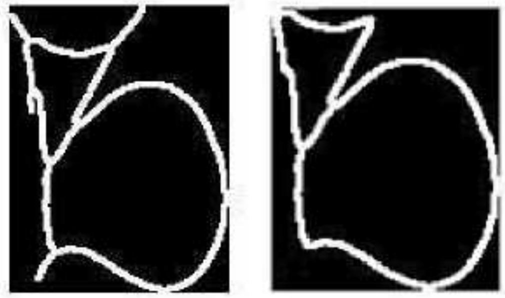


Figure 56: Unnecessary branch cut

as many negative components as possible for all the images.

7.0.4 Image Refinement

The extracted character components need to be further refined to better serve the future steps of recognition or matching. A series of techniques is designed and introduced as follows:

1) Inside loop filling: Even though we have excluded the background components that have large numbers of inner loops, the remaining character components will inevitably have holes due to the rough quality of the painting. The oil paint or ink is not thick so small spots will be left unpainted within the strokes of the character. We apply the filling algorithm that detects the small holes inside the strokes and fill the holes. This step is essential for the later step of skeleton extraction, because the small holes inside the stroke may cause unnecessary branches of the skeleton.

2) Skeleton extraction: We use the thinning algorithm [84] to extract the skeleton structure of the character. We set the number of iterations to infinite so that the iteration repeats until the image stops changing and results in a single-pixel-width

skeleton. If we define the degree of a pixel as the number of non-zero pixels from its 8 neighbors, then we can further define pixels in the components as endpoints if their degree equals 1, inline points if their degree equals 2, or junction points if their degree is more than 3.

3) Unnecessary branch cut: For certain printed uppercase English letters, there are at most 4 endpoints, such as the letters “H” and “K”. On the other hand, the skeleton extraction results usually have a much larger number of endpoints. The large number of unnecessary branches is usually caused by the skeleton extraction results from raw edges of the original character. The branches are defined as the edges linked by an endpoint and a junction point, so we examine all the branches and compare their length with the neighboring edges and longest edge. We then cut the branches shorter than a threshold because they are very likely to be unnecessary branches. A sample branch cut result of letter “B” is shown in Figure 56.

4) Background stripe elimination: Background stripes (as shown in Figure 57 (a)) have a very similar pattern to the character strokes, so they usually cannot be eliminated during the initial character extraction stages (as shown in Figure 57 (b)). These background stripes usually come from some solid background structure such as the edges and frames of the architecture. We choose to use Hough transform because it is a good detector of straight lines. The Hough transform line detection results are shown in Figure 57 (c) in green. We then apply algorithms to eliminate the four detected horizontal lines without breaking the vertical character strokes, as shown in Figure 57 (d). Specifically, we delete all the pixels connected along the detected hough lines, then reconnect the components which are originally connected, such as

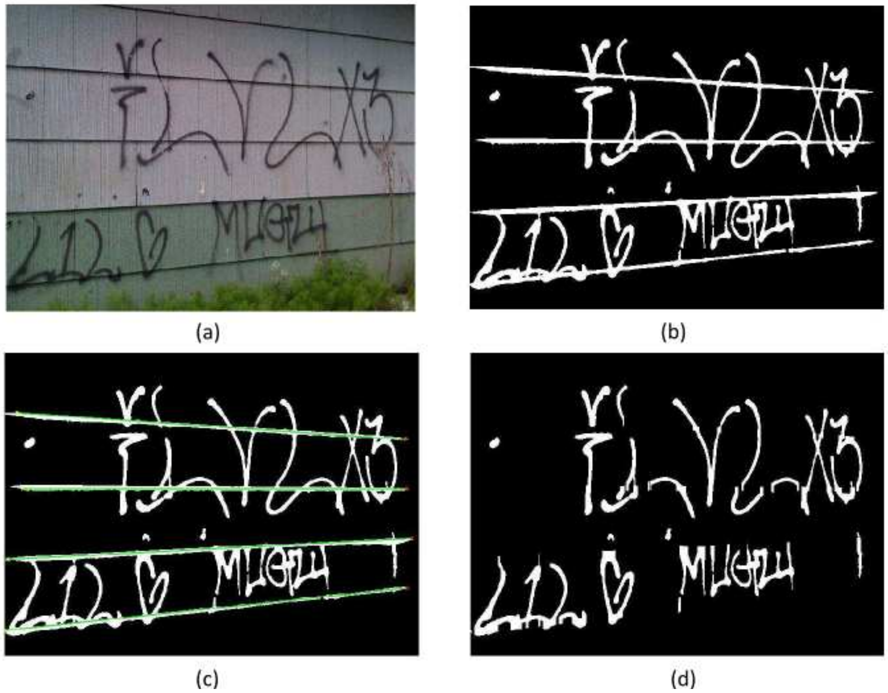


Figure 57: Background stripe elimination example

the separated vertical stroke.

7.0.5 Graffiti Retrieval

The key operation that links the character detection process to graffiti retrieval is to effectively bound each of the individual connected components (candidate characters) into meaningful strings with a larger bounding box.

The left image in Figure 59 is the bounding box result for each of the individual connected components. We can see that each character is bounded by a single box; however, the retrieval result of each character doesn't help the retrieval of all the graffiti images. Similar to OCR, we are seeking meaningful character sets, or a string of characters, that can be considered as a proper retrieval unit. We have proposed rules to combine multiple geographically aggregated components into a larger component, such as components close enough to each other in horizontal direction. Specifically, we merge two individual components together into a larger bounding box if the y coordinate value of the center of one component falls into the range of the other component in y direction. Then we repeat this operation until no more components are added in. The combination results are shown as the right image in Figure 59. We can see the proposed combining rule results in two strings, which are “vBPLx3” and ”Snoopy”. The proposed rule does not apply to characters that are written in a vertical or diagonal direction.

After this step, any traditional OCR techniques, such as handwriting recognition techniques, can be applied to recognize the characters in the extracted strings. The characters are recognized based on the individual connected components extracted as in Figure 59: left. Then the recognition results of each character are organized together based on the string extracted in Figure 59: right in horizontal order. We are using the template matching method that matches the character patch with each of the templates (0-9, A-Z and a-z, created as universal template [113, 106]) and find the best match. Sample character matching templates can be found in Figure 58. The matching score, or the semantic-wise retrieval score, between two strings is defined as the length of the Longest Common Subsequence (LCS),

$$D_s(s,t) = |LCS(s,t)| \tag{72}$$

where s and t are two strings. The LCS does not merely count the frequency of appearance of the character; it also requires the sequence of the appearance of the corresponding character to be the same. The proposed metric is more reasonable for

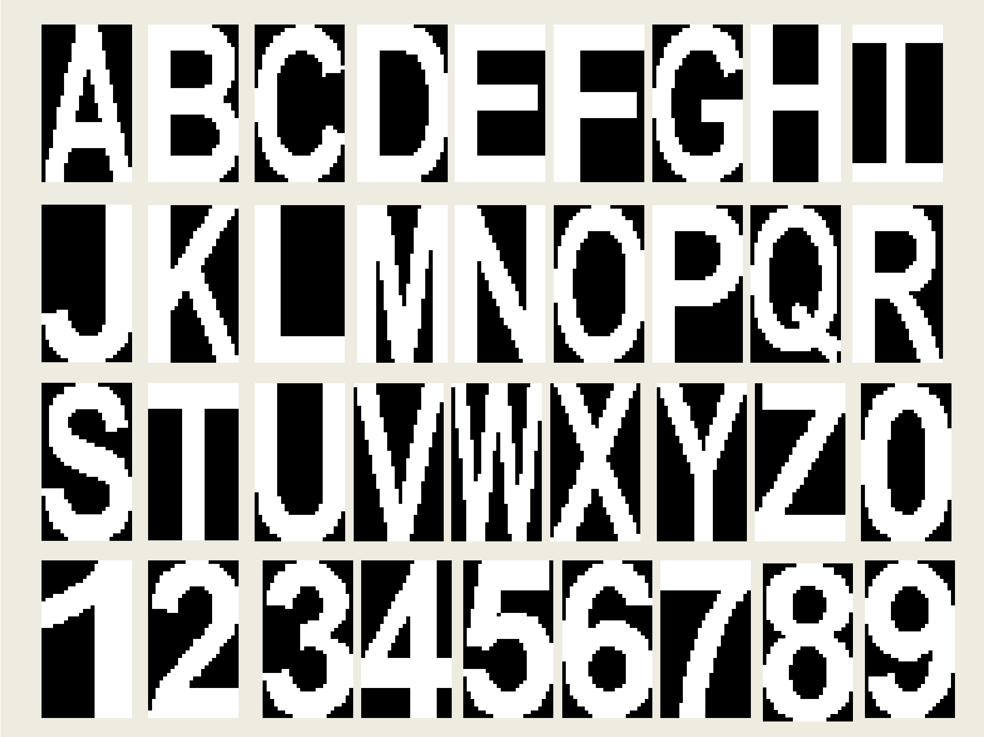


Figure 58: templates for character matching: A-Z and 0-9.

the semantic-level matching of graffiti words. Other types of recognition techniques can also be applied here; however, these are not the focus of this work.

For image-level matching of the two string components, we count the number of matches of interest points in the two string patches as the retrieval score. We have found that this matching metric performs better than having the score normalized with the total number of interest points detected. For one interest point k in string patch S_i, we will calculate the Euclidean distance of the SIFT descriptor [5] [96] from k to all the interest points in the other string patch S_j, and find the closest distance d_1 and the second closest distance d_2. A match is considered to be found if the ratio d_1/d_2 is smaller than a threshold (0.7 in this work). We count the total number of

[5] SIFT descriptor is known to be scale and rotation invariant, thus a suitable descriptor for local texture matching. A 128-dimensional feature vector is used in this experiment.

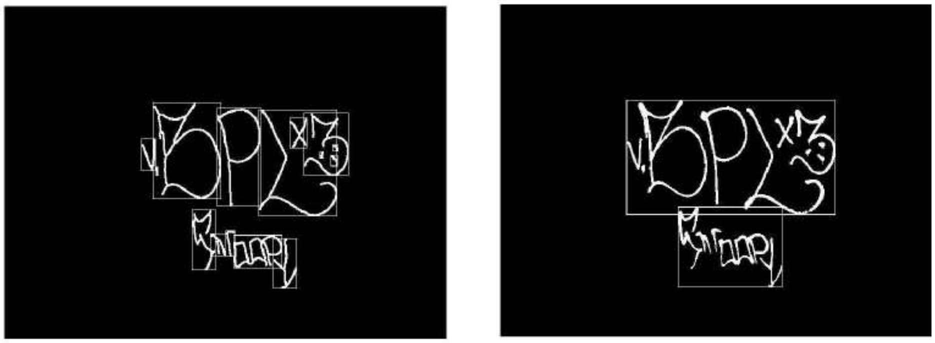


Figure 59: Semantic string construction with bounding box

matchings between the interest points of two string patches.

$$D_i(s,t) = |Match(s,t)| \tag{73}$$

There are two major benefits for the proposed interest point matching scheme compared to the traditional interest point matching scheme that is conducted on the entire image. First, the interest points in the proposed framework are only extracted from the neighborhood of the character components as discovered in Figure 59. Such design will dramatically decrease the influence of the interest points from the background as shown in Figure 60. The matching score for the top image is 113 and reduced to 71 for the bottom image. We have observed false-positive matches from the top image in Figure 60, such as matches on date tags from the camera, matches from background trees, and false matches of the object. Such matches are eliminated with the proposed framework as shown in the bottom image. Second, the number of interest points in the bounding box is much smaller than in the entire image; thus, the number of comparisons and computation time are dramatically reduced.

The matching score $R(i,j)$ between two images therefore can be represented by the

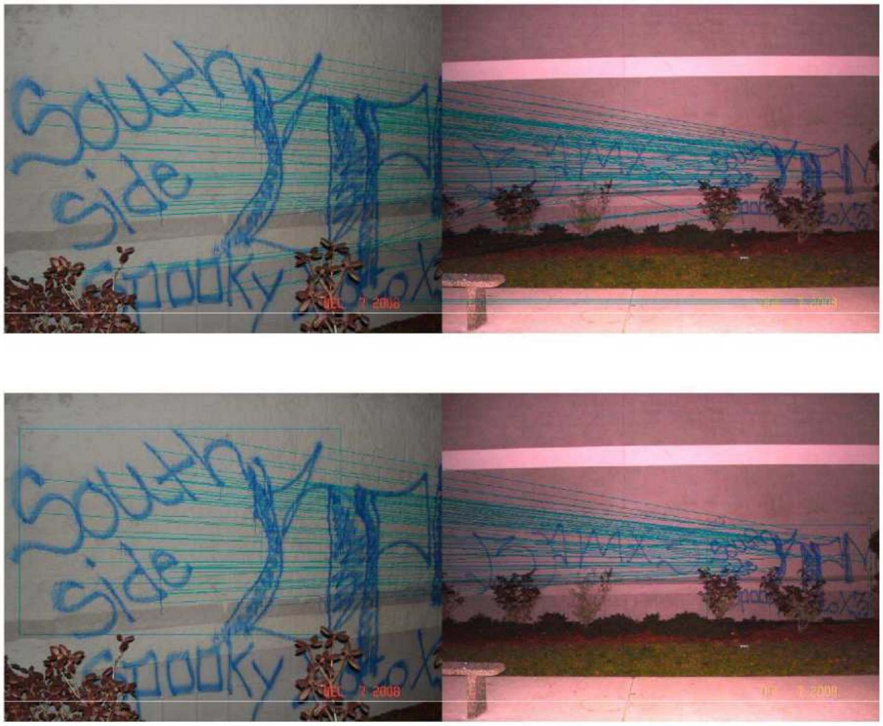


Figure 60: Top: Interest point matching on the entire image with matching score 113; Bottom: interest point matching on the string patch (bounding box area) with matching score 71.

maximum matching score between the string pairs from the two images. Specifically,

$$R(i,j) = \max_{s \in I_i, t \in I_j} (\alpha \overline{D}_i(s,t) + (1-\alpha)\overline{D}_s(s,t)) \tag{74}$$

where $R(i,j)$ is defined as the maximum matching score of the string pairs from two images. $\overline{D}_i$ and $\overline{D}_s$ are normalized image-wise retrieval score and semantic-wise retrieval score across all the available images in the database. I is the string set for a specific image. $\alpha \in [0,1]$ is the weight for image-wise retrieval score. α is learned by maximizing the accumulated matching score across all the correct matches; while minimizing the accumulated matching score across all the false matches.

7.1 Experiment and evaluation

The graffiti database used in the paper was provided by the law enforcement community of the Pacific Northwest. 62% (120/194) of the images in the database have clear character component detection; 38% (74/194) do not have clear detection of the character components, which means either the characters are eliminated with the proposed image refining methods or they cannot be distinguished from the background. Specifically, 4%(8/194) of the images do not have any textual parts, or the textual area is not visible.

We have built an interactive interface to conduct the retrieval operation as shown in Figure 62. The user may upload a query image; the system then performs character detection and shows the binary result in the left column. Next, the user may start the retrieval operation on the given database and get the top 15 retrieval results on the main panel. There are currently 194 graffiti images in the database and more are expected to come. The ground truth is constructed by human labor to find all the matching pairs or groups. The ground truth includes 14 extracted queries, with each query image having 1 to 4 matches in the database. The cumulative matching accuracy [105] curve is used as the evaluation metric with each value in the graph representing the average accumulated accuracy on a certain rank. The cumulative matching accuracy on a specific rank is calculated as the number of correctly retrieved matches on and before this rank divided by the total number of ground truth. Therefore, this curve is monotonically increasing along the axis of rank. The experiment results are shown in Figure 63. The proposed bounding box framework achieves an average of 88% on cumulative accuracy on rank top 8, while the image-wise framework achieves an average of 75% on cumulative accuracy on rank top 8. These results show the advantage of the proposed framework on cumulative retrieval accuracy. Both frameworks achieve similar performance on rank top 1. More results of the proposed framework can be found in Table 20.

It is easy to understand why matching on bounding box framework outperforms the matching on the entire image. The query image may share some similar background patches with an unrelated graffiti image in the database; thus, there could be a large number of false-positive matches coming from the background to overwhelm the matching of the actual character area. Similar background patterns can be easily found in graffiti images. It is therefore essential to extract the meaningful character components from the background. Figure 61 shows the comparison result between the two framework on the example image.

On the other hand, the improvement in computation efficiency for the proposed bounding box framework is also noticeable. Consider the query image in Figure 62. The number of interest points of the query image and top 1 retrieval result are 1425 and 2425 respectively; after bounding box extraction, the number of interest points in the best matched bounding boxes are 75 and 87 respectively. As a result, the number of actual key comparisons is reduced to less than 1/400 of the original scale under the new framework. Such improvement is significant in large-scale image retrieval tasks.

The semantic retrieval score contributes less than the image retrieval score. This is because of the inherent difficulty of the semantic-level understanding of the graffiti characters. The character recognition results for the two strings in the example image

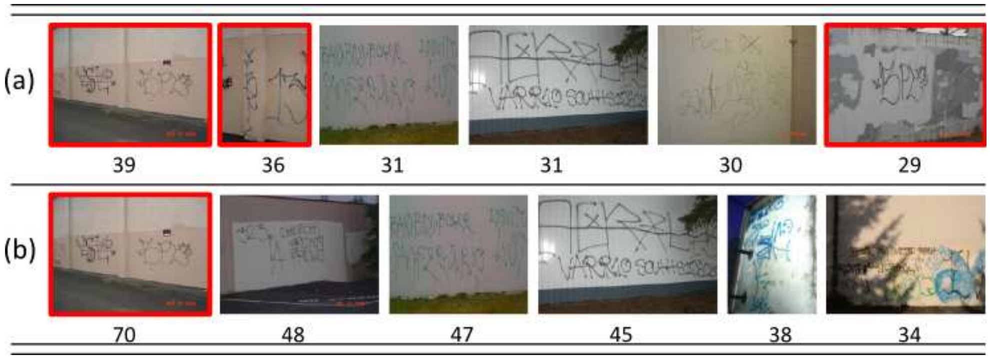
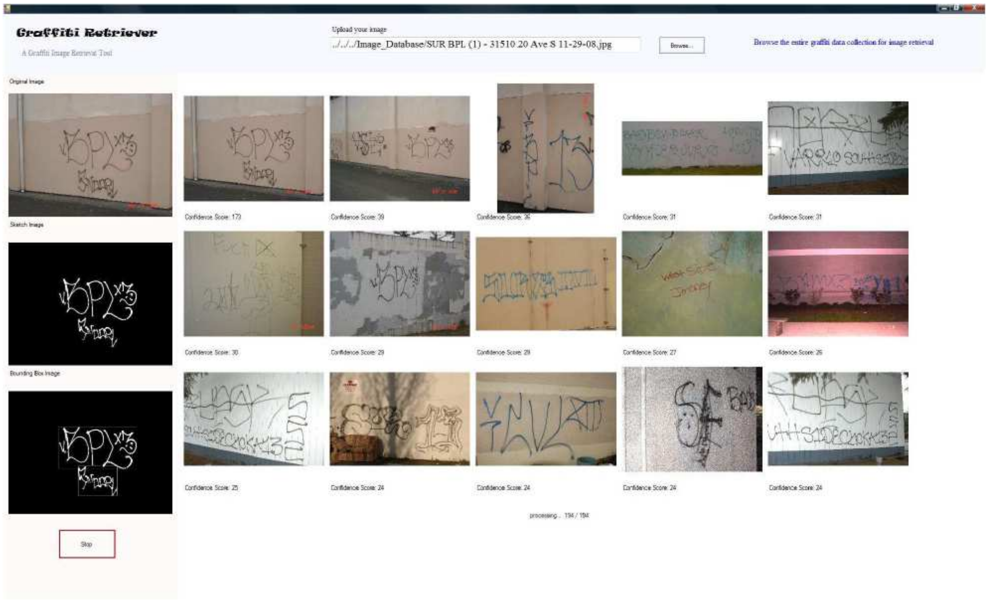


Figure 61: Top 6 retrieval results under two frameworks for the query image in Figure 53: Top row (a) is derived with the bounding box framework; bottom row (b) is derived with the image-based framework. The correct matches are circled with red boxes and corresponding matching score is indicated below each image.


Graffiti Retriever
A Graffiti Image Retrieval Tool
Browse...
Browse the entire graffiti data collection for image retrieval
Stop

Upload your image
../../../Image_Database/SUR BPL (1) - 31510 20 Ave S 11-29-08.jpg

Original Image

Sketch Image

Bounding Box Image

Confidence Score: 173
Confidence Score: 39
Confidence Score: 36
Confidence Score: 31
Confidence Score: 31
Confidence Score: 30
Confidence Score: 29
Confidence Score: 29
Confidence Score: 27
Confidence Score: 26
Confidence Score: 25
Confidence Score: 24
Confidence Score: 24
Confidence Score: 24
Confidence Score: 24

processing... 194 / 194

Figure 62: Interactive system screen shot. Top: upload menu; Left: query image processing result; Main: top 15 retrieval results

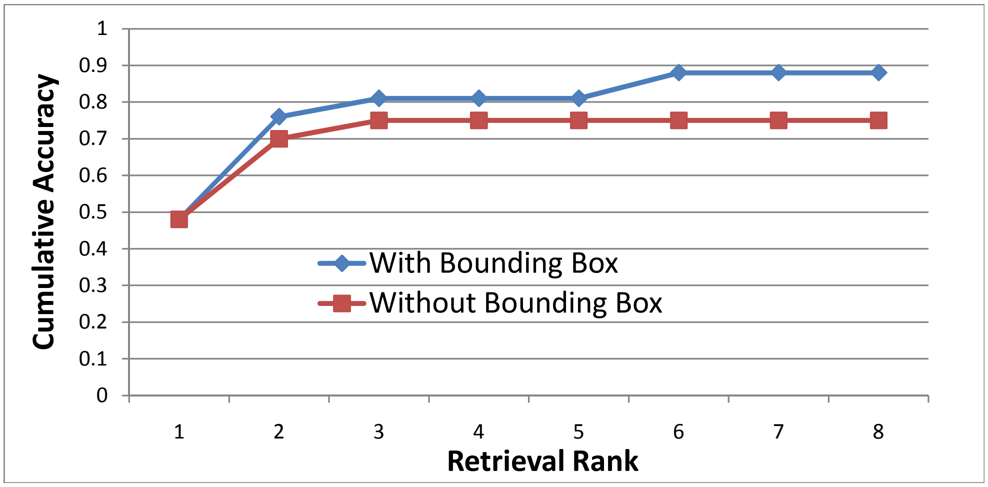


Figure 63: Comparison between cumulative accuracy curve (CAC) with bounding box framework and CAC without bounding box framework.

are “vKPLX?” and “VXVJAR” . The maximum semantic retrieval score is 3 in this case (between “vKPLX?” and “?PLx3” from the top 1 retrieval result). The symbol “?” indicates a failed detection; in other words, there is not enough confidence to assign any value. This value is not as convincing as the image-wise retrieval score based on the current result. Correspondingly, the image-retrieval score will dominate the final matching function of Equation 74 For example, we got a semantic-level score of 3 and image-level score of 36 for the top 1 score in Figure 62. The improvement achieved by integrating the semantic-wise retrieval score can be found in Figure 64.

Strengths and weaknesses: The proposed system achieves better retrieval performance compared to solely applying either image-base retrieval or OCR-related retrieval. The bounding box framework not only improves the accuracy of the local feature matching but also reduces the computation burden by eliminating unnecessary interest points. Reducing the number of false matches by applying geometric constraints is another well known technique to improve matching and retrieval results.

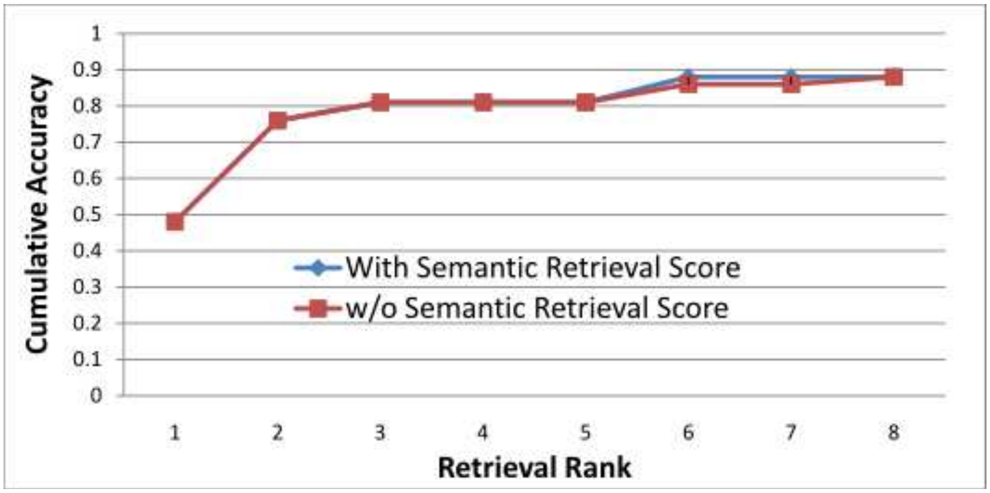


Figure 64: Comparison between cumulative accuracy curve with semantic retrieval score and cumulative accuracy curve without semantic retrieval score

However, based on current scale of database, we didn't observed prominent improvement by applying the geometric constraints. The semantic-level understanding of graffiti images, on the other hand, is not equally satisfactory, as shown in Figure 64. It requires us to bring in better semantic understanding techniques without sacrificing the computation efficiency. This weakness suggests a path for future work on the graffiti retrieval task as described in the next section.

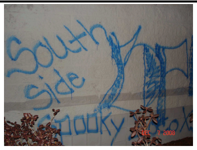
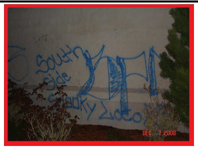
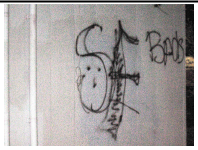
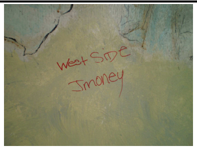
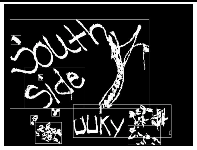
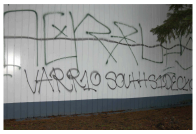
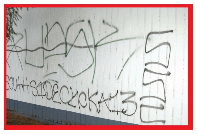
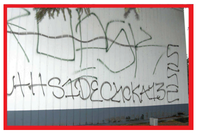
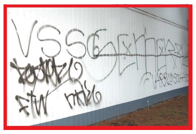
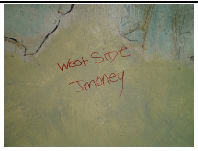
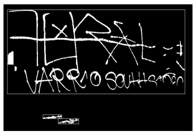
 query 1	 76	 35	 33	 27
 character extraction	 26	 26	 26	 26
 query 2	 76	 73	 48	 46
 character extraction	 44	 42	 42	 38

Table 20: Two query examples under the proposed framework: The left 2 images in each case are the query image and character detection result; the other 8 images are the top 8 retrieval results, listed in decreasing order with regard to the matching score. The correct matches are bounded with red boxes.

CHAPTER 8: CONCLUSION

In this thesis, we have developed novel frameworks and algorithms for the research tasks of junk image filtering, near duplicate detection, concept network generation and organization, image collection summarization and graffiti image retrieval. The listed components sequentially built up an interactive large-scale image collection system, which enables efficient exploration, recommendation and information retrieval tasks.

Firstly, for junk image filtering, a novel bilingual inter-cluster correlation analysis algorithm is developed for integrating bilingual image search results for automatic junk image filtering. To achieve more accurate partition of the returned images, multiple kernels are seamlessly integrated for diverse image similarity characterization and a K-way min-max cut algorithm is developed for achieving more precise image clustering. To filter out the junk images, the returned images for the same keyword-based query (which are obtained from two different keyword-based image search engines) are integrated and inter-cluster visual correlation analysis is further performed to automatically identify the clusters for the relevant images and the cluster for the junk images. Experiments on diverse keyword-based queries (5000 bilingual queries in our current experiments) from bilingual image search engines (Google Images in English and Baidu Images in Chinese) have obtained very promising results.

Secondly, we proposed a cascading coarse-to-fine model for near duplicate detection on large scale data set. We applied clustering method to roughly partition the data set based on global features (color histogram); then conducted pair-wise image comparison within the clusters with more complex local features (SIFT and CPAM BoW model) for the accurate detection. For the proposed model, experiment results have shown the correctness of the design, as well as the efficiency of the computation.

Thirdly, we benchmarked multiple approaches for feature extraction and image similarity characterization for the first step, which are very important for image/video retrieval community. We obtained multiple impressive results:(a) simple image segmentation via partition-based partition may significantly enhance the discrimination power of visual features; (b) different visual features may play different roles for different image classification tasks and thus integrating feature subset selection with image classifier training may significantly improve the performance of the image classifiers; (c) combining multiple feature subsets and their kernels may improve the discrimination power of the image classifiers as well. With partition-based feature extraction framework and multi-kernel similarity characterization, we generate the concept network based on their inter-concept visual correlations. Specifically, multiple kernels and kernel canonical correlation analysis are combined to characterize the diverse inter-concept visual similarity relationships in a high-dimensional multi-modal feature space. MDS projection and Fisheye visualization technique are used to build the interactive user interface for concept network exploration. Our experimental results on large-scale image collections have observed very good results in terms of characterizing inter-concept correlations. For the next step, we will evaluate the advantage of the proposed concept network on multi-task multi-label image classification tasks.

Fourthly, due to the fact that most existing algorithms for image summarization lack either explicit formulation or quantitative evaluation metric, we determine to design novel framework for image collection summarization task. We had discovered that there is an intrinsic coherence between the problem of image collection summarization and the issue of dictionary learning for sparse representation, which both focus on selecting a small set of the most representative images to sparsely reinterpret the original image set in large size. We have explicitly reformulated the problem of automatic image summarization by using an sparse representation model and the simulated annealing algorithm is adopted to solve the optimization function more effectively. The reconstruction ability in terms of the MSE are used to objectively evaluate various algorithms for automatic image summarization. Our proposed algorithm outperformed 6 baseline algorithms both objectively and subjectively on 3 different image sets. As can be seen, the computation speed bottleneck of the system lies on addressing the optimization function. Due to the complexity of the proposed model, the current simulated annealing is far from producing reliable real-time solutions. For the next step, we will seek for better optimizers in terms of both reliability and computation speed.

Lastly, we have developed an efficient graffiti image retrieval system that uses the character detection results and integrates both image-level understanding and semantic-level understanding of the graffiti characters. The experiment result has shown the bounding box framework is both efficient and effective in the graffiti retrieval task, especially when compared with the traditional image retrieval framework. Our proposed system makes 4 primary contributions: a) Effective character extraction and noise elimination techniques to detect the character components in graffiti images; b) semantic-level bounding of meaningful character strings to facilitate semantic-level retrieval of graffiti images; c) fusing image-wise and semantic-wise scores for integral retrieval results; d) an interactive interface for graffiti exploration and retrieval.

The proposed system potentially can be used on, for example, mobile platforms that take photos as inputs and retrieve related information by connecting to a remote database. It could also be used for off-line tasks like large-scale graffiti image organization or classification. Future work includes developing a more efficient retrieval schema in order to extend the database to a much larger scale. We also want to apply more robust techniques to improve the semantic-level retrieval performance.

REFERENCES

[1] Aharon, M., Elad, M., and Bruckstein, A. K-svd: Design of dictionaries for sparse representation. *Proceedings of SPARS 5* (2005), 9–12.

[2] Aharon, M., Elad, M., and Bruckstein, A. The k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on 54*, 11 (2006), 4311–4322.

[3] B. Frey, D. D. Clustering by passing messages between data points. *Science*, 315 (2007), 972–977.

[4] Barnard, K., Duygulu, P., and Forsyth, D. Clustering art. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (2001), vol. 2, IEEE, pp. II–434.

[5] Barnard, K., and Forsyth, D. Learning the semantics of words and pictures. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001), vol. 2, IEEE, pp. 408–415.

[6] Battiato, S., Farinella, G., Gallo, G., and Ravì, D. Exploiting textons distributions on spatial hierarchy for scene classification. *Journal on Image and Video Processing 2010* (2010), 7.

[7] Baudisch, P., Good, N., Bellotti, V., and Schraedley, P. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves* (2002), ACM, pp. 259–266.

[8] Bay, H., Tuytelaars, T., and Van Gool, L. Surf: Speeded up robust features. *Computer Vision–ECCV 2006* (2006), 404–417.

[9] Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Transactions on PAMI 24*, 4 (Apr 2002), 509 –522.

[10] Bengio, Y., Bastien, F., Bergeron, A., Boulanger-Lewandowski, N., Breuel, T., Chherawala, Y., Cisse, M., Côté, M., Erhan, D., Eustache, J., et al. Deep learners benefit more from out-of-distribution examples. In *JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS* (2011).

[11] Benitez, A., Smith, J., and Chang, S. Medianet: A multimedia information network for knowledge representation. In *Proc. SPIE* (2000), vol. 4210, pp. 1–12.

[12] Cai, D., He, X., Li, Z., Ma, W., and Wen, J. Hierarchical clustering of www image search results using visual, textual and link information. In *Proceedings of the 12th annual ACM international conference on Multimedia* (2004), ACM, pp. 952–959.

[13] Caicedo, J. C., Moreno, J. G., Niño, E. A., and González, F. A. Combining visual features and text data for medical image retrieval using latent semantic kernels. In *Multimedia Information Retrieval* (2010), pp. 359–366.

[14] Camargo, J., and Gonzalez, F. Multimodal image collection summarization using non-negative matrix factorization. In *Computing Congress (CCC), 2011 6th Colombian* (2011), IEEE, pp. 1–6.

[15] Chen, J., Bouman, C., and Dalton, J. Hierarchical browsing and search of large image databases. *Image Processing, IEEE Transactions on 9*, 3 (2000), 442–455.

[16] Chen, Y., Wang, J., and Krovetz, R. Clue: Cluster-based retrieval of images by unsupervised learning. *Image Processing, IEEE Transactions on 14*, 8 (2005), 1187–1201.

[17] Chua, T., Tang, J., Hong, R., Li, H., Luo, Z., and Zheng, Y. Nus-wide: A real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval* (2009), ACM, p. 48.

[18] Chum, O., Philbin, J., and Zisserman, A. Near duplicate image detection: min-hash and tf-idf weighting. In *Proceedings of the British Machine Vision Conference* (2008), vol. 3, p. 4.

[19] Cilibrasi, R., and Vitanyi, P. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on 19*, 3 (2007), 370–383.

[20] Clocksin, W. Handwritten syriac character recognition using order structure invariance. In *Proceedings on Pattern Recognition* (Aug 2004), pp. 562 – 565 vol 2.

[21] Clough, P., Joho, H., and Sanderson, M. Automatically organising images using concept hierarchies. In *proceedings of the Multimedia Workshop running at ACM SIGIR conference* (2005).

[22] Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D., and Ng, A. Text detection and character recognition in scene images with unsupervised feature learning. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on* (2011), IEEE, pp. 440–445.

[23] Cong, Y., Yuan, J., and Luo, J. Towards scalable summarization of consumer videos ia sparse dictionary selection. *Multimedia, IEEE Transactions on*, 99 (2012), 1–1.

[24] Cox, T., and Cox, M. *Multidimensional scaling*, vol. 1. CRC Press, 2001.

[25] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV* (2004), vol. 1, p. 22.

[26] Deng, D. Content-based image collection summarization and comparison using self-organizing maps. *Pattern recognition 40*, 2 (2007), 718–727.

[27] Denton, T., Demirci, M., Abrahamson, J., Shokoufandeh, A., and Dickinson, S. Selecting canonical views for view-based 3-d object recognition. *ICPR* (2004).

[28] Ding, C., He, X., Zha, H., Gu, M., and Simon, H. A min-max cut algorithm for graph partitioning and data clustering. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on* (2001), IEEE, pp. 107–114.

[29] Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on* (2005), IEEE, pp. 65–72.

[30] Donoho, D. L., and Elad, M. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. In *Proc. Natl Acad. Sci. USA 100 2197-202* (2003).

[31] Duda, R., and Hart, P. *Pattern classification and scene analysis.* Wiley, 1996.

[32] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of statistics 32*, 2 (2004), 407–499.

[33] Engan, K., Aase, S., and Hakon Husoy, J. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. ICASSP'99. Proceedings., 1999 IEEE International Conference on* (1999), vol. 5, IEEE, pp. 2443–2446.

[34] Engan, K., Aase, S., and Husoy, J. Frame based signal compression using method of optimal directions (mod). In *Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on* (1999), vol. 4, IEEE, pp. 1–4.

[35] Fan, J., Gao, Y., and Luo, H. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *Proceedings of the 12th annual ACM international conference on Multimedia* (New York, NY, USA, 2004), MULTIMEDIA '04, ACM, pp. 540–547.

[36] Fan, J., Gao, Y., and Luo, H. Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation. *Image Processing, IEEE Transactions on 17*, 3 (2008), 407–426.

[37] Fan, J., Gao, Y., and Luo, H. Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation. *Image Processing, IEEE Transactions on 17*, 3 (2008), 407–426.

[38] Fan, J., Keim, D., Gao, Y., Luo, H., and Li, Z. Justclick: Personalized image recommendation via exploratory search from large-scale flickr images. *Circuits and Systems for Video Technology, IEEE Transactions on 19*, 2 (2009), 273–288.

[39] Fan, J., Yang, C., Shen, Y., Babaguchi, N., and Luo, H. Leveraging large-scale weakly-tagged images to train inter-related classifiers for multi-label annotation. In *Proceedings of the First ACM workshop on Large-scale multimedia retrieval and mining* (2009), ACM, pp. 27–34.

[40] Fei-Fei, L., and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 2, Ieee, pp. 524–531.

[41] Feng, S., Manmatha, R., and Lavrenko, V. Multiple bernoulli relevance models for image and video annotation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 2, IEEE, pp. II–1002.

[42] Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. Learning object categories from google's image search. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 2, IEEE, pp. 1816–1823.

[43] Fergus, R., Perona, P., and Zisserman, A. A visual category filter for google images. *Computer Vision-ECCV 2004* (2004), 242–256.

[44] Fischler, M., and Bolles, R. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395.

[45] Foo, J., and Sinha, R. Pruning sift for scalable near-duplicate image matching. In *Proceedings of the eighteenth conference on Australasian database-Volume 63* (2007), Australian Computer Society, Inc., pp. 63–71.

[46] Furnas, G. *Generalized fisheye views*, vol. 17. ACM, 1986.

[47] Gao, B., Liu, T., Qin, T., Zheng, X., Cheng, Q., and Ma, W. Web image clustering by consistent utilization of visual features and surrounding texts. In *Proceedings of the 13th annual ACM international conference on Multimedia* (2005), ACM, pp. 112–121.

[48] Gao, Y., Fan, J., Luo, H., and Satoh, S. A novel approach for filtering junk images from google search results. *Advances in Multimedia Modeling* (2008), 1–12.

[49] Gersho, A., and Gray, R. *Vector quantization and signal compression*, vol. 159. Springer, 1992.

[50] Ghamrawi, N., and McCallum, A. Collective multi-label classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (2005), ACM, pp. 195–200.

[51] Goemans, M., and Williamson, D. . 879-approximation algorithms for max cut and max 2sat. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing* (1994), ACM, pp. 422–431.

[52] Goldberger, J., and Tassa, T. The hungarian clustering method.

[53] Graham, A., Garcia-Molina, H., Paepcke, A., and Winograd, T. Time as essence for photo browsing through personal digital libraries. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries* (2002), ACM, pp. 326–335.

[54] Grauman, K., and Darrell, T. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 2, Ieee, pp. 1458–1465.

[55] Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset.

[56] Gunther, N., and Beretta, G. A benchmark for image retrieval using distributed systems over the internet: Birds-i. *Arxiv preprint cs/0012021* (2000).

[57] Guttman, A. *R-trees: a dynamic index structure for spatial searching*, vol. 14. ACM, 1984.

[58] Hardoon, D., Szedmak, S., and Shawe-Taylor, J. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation 16*, 12 (2004), 2639–2664.

[59] Hauptmann, A., Yan, R., Lin, W., Christel, M., and Wactlar, H. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *Multimedia, IEEE Transactions on 9*, 5 (2007), 958–966.

[60] He, X., King, O., Ma, W., Li, M., and Zhang, H. Learning a semantic space from user's relevance feedback for image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on 13*, 1 (2003), 39–48.

[61] He, X., Ma, W., and Zhang, H. Imagerank: spectral techniques for structural analysis of image database. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on* (2003), vol. 1, Ieee, pp. I–25.

[62] Hoyer, P. Non-negatie sprase coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on* (2002), IEEE, pp. 557–565.

[63] http://www.flickr.com.

[64] http://www.picasa.com.

[65] Huang, J., Kumar, S., and Zabih, R. An automatic hierarchical image classification scheme. In *Proceedings of the sixth ACM international conference on Multimedia* (1998), ACM, pp. 219–228.

[66] Indyk, P. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on* (2000), IEEE, pp. 189–197.

[67] Indyk, P., and Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (1998), ACM, pp. 604–613.

[68] Jaffe, A., Naaman, M., Tassa, T., and Davis, M. Generating summaries and visualization for large collections of geo-referenced photographs. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval* (2006), ACM, pp. 89–98.

[69] Jaimes, A., Chang, S., and Loui, A. Detection of non-identical duplicate consumer photographs. In *Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on* (2003), vol. 1, Ieee, pp. 16–20.

[70] Jain, A. K., eun Lee, J., and Jin, R. Graffiti-id: Matching and retrieval of graffiti images. *Proceeding on MiFor* (2009).

[71] Jain, P., Kulis, B., and Grauman, K. Fast image search for learned metrics. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), Ieee, pp. 1–8.

[72] Jiang, J., and Conrath, D. Semantic similarity based on corpus statistics and lexical taxonomy. *Arxiv preprint cmp-lg/9709008* (1997).

[73] Jiang, Y., Ngo, C., and Yang, J. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval* (2007), ACM, pp. 494–501.

[74] Jing, Y., and Baluja, S. Pagerank for product image search. In *Proceeding of the 17th international conference on World Wide Web* (2008), ACM, pp. 307–316.

[75] Jing, Y., Baluja, S., and Rowley, H. Canonical image selection from the web. In *Proceedings of the 6th ACM international Conference on Image and Video Retrieval* (2007), ACM, pp. 280–287.

[76] Jing, Y., Wang, H., and Covell, D. Google image swirl.

[77] Johnson, D., and Garey, M. Computers and intractability: A guide to the theory of np-completeness. *Freeman&Co, San Francisco* (1979).

[78] Kankanhalli, M., and Rui, Y. Application potential of multimedia information retrieval. *Proceedings of the IEEE 96*, 4 (april 2008), 712 –720.

[79] Ke, Y., Sukthankar, R., and Huston, L. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia* (2004), vol. 4, p. 5.

[80] Kim, S., Park, S., and Kim, M. Central object extraction for object-based image retrieval. CIVR'03, Springer-Verlag, pp. 39–49.

[81] Krause, A., and Cevher, V. Submodular dictionary selection for sparse representation. In *Proc. ICML* (2010).

[82] Kruskal, J. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27.

[83] Kuhn, H. The hungarian method for the assignment problem. *Naval research logistics quarterly 2*, 1-2 (1955), 83–97.

[84] Lam, L., Lee, S.-W., and Suen, C. Thinning methodologies-a comprehensive survey. *IEEE Transactions on PAMI 14*, 9 (sep 1992), 869 –885.

[85] Lamping, J., and Rao, R. The hyperbolic browser: A focus+ context technique for visualizing large hierarchies. *Card, Stuart K., Mackinley, Jock D., Shneiderman: Readings in Information Visualisation Using Vision to Think* (1999), 381–408.

[86] Lamping, J., Rao, R., and Pirolli, P. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1995), ACM Press/Addison-Wesley Publishing Co., pp. 401–408.

[87] Lazebnik, S., Schmid, C., and Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 2, Ieee, pp. 2169–2178.

[88] Li, J., and Wang, J. Automatic linguistic indexing of pictures by a statistical modeling approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 25*, 9 (2003), 1075–1088.

[89] Li, L., and Fei-Fei, L. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), IEEE, pp. 1–8.

[90] Li, Y., Shapiro, L., and Bilmes, J. A generative/discriminative learning algorithm for image classification. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 2, IEEE, pp. 1605–1612.

[91] Ling, H., and Soatto, S. Proximity distribution kernels for geometric context in category recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), IEEE, pp. 1–8.

[92] Lipson, P., Grimson, E., and Sinha, P. Configuration based scene classification and image indexing. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on* (1997), IEEE, pp. 1007–1013.

[93] Liu, Y., Zhang, D., Lu, G., and Ma, W. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition 40*, 1 (2007), 262–282.

[94] Loeff, N., Alm, C., and Forsyth, D. Discriminating image senses by clustering with multimodal features. In *Proceedings of the COLING/ACL on Main conference poster sessions* (2006), Association for Computational Linguistics, pp. 547–554.

[95] Lowe, D. Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2 (2004), 91–110.

[96] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *IJCV, 60, pp. 91-110* (2003).

[97] Luo, J., Savakis, A., and Singhal, A. A bayesian network-based framework for semantic image understanding. *Pattern Recognition 38*, 6 (2005), 919–934.

[98] Ma, W., and Manjunath, B. Texture features and learning similarity. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on* (1996), IEEE, pp. 425–430.

[99] MacCuish, J., McPherson, A., Barros, J., and Kelly, P. Interactive layout mechanisms for image database retrieval. In *Proceedings of SPIE* (1996), vol. 2656, p. 104.

[100] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, pp. 689–696.

[101] Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. Discriminative learned dictionaries for local image analysis. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[102] Mallat, S., and Zhang, Z. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on 41*, 12 (1993), 3397–3415.

[103] Mehta, B., Nangia, S., Gupta, M., and Nejdl, W. Detecting image spam using visual features and near duplicate detection. In *Proceedings of the 17th international conference on World Wide Web* (2008), ACM, pp. 497–506.

[104] Meng, Y., Chang, E., and Li, B. Enhancing dpf for near-replica image recognition. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 2, IEEE, pp. II–416.

[105] Moon, H., and Phillips, P. Computational and performance aspects of pca-based face recognition algorithms. *Perception, Vol 30* (2001), 302–321.

[106] Nadira, M., Nik Kamariah, N. I., Jasni, M. Z., and Siti Azami, A. B. Optical character recognition by using template matching (alphabet). In *National Conference on Software Engineering and Computer Systems* (2007).

[107] Naphade, M., Smith, J., Tesic, J., Chang, S., Hsu, W., Kennedy, L., Hauptmann, A., and Curtis, J. Large-scale concept ontology for multimedia. *Multimedia, IEEE 13*, 3 (2006), 86–91.

[108] Naphide, H., and Huang, T. A probabilistic framework for semantic video indexing, filtering, and retrieval. *Multimedia, IEEE Transactions on 3*, 1 (2001), 141–151.

[109] Nhung, N., and Phuong, T. An efficient method for filtering image-based spam. In *Research, Innovation and Vision for the Future, 2007 IEEE International Conference on* (2007), IEEE, pp. 96–102.

[110] Nister, D., and Stewenius, H. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 2, Ieee, pp. 2161–2168.

[111] Oliva, A., and Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision 42*, 3 (2001), 145–175.

[112] Oliva, A., and Torralba, A. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research 155* (2006), 23–36.

[113] Pandey, A., Sawant, S., Eric, D., and Schwartz, M. Handwritten character recognition using template matching, 2010.

[114] Pei, G., Gerla, M., and Chen, T. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on* (2000), vol. 1, Ieee, pp. 70–74.

[115] Pervouchine, V., and Leedham, G. Document examiner feature extraction: Thinned vs. skeletonised handwriting images. In *TENCON IEEE Region 10* (Nov 2005), pp. 1–6.

[116] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), Ieee, pp. 1–8.

[117] Qiu, G. Image coding using a coloured pattern appearance model. In *Visual Communication and Image Processing* (2001).

[118] Quelhas, P., Monay, F., Odobez, J., Gatica-Perez, D., Tuytelaars, T., and Van Gool, L. Modeling scenes with local descriptors and latent aspects. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 1, Ieee, pp. 883–890.

[119] Rege, M., Dong, M., and Hua, J. Graph theoretical framework for simultaneously integrating visual and textual features for efficient web image clustering. In *Proceeding of the 17th international conference on World Wide Web* (2008), ACM, pp. 317–326.

[120] Renninger, L., and Malik, J. When is scene identification just texture recognition? *Vision research 44*, 19 (2004), 2301–2311.

[121] Rubner, Y., Guibas, L., and Tomasi, C. The earth movers distance, multi-dimensional scaling, and color-based image retrieval. In *Proceedings of the ARPA Image Understanding Workshop* (1997), pp. 661–668.

[122] Rui, Y., Huang, T., and Chang, S. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation 10*, 1 (1999), 39–62.

[123] Rui, Y., Huang, T., Ortega, M., and Mehrotra, S. Relevance feedback: A power tool for interactive content-based image retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on 8*, 5 (1998), 644–655.

[124] Russell, B., Torralba, A., Murphy, K., and Freeman, W. Labelme: a database and web-based tool for image annotation. *International journal of computer vision 77*, 1 (2008), 157–173.

[125] Sapiro, G., and Castrodad, A. Sparse modeling of human actions from motion imagery, 2011.

[126] Schmitz, P. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland* (2006), pp. 210–214.

[127] Sebe, N., Lew, M., and Huijsmans, D. Multi-scale sub-image search. In *Proceedings of the seventh ACM international conference on Multimedia (Part 2)* (1999), ACM, pp. 79–82.

[128] Sethi, I., Coman, I., Day, B., Jiang, F., Li, D., Segovia-Juarez, J., Wei, G., and You, B. Color-wise: A system for image similarity retrieval using color. In *Proceedings of SPIE* (1997), vol. 3312, p. 140.

[129] Seung, D., and Lee, L. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems 13* (2001), 556–562.

[130] Shen, Y., and Fan, J. Leveraging loosely-tagged images and inter-object correlations for tag recommendation. In *Proceedings of the international conference on Multimedia* (2010), ACM, pp. 5–14.

[131] Shi, J., and Malik, J. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 8 (2000), 888–905.

[132] Shroff, N., Turaga, P., and Chellappa, R. Video précis: Highlighting diverse aspects of videos. *Multimedia, IEEE Transactions on 12*, 8 (2010), 853–868.

[133] Simon, I., Snavely, N., and Seitz, S. Scene summarization for online image collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), Ieee, pp. 1–8.

[134] Sinha, P. Summarization of archived and shared personal photo collections. In *Proceedings of the 20th international conference companion on World wide web* (2011), ACM, pp. 421–426.

[135] Sivic, J., and Zisserman, A. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (2003), Ieee, pp. 1470–1477.

[136] Smeulders, A., Worring, M., Santini, S., Gupta, A., and Jain, R. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 12 (2000), 1349–1380.

[137] Stan, D., and Sethi, I. eid: a system for exploration of image databases. *Information processing & management 39*, 3 (2003), 335–361.

[138] Stark, M., and Riesenfeld, R. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering* (1998), Citeseer.

[139] Sze, K., Lam, K., and Qiu, G. Scene cut detection using the colored pattern appearance model. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on* (2003), vol. 2, IEEE, pp. II–1017.

[140] Tang, F., and Gao, Y. Fast near duplicate detection for personal image collections. In *Proceedings of the 17th ACM international conference on Multimedia* (2009), ACM, pp. 701–704.

[141] Tao, D., Tang, X., Li, X., and Rui, Y. Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. *Multimedia, IEEE Transactions on 8*, 4 (2006), 716–727.

[142] Tao, D., Tang, X., Li, X., and Wu, X. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 7 (2006), 1088–1099.

[143] Thomee, B., Huiskes, M., Bakker, E., and Lew, M. Large scale image copy detection evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval* (2008), ACM, pp. 59–66.

[144] Tong, S., and Chang, E. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia* (2001), ACM, pp. 107–118.

[145] Torralba, A., Fergus, R., and Weiss, Y. Small codes and large image databases for recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[146] Torralba, A., Murphy, K., and Freeman, W. Sharing features: efficient boosting procedures for multiclass object detection. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 2, Ieee, pp. II–762.

[147] Torralba, A., and Oliva, A. Semantic organization of scenes using discriminant structural templates. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (1999), vol. 2, IEEE, pp. 1253–1258.

[148] Trahanias, P. E., Stathatos, K., Stamatelopoulos, F., and Skordalakis, E. Morphological hand-printed character recognition by a skeleton-matching algorithm. *J. Electron. Imaging 2, 114* (1993).

[149] van de Sande, K., Gevers, T., and Snoek, C. A comparison of color features for visual concept classification. In *Proceedings of the 2008 international conference on Content-based image and video retrieval* (2008), ACM, pp. 141–150.

[150] Vasconcelos, N. Image indexing with mixture hierarchies.

[151] Veltkamp, R., and Tanase, M. Content-based image retrieval systems: A survey.

[152] Wang, B., Li, Z., Li, M., and Ma, W. Large-scale duplicate detection for web image search. In *Multimedia and Expo, 2006 IEEE International Conference on* (2006), Ieee, pp. 353–356.

[153] Wang, J., Jia, L., and Hua, X. Interactive browsing via diversified visual summarization for image search results. *Multimedia Systems 17*, 5 (2011), 379–391.

[154] Wang, J., Li, J., and Wiederhold, G. Simplicity: Semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 9 (2001), 947–963.

[155] Wang, X.-J., Ma, W.-Y., Xue, G.-R., and Li, X. Multi-model similarity propagation and its application for web image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia* (New York, NY, USA, 2004), MULTIMEDIA '04, ACM, pp. 944–951.

[156] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on 13*, 4 (2004), 600–612.

[157] Wang, Z., Josephson, W., Lv, Q., Charikar, M., and Li, K. Filtering image spam with near-duplicate detection. In *Proceedings of CEAS* (2007).

[158] Weston, J., Bengio, S., and Usunier, N. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning 81*, 1 (2010), 21–35.

[159] White, D., and Jain, R. Similarity indexing: Algorithms and performance. In *Storage and Retrieval for Image and Video Databases (SPIE)* (1996), pp. 62–73.

[160] Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems 2*, 1 (1987), 37–52.

[161] Wright, J., Yang, A., Ganesh, A., Sastry, S., and Ma, Y. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31*, 2 (2009), 210–227.

[162] Wu, L., Hua, X., Yu, N., Ma, W., and Li, S. Flickr distance. In *Proceedings of the 16th ACM international conference on Multimedia* (2008), ACM, pp. 31–40.

[163] Wu, L., Li, M., Li, Z., Ma, W., and Yu, N. Visual language modeling for image classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval* (2007), ACM, pp. 115–124.

[164] Wu, X., Ngo, C., Hauptmann, A., and Tan, H. Real-time near-duplicate elimination for web video search with content and context. *Multimedia, IEEE Transactions on 11*, 2 (2009), 196–207.

[165] Wu, Z., Ke, Q., Isard, M., and Sun, J. Bundling features for large scale partial-duplicate web image search. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 25–32.

[166] Xu, D., Cham, T., Yan, S., and Chang, S. Near duplicate image identification with patially aligned pyramid matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–7.

[167] Xu, D., and Chang, S. Visual event recognition in news video using kernel methods with multi-level temporal alignment. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8.

[168] Y. Hadi, F. Essannouni, R. T. Video summarization by k-medoid clustering. *SAC* (2006).

[169] Yang, C., Feng, X., Peng, J., and Fan, J. Efficient large-scale image data set exploration: visual concept network and image summarization. *Advances in Multimedia Modeling* (2011), 111–121.

[170] Yang, C., Shen, J., and Fan, J. Effective summarization of large-scale web images. In *Proceedings of the 19th ACM international conference on Multimedia* (2011), ACM, pp. 1145–1148.

[171] Yang, X., Zhu, Q., and Cheng, K. Near-duplicate detection for images and videos. In *Proceedings of the First ACM workshop on Large-scale multimedia retrieval and mining* (2009), ACM, pp. 73–80.

[172] Yeh, M.-C., and Cheng, K.-T. A string matching approach for visual retrieval and classification. In *Multimedia Information Retrieval* (2008), pp. 52–58.

[173] Zhang, D., and Chang, S. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *Proceedings of the 12th annual ACM international conference on Multimedia* (2004), ACM, pp. 877–884.

[174] Zhang, H., and Zhong, D. A scheme for visual feature based image indexing. *Readings in multimedia computing and networking* (2002), 278.

[175] Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. Local features and kernels for classification of texture and object categories: A comprehensive study. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on* (2006), Ieee, pp. 13–13.

[176] Zhang, R., Zhang, Z., Li, M., Ma, W., and Zhang, H. A probabilistic semantic model for image annotation and multi-modal image retrieval. *Multimedia Systems 12*, 1 (2006), 27–33.

[177] Zhou, X., and Huang, T. Small sample learning during multimedia retrieval using biasmap. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (2001), vol. 1, IEEE, pp. I–11.

[178] Zhu, J., Hoi, S., Lyu, M., and Yan, S. Near-duplicate keyframe retrieval by nonrigid image matching. In *Proceedings of the 16th ACM international conference on Multimedia* (2008), ACM, pp. 41–50.

[179] Zhu, K., Qi, F., Jiang, R., Xu, L., Kimachi, M., and Wu, Y. Using adaboost to detect and segment characters from natural scenes. *In Proc. International Workshop on CBDAR* (2005), 52 – 59.