Chapter XIV
Annotating Images by Mining Image Search

Xin-Jing Wang
Microsoft Research Asia, China

Lei Zhang
Microsoft Research Asia, China

Xirong Li
Microsoft Research Asia, China

Wei-Ying Ma
Microsoft Research Asia, China

ABSTRACT

Although it has been studied for years by computer vision and machine learning communities, image annotation is still far from practical. In this chapter, the authors propose a novel attempt of modeless image annotation, which investigates how effective a data-driven approach can be, and suggest annotating an uncaptioned image by mining its search results. The authors collected 2.4 million images with their surrounding texts from a few photo forum Web sites as our database to support this data-driven approach. The entire process contains three steps: (1) the search process to discover visually and semantically similar search results; (2) the mining process to discover salient terms from textual descriptions of the search results; and (3) the annotation rejection process to filter noisy terms yielded by step 2. To ensure real-time annotation, two key techniques are leveraged – one is to map the high dimensional image visual features into hash codes, the other is to implement it as a distributed system, of which the search and mining processes are provided as Web services. As a typical result, the entire process finishes in less than 1 second. Since no training dataset is required, our proposed approach enables annotating with unlimited vocabulary, and is highly scalable and robust to outliers. Experimental results on real Web images show the effectiveness and efficiency of the proposed algorithm.
INTRODUCTION

The number of digital images has exploded with the advent of digital cameras which requires effective image search techniques. Although it is an intuitive way of image search since “a picture worths a thousand word”, currently the Query-By-Example (QBE) scheme (i.e. using images as queries) is seldom adopted by commercial image search engines. The reasons are at least twofold: 1) the semantic gap problem. It is also the fundamental problem in Content-based Image Retrieval (CBIR) field. The techniques to extract low-level features, such as color, texture, and shape etc., are far from powerful to represent the semantics contained in an image, and hence given an image, its search results may be quite different conceptually to the query, although they possess same chromatic or textural appearances. 2) Computational expensiveness. It is well known that the inverted indexing technique ensures the practical and successful usage of current text search engines. This technique uses keywords as entries to index documents containing them, which also aligns seamlessly with the Query-by-Keyword (QBK) scheme adopted by text search engines. Thus given a number of query keywords, the search result can just be the intersection of documents indexed by these keywords separately (if no ranking functions applied). However, in the image search case, since images are 2D media, and the spatial relationship between pixels is critical in conveying the semantics of an image, how to define image “keyword” is still an open question. This prevents the inverted indexing technique from being directly used in image search, which creates a critical efficiency problem of using image visual features to search.

Due to the above reasons, there is a surge of interests on image auto-annotation and object recognition in recent years. Researchers try to define ways to automatically assign keywords onto images or image regions. While all the previous work build learning models to annotation images, in this chapter, we attempt to investigate how effective a modeless and data-driven approach, which leverages the huge amount of commented image data in the Web, could be.

This is reasonable because (1) to manually label images is a very expensive task (Naphade, Smith, Tesic, Chang, Hsu, Kennedy, Hauptmann, & Curtis, 2006), and in many cases, different people tend to have different explanations on a single image. Such inconsistency even makes image labeling a research topic that many questions should be addressed: What kind of images can be labeled consistently? What is the strategy of labeling images that can ensure the consistency, etc? All these obstacles lead to the lack of training data, which discourages researchers from learning a practically effective annotation model even if we have such a huge image data set in the Web. (2) Some researchers proposed very efficient and effective image encoding approaches (Fan, Xie, Li, Li, & Ma, 2005) which convert an image into an N-bit hash code so that image visual distance can be measured simply by the Hamming distance between the hash codes. This enables large-scale content-based image search in real-time and set a light on combining image visual appearance into the commercial search engines that are purely based on text search since both image indexing (e.g. matching the first n bits of two hash codes) and retrieval are of O(1) complexity.

Imagine that an image set that is large enough is available, so that for each query image at least one duplication can be found in the image set, and then what we need to do is just to annotate the query by the comments or surrounding texts of the duplication. As analyzed above, this is applicable in real-time since each image is converted into a hash code so that large-scale image indexing is enabled.

This motivated us to leverage Web data and crawl Web images for a data-driven image annotation approach since the Web not only deposits huge amount of images but also they generally have human assigned comments or descriptions as their surrounding texts. Apparently the ideal case above is still far from what our current technique can achieve; however, we can retrieve a group of very similar im-
Annotating Images by Mining Image Search

ages to the query, or the near-duplicate, instead of the duplications. These search results, intuitively, will cover the major semantics inside the query image. On the other hand, the textual descriptions of these near duplications are most often noisy and diverse which are far from optimal as annotations, so that intuitively, data mining techniques can successively be applied onto the search results to learn the common keywords or phrases in a majority voting sense, which can thus be assigned to the query image as the learned annotations. This is the key idea of our proposed modeless and data-driven approach in this paper. It not only investigates how much data can help us in recognizing an image as the saying “data is the king”, but also demonstrates a way to support practical image annotation which can handle large-scale image data in real-time.

The paper is organized as follows. Section II gives a survey on the previous works on image annotation, which provides the reader a general picture on current art. Section III describes our insights of the image annotation problem, which motivate us the proposed modeless and data-driven approach as a novel and practical solution compared with the previous model-based ones. Section IV presents our modeless approach in detail and Section V provides experimental results. Several discussions are inspired in Section VI and the paper is concluded in Section VII with a few outlooks of possible improvements.

RELATED WORKS

In the early works on image annotation, many researchers sought to users’ relevance feedback to assign labels to a given image. For example, Liu, Dumais, Sun, Zhang, Czerwinski, & Field (2001) asked the user to label a given image in the relevance feedback stage and then propagate these labels to all the positive images suggested by the retrieval system. Shevade and Sundaram (2006) made a further step. They attempted to propagate the image annotations by calculating the propagation likelihood based on WordNet (Fellbaum, 1998) synonym sets as well as image low-level features, and present only those images that are most ambiguous to the user for relevance feedback.

Due to the explosion of digital images and the semantic gap problem, image auto-annotation has become a hot research topic in recent years. And most of the researchers work on two major research directions. One attempt is to learn the joint probabilities between images and words. Most of this type of research tries to learn a generative model leveraging a training dataset and then infer the labels for the new-coming images according to the learnt correlations. For example, Blei et al. extended the Latent Dirichlet Allocation (LDA) model to the mix of words and images and proposed a so-called Correlation LDA model (Blei and Jordan, 2003). This model assumes that there is a hidden layer of topics, which are a set of latent factors and obey the Dirichlet distribution. And words and regions are conditionally independent given the topics. This work used 7,000 Corel photos and a vocabulary of 168 words for annotation.

Barnard, Duygulu, Freitas, Forsyth, Blei, & Jordan (2003) proposed a generative model and tested various statistical models to learn the joint probabilities of image regions and words. They used 16,000 Corel photos with 155 words and automatically annotated 10,000 test images.

Li and Wang (2003) proposed a two-dimensional multi-resolution hidden Markov model to connect images and concepts. They used 60,000 Corel photos with 600 concepts. In one of their recent work, they improved this model and propose a real-time annotation approach named Alipr which attracts attention from both academic and industry. Lavrenko, Manmatha, & Jeon (2003) proposed a continuous relevance model, a generative model which directly associates continuous features with words and achieved significant improvement in performance. Jeon, Lavrenko, & Manmatha (2003) and Jeon and
Manmatha (2004) extended this model to learn the joint probabilities between a set of words and image features. The training set they used were about 56,000 Yahoo! news images with noisy annotations and a vocabulary of 4,073 words. This is the largest vocabulary used in the previous work, and they discussed noisy-annotation filtering and speeding-up schemes.

The generative model proposed by Zhang, Zhang, Li, Ma, & Zhang (2005) added a hidden layer of semantic concepts to connect visual features and words. The concepts are discovered to explicitly exploit the synergy among the modalities.

All the approaches mentioned above suggest generative models. As a different approach, Pan, Yang, Faloutsos, & Duygulu (2004a) formulated this problem as learning a graph structure. They constructed a two-layer graph on training images and their associated captions, and used the random-walk-with-restarts algorithm to estimate the correlations between new images and the existed captions. In another work of Pan, et al. (2004b), they further extended the model into a three-layered graph by taking into consideration the image regions. Also, Monay and Gatica-Perez (2003, 2004) tested the performance of latent space models on this task.

Duygulu, Barnard, Freitas, & Forsyth (2002) and Barnard, Duygulu, Freitas, & Forsyth (2003) represented images by a group of blobs, and then used a statistical machine translation model to translate the blobs into a set of keywords.

Another group of researchers attempted to learn classifiers based on the training data. Chang, King-shy, Sychay, & Wu (2003) started their process by manually labeling one caption for each image in a small training set, then train an ensemble of binary classifiers, each classifier for a specific label. Li, Goh, & Chang (2003) proposed a Confidence-based Dynamic Ensemble model which was a two-stage classification method. Carneiro and Vasconcelos (2005) attempted to establish a one-to-one mapping between semantic classes and the groups of database images that share semantic labels, and the criterion is to minimize the probability of errors. Mori, Takahashi, & Oka (1999) uniformly divided each image into sub-images with key words, and then used vector quantization of the feature vectors of sub-images to estimate which words should be assigned to a new image.

The work above generally requires a training stage to learn prediction models, thus the generalization capability should be a crucial assessment for practical effectiveness of the proposed approaches.

Recently, some researchers began to leverage Web-scale data for image understanding (Duygulu, et al., 2002, Yeh, Tollmar, & Darrell, 2004). An interesting work was proposed by Yeh, Tollmar, & Darrell (2004) which identifies locations by searching the Internet. Given a picture of an unknown place, it firstly obtains a small number of visually relevant Web images using content-based search, then extracts a few keywords from the descriptions of these images. A text-based search is successively performed and the search results are further filtered by visual features.

The disadvantages of (Yeh, et al. 2004) are that due to the efficiency problem, only a small number of relevant images can be retrieved as seeds which possibly degrades the performance, and the semantic gap problem will inevitably bias the final results. However, ignoring all these vulnerabilities, it is still an important work which pioneers a different point of view to investigate the image annotation problem.

THEORY BEHIND – THE MOTIVATION

This section provides our insights into the image auto-annotation problem, which directly inspired the idea of our modeless and data-driven image annotation approach.
Annotating Images by Mining Image Search

Fundamentally, the aim of image auto-annotation is to find a group of keywords $w^*$ that maximizes the conditional distributions $p(w \mid I_q)$, as shown the first row in Eq.1, where $I_q$ is the un-captioned query image and $w$ are keywords or phrases in the vocabulary. Applying the Bayes rule, we obtain Eq.1(a), where $I_i$ denotes the $i^{th}$ image in the database. This corresponds to the generative model shown in Fig. 1(a), in which annotations are generated directly from the images. If we assume that there is a hidden layer of “topics”, so that images are distributed w.r.t. a group of topics, and it is from these topics that words are generated, then we obtain a topic model as shown in Fig. 1(b), which corresponds to the function shown in Eq.1(b), where $t_j$ represents the $j^{th}$ topic in the topic space.

$$
\begin{align}
    w^* &= \arg \max_w p(w \mid I_q) \\
    &= \arg \max_w \sum_i p(w \mid I_i) \cdot p(I_i \mid I_q) \quad \cdots (a) \\
    &= \arg \max_w \sum_i \left( \sum_j p(w \mid t_j) \cdot p(t_j \mid I_i) \right) \cdot p(I_i \mid I_q) \quad \cdots (b)
\end{align}
$$

Most of the previous generative approaches can be categorized into these two formulations. Moreover, since the model of Eq.1(b) investigates the relationships between images and words in a more exhaustive way, it is generally reported as more effective in many previous works (Barnard, et al., 2003, Blei and Jordan, 2003, Monay and Gatica-Perez, 2004, Zhang, et al. 2005).

However, in this paper we interpret Eq.1 in a different way. More specifically, we view Eq.1 from the angle of search and data mining.

Recall that the goal of a typical search process is to retrieve a set of images $\Gamma$ given a query $I_q$ such that for each $I_i \in \Gamma$, we have

$$
I_q = \arg \max_{i \in \Gamma} p(I_i \mid I_q)
$$

Moreover, contrary to traditional approaches which learn $p(w \mid I)$ by defining either a generative model or discriminative model, we can just mine it out given a set of images, and such a mining process attracts interests recently, which is typically called label-based clustering techniques (Toda and Kataoka 2005, Zeng, He, Chen, & Ma, 2004).

Figure 1. Generative models for image auto-annotation
Different from the traditional document-based clustering approaches, e.g. k-means, which cluster documents based on the similarity of features and yield non-overlapped clusters, label-based approaches are typically applied to search results, which attempt to generate indicative and detailed descriptions to categorize the retrieved documents into clusters as well as to facilitate users’ browsing, and generally yield overlapped clusters. The art is in learning the informative and classifiable words or phrases which summarize and categorize a subset of the documents. For example, Zeng, et al. (2004) defined five features to represent each n-gram and then learnt a linear regression model to rank the n-grams; the top-ranked n-grams were used as the corresponding clusters’ names to differentiate the documents into clusters.

Note that a pre-requisition of label-based clustering approaches is that such techniques should be applied to search results. Since all the documents are retrieved given a single query, or say they are relevant in some sense, it reduces the noise and diversity. Hence the most important feature of the “cluster names”, or the learnt indicative words or phrases, is to provide further details of a cluster of documents, and hence such techniques are typically used for improving the display of a search engine (Vivisimo, 2007).

This pre-requisition is automatically satisfied in our case, as stated in Section I. Instead of finding duplications for a query image (as is an excessive demand), we find each query image a group of near-duplications, or relevant images both visually and semantically (i.e. have similar textual descriptions), which are just the “search results” required by a label-based clustering technique, and hence we get to the solution of $p(w | I)$.

Based on the analysis above, we propose a novel solution of image annotation by first retrieving a subset of similar images $I$ to the query (i.e. $p(I | I_q)$), and then mining annotations from this image subset $I$ leveraging a label-based clustering technique (i.e. $p(w | I)$), which solves Eq.1(a).

THE MODELESS ANNOTATION APPROACH OF MINING SEARCH RESULTS

We have illustrated that the idea of annotating an image by those informative and distinctive phrases mined from its search results is reasonable. Here in this section we not only provide the detailed solution, but also show our ways of ensuring its effectiveness and efficiency simultaneously.

The Reformulation

Before the illustration, let us reformulate Eq.1 in a more “image” way. Let $\Theta = \bigcup I_q$ denotes the set of images relevant images $I_q$, so that $p(\Theta | I_q)$ simulates the search process, and $p(w | \Theta_q)$ simulates the mining process. The label-based clustering approach is then to discover topics such that

$$p(w | \Theta_q) = \arg \max_t p(w | t) \cdot p(t | \Theta_q)$$

where $t$ is approximated by the “cluster names” given by the clustering approach.
Hence we have
\[
\begin{align*}
\mathbf{w}^* &= \arg\max_{\mathbf{w}} p(\mathbf{w} | I_q) \\
&= \arg\max_{\mathbf{w}} \sum_q p(\mathbf{w} | \Theta_q) \cdot p(\Theta_q | I_q) \quad \cdots (a) \\
&= \arg\max_{\mathbf{w}} \left[ \arg\max_t p(\mathbf{w} | \mathbf{t}) \cdot p(\mathbf{t} | \Theta_q) \right] \cdot p(\Theta_q | I_q) \quad \cdots (b)
\end{align*}
\]

From Eq. 4, we can see that there are three critical factors that affect the effectiveness of the proposed approach, namely the retrieval process \(p(\Theta_q | I_q)\), the mining process \(p(\mathbf{t} | \Theta_q)\), and the filtering process \(p(\mathbf{w} | \mathbf{t})\).

We do not want to propose a label-based clustering technique here in this paper, but adopt an existing and proved-effective approach called Search Result Clustering (SRC) proposed by Zeng, et al. (2004) which is a released product by Microsoft Research Asia (SRC, 2006).

As for the filtering process, we propose a simple but effective relevance ranking and annotation rejection approach to select and rank the mined cluster names, and use the top ones as the output annotations. Intuitively, the preciseness of the identified annotations depends on the effectiveness of the SRC technique. However, since SRC is purely based on textual descriptions rather than taking image visual features into consideration as well, the effectiveness of SRC is thus degraded in our scenario. We will leave it as a future work and concentrate on the entire solution of a search and mining based image auto-annotation approach in this paper.

Figure 2. The flowchart of the modeless image annotation process of mining annotations from image search results. It contains three steps: the search stage (labeled by “(1)” in the figure), the mining stage (“(2)”) and the filtering stage (“(3)”). The uncaptioned image and an initial keyword “sunset” assigned serve as the queries. The highlighted yellow block shows the output annotations.
Annotating Images by Mining Image Search

The retrieval process is a bit tricky in our current implementation, but we believe that it helps move a step forward in image understanding. It is well known that the semantic gap problem dooms the Content-based Image Retrieval (CBIR) approaches, which is thus called a fundamental problem in CBIR fields. It is caused by the gap between human semantics and the low-level visual features insufficiently effective given current feature extraction skills (Huang, Kumar, Mitra, Zhu, & Zabih, 1997). On the other hand, many researchers have proved that although individually the visual features and textual descriptions of images are ambiguous, e.g. both images labeled as “tiger lily” and “white tiger” are relevant to query keyword “tiger”, they tend not to be when combined together (Barnard, Duygulu, Freitas, & Forsyth, 2001, Blei and Jordan, 2003, Li and Wang, 2003, Li and Wang, 2006).

Hence we suggest to divide-and-conquer the semantic gap problem. Suppose two steps: (1) given a query image, find one accurate keyword (or phrase); and (2) given the image and the keyword, find more complementary keywords or phrases that detail the content of the image. The second step is easy to comprehend. By providing an initial keyword, we reduce the ambiguity of the image query and promise the search results to be relevant both visually and semantically at least to some extent. On the other hand, the requirement in the first step is not as lacking in subtlety as it may first seem. For example, for desktop images, users usually name the folders by locations or event names, and for Web images, generally there are plenty of textual descriptions from which the initial keyword can be chosen.

We shrink from discussing the first step and leave it as another future work, and start our approach from the second step, thus the image annotation approach formulate our current proposal is given in Eq.5 below:

\[
\begin{align*}
w^* &= \arg \max_w p(w | I_t, w_p) \\
&= \arg \max_w \sum_p p(\Theta_q | I_t, w_p) \cdot p(\Theta_q | I_t, w_p) \\
&= \arg \max_w \left[ \arg \max_t p(w | t) \cdot p(t | \Theta_q) \right] \cdot p(\Theta_q | I_t, w_p)
\end{align*}
\]

\(5\)

Sketch of the Proposed Modeless Image Annotation Approach

The flowchart of the modeless image annotation process of mining annotations from image search results is shown in Fig. 2. It contains three steps: the search stage (labeled by “(1)” in the figure), the mining stage (“(2)”) and the filtering stage (“(3)”). The uncaptioned image and an initial keyword “sunset” serve as the queries. The highlighted yellow block shows the output annotations.

As a pre-processing step, a large scale Web image dataset is firstly crawled and reserved as a pre-requisition of a data-driven approach, which will be illustrated in Section IV.C, and then visual and textual indices of the images are built up to support the real-time process, on which we give the details in Section IV.D.

Given the query image, we adopt the traditional Query-By-Example (QBE) method in CBIR to retrieve visually similar images. Intuitively, when the image database contains millions or billions of images, image retrieval based on pair-wise computation of the Euclidean distance is too time consuming and thus impractical, and we solve this problem by two means: Firstly, Query-By-Keyword (QBK) method is conducted before QBE retrieval, and thus we can leverage the inverted indexing technique to retrieve the “semantically” similar images in real time (the time complexity is O(1)). Normally, the number of candidate images that are possibly relevant to the query will be greatly reduced in this step, and hence improves a great deal of the efficiency of the QBE retrieval step. Secondly, instead of calculating the Euclidean distances of two images based on the original low-level feature extracted, we encode each
feature vector into a hash code which is binary bitwise, so that efficient distance measures such as the Hamming distance can be leveraged to greatly increase the search efficiency.

We detail this step in Section IV.D. In this way, we obtain a small subset of images that are both visually and semantically similar to the query image, which concludes the search stage.

The next step is to mine a number of terms (i.e. words or phrases) from the textual descriptions of the search results, which we adopt a previous work by Zeng et al. (Zeng, et al., 2004) named SRC (SRC, 2006), as described in Section III for this task. The cluster names output by SRC are assumed the topics in Eq.5.

Then an annotation rejection step which we call “filtering” is applied onto $t$ to approximate $p(w | t)$ as annotation selection. The intention of this step is to further improve the precision of the output annotations for two reasons: (1) as aforementioned, the SRC technique (Zeng, et al., 2004) is based on textual features only and does not take visual features into consideration, hence the cluster names learnt may contain unsuitable terms for the query image. (2) Current SRC generates about 20 clusters, which is too large a number to promise high precision in our image annotation scenario. We detail this step in Section IV.E.

The user interface of our propose approach is shown in Fig. 3. It supports various query image suggestion schemes – the user can either upload a query image or select one from its QBE search results; s/he can type in a query term and select a query from its QBK results, or click the “Random” button to randomly select one.

The gray block highlights the query image and when clicking on the word “annotate,” the annotation process begins. The statistical number of time cost in search and mining are shown in the blue bar below the query image. As for the real example in Fig. 3, 500 images are retrieved from the 2.4 million Web image database and the time cost in the search and mining process are 15 and 563 milliseconds respectively, which is in real time. The terms in blue are the output annotations, which are cluster names yielded by SRC (Zeng, et al., 2004) and separated with the sign “|”. In Fig. 3, three terms survived the annotation filtering step. The underneath tab form displays a subset of the 500 relevant images retrieved in the search stage whose textual descriptions took part in the SRC mining.

In the following sections, we give the details of both technical and engineering designs which support effective data-driven annotation in real time.

**Dataset Selection: One Factor to Ensure an Effective Data-Driven Approach**

Since it is not only a very tedious and expensive task to manually label training data, but also very difficult to produce consistent labeling (Naphade, et al., 2006), the learning approaches always suffer from the small training data problem which degrades their power in handling outliers. For example, most previous works propose their researches based on Corel images, which contains only about 60 thousands high-quality images, and the images of the same category tend to be very similar visually.

On the other hand, recall that to ensure the effectiveness our modeless and data-driven approach of annotating images by mining search results, a basic assumption is that we can find near-duplicate images, or images that share semantics with the uncaptioned query image, so collecting a large scale image dataset is necessary. Due to these reason, we resort to the Web.

In fact, recently, some researchers have begun to investigate the usage of Web images and reported its effectiveness in many areas (Yeh, Tollmar, & Darrell, 2004, Weijer, Schmid, & Verbeek, 2006, Li, Wang, & Fei-Fei, 2007). However, since Web images are generally noisy and their surrounding text may
be of very low quality, it is not necessary that any Web images will help and hence post-processing is generally applied to ensure the quality of collected Web images (Weijer, et al., 2006, Li, et al., 2007). Hence in our approach, we collect 2.4 million images from a few photo forums. The advantages are that (1) since these photos are taken by photographers, they generally have high resolution; (2) when uploading their works, the photographers will provide comments which more or less describe the photos; (3) most of the photos are nature images which have simpler semantics compared to artificial ones or portraits like the Corbis dataset; (4) there are plenty of images which are of the same semantic but with diverse appearance, which ensures a higher generalization ability of the proposed approach than using Corel images as the training data.

Three examples from the 2.4 million photo forum images are shown in Fig. 4. We can see that although the descriptions are noisy, more or less they hit a few contents in the corresponding images (e.g. “tiger” in the first example) or suggest keywords (e.g. “forest”) that statistically co-occur with the central objects.

Table I provides statistics on the 2.4 million photos. The dictionary contains 362,669 words and each photo has 19.6 textual descriptions on average, which to our knowledge, is the largest database used in image annotation, without saying real-time annotation systems.
Annotating Images by Mining Image Search

The Promise of Real-Time

The obstacles to real-time annotation in our case include the content-based retrieval process and the handling of the 2.4 million images. We solve the first problem by mapping image visual feature vectors into hash codes and the second one by setting up a distributed system.

Accelerating the Search Process by Mapping Visual Features to Hash Codes

Typically, the similarity of two images is measured in Euclidean space. However, since visual features are generally of high dimensional, measuring pair-wise Euclidean distance becomes a bottleneck in retrieval efficiency even if the search space is greatly reduced by text-based search given the query keyword. To accelerate the content-based search process, an effective way is to compress the images. Some previous works proposed vector quantization techniques, for example, Zhu, Zhang, Rao, & Srihari (2000) segments an image into regions, and quantizes the regions to obtain a few keyblocks. The widely used inverted index technique in text-based search field is thus naturally applied onto such a kind of keyblock-based discrete representation, and in this way, the content-based image retrieval problem is converted seamlessly into a text-based one which greatly improves the retrieval efficiency.

In this paper, we adopt another technique of image hash code generation (Fan, et al., 2005, Wang, Li, Li, & Ma, 2006) since it is more scalable than vector quantization methods (Zhu, et al., 2000) and hence is better fit for large-scale databases.

**Related Works on Hash Encoding.** The hash code generation algorithms (Fan, et al., 2005, Wang, et al., 2006) were originally proposed to detect duplicate images, i.e. images which are visually identical to each other. The basic idea is: suppose that visual features are mapped into bit streams, with higher bits representing more important contents of an image, and lower bits representing less important contents, thus duplicate images should have equal hash codes (i.e. the bit streams), and the equality is easily measured by the Hamming distance (i.e. the “AND” operation). This technique can also be applied to discover near-duplicates, by comparing only the higher bits of two hash codes. Intuitively, since the higher bits represent the more important contents, so that two images are more possibly alike if more highest bits match.

The hash code generation algorithm proposed in (Fan, et al., 2005) is shown in Fig. 5. Firstly, it transforms an input color image to gray scale and divides it evenly into 8x8 blocks. Each block is represented by the average intensity so that the original image is converted into an 8x8 matrix M with each element $I_y$ in Eq.6.

{| Table 1. Statistics on the 2.4 million database |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of images indexed</td>
</tr>
<tr>
<td>No. of images having titles or comments</td>
</tr>
<tr>
<td>No. of distinctive words</td>
</tr>
<tr>
<td>No. of distinctive words with frequency ≥ 10</td>
</tr>
<tr>
<td>Average doc length of “title” field</td>
</tr>
<tr>
<td>Average doc length of “comment” field</td>
</tr>
</tbody>
</table>
Figure 4. Three examples of the 2.4 million photos collected from a few photo forum websites. These photos are generally of high quality and have comments from the photographer.

<table>
<thead>
<tr>
<th>Title</th>
<th>Categories</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiger Grooming</td>
<td>Animal, Nature</td>
<td>I saw this tiger grooming itself at the Miami Zoo. It resembled me of a grey pet cat.</td>
</tr>
<tr>
<td>Butterfly</td>
<td>Animal</td>
<td>Took this pic around KhaoKan forest last week.</td>
</tr>
<tr>
<td>Snowy Water Fall</td>
<td>Landscape, Nature</td>
<td>It's Going on Three weeks now that I've had my camera and as time goes on my pictures Quality is getting better...I think and Hope :)</td>
</tr>
</tbody>
</table>

Figure 5. The hash code generation process proposed in (Fan, et al., 2005), which is originally used for identical image search. This figure is adapted from (Fan, et al., 2005).

\[ I_y = \frac{\sum_{x=0}^{w-1} \sum_{y=0}^{h-1} \text{Int}(x, y)}{w \times h} \quad (6) \]

where \( \text{Int}(x, y) \) is the intensity of the \((i, j)\)-th matrix entry, and \( w, h \) denote the block width and height respectively.

2-D DCT transformation (Rao and Yip, 1990) is then applied to \( M \). The DC coefficient, which is the average of \( M \), is omitted to eliminate the effect of luminance. The remaining AC coefficients are zigzagged into a sequence, and the first 48 coefficients in lower frequencies are collected as the feature vector \( X_m \) of the original image, which is further mapped to a \( n \)-dimensional vector \( Y_n \) by a PCA (Joliffe, 1986) model \( P^T \) trained on 5,500 Web, as shown in Eq.7:

\[ Y_n = P^T X_m \quad (7) \]

The hash code is thus generated from \( Y_n \) using the encoding function shown in Fig. 6.
Wang, et al. (2006) proposed a similar idea but is more efficient, as shown in Fig. 7. Images are hierarchically divided into blocks and average luminance is used as features. The projection model is still PCA trained from 11 million iFind (Chen, Liu, Hu, Li, & Zhang, 2001) images, and the 24 eigenvectors which correspond to the 24 largest eigenvalues are retained. The intuition behind using PCA for dimension reduction is that the PCA space is essentially a rotated version of the original feature space. If all the feature dimensions are kept, it draws the same conclusion on data points’ distances as measured in the Euclidean space. Moreover, since PCA dimension reduction is achieved by cutting off the lower variance dimensions, the information loss is so small that can be ignored, while on the other hand, the hash code based matching can be significantly speeded up.

Mapping Visual Features to Hash Codes. Though it is proposed in duplicate finding, the same technique can be applied to accelerate search process with low information loss. Recall that since the higher bits represent the more important contents in an image, if some hash codes have equal n highest bits, they are thus indexed by the same key. Henceforth we can significantly speed up the search process in visual space by creating an inverted index based on the hash codes.

In our approach, we use 36-bin color Correlogram (Huang, et al., 1997) instead of intensity and luminance (Fan, et al., 2005, Wang, et al., 2006) as the original visual features. The reasons are two-fold: (1) the goal of (Fan, et al., 2005, Wang, et al., 2006) is to find duplicate images which are identical both in layout and contents, hence luminance of gray-scale images is an effective feature which is also efficient. However, our goal is to find relevant images — images that may not have the same layout and contents but share similar concepts. Hence to keep the color, texture and shape, etc., features is important in our case. Obviously Correlogram features keep more information than the gray-image based ones; and (2) Correlogram is also a widely used feature in content-based image retrieval (Rui, Huang, & Chang, 1999, Carson, Thomas, Belongie, Hellerstein, & Malik, 2002, Cox, Miller, Minka, Paphathomas, & Yianilos, 2000).

Since 3-channel (e.g. RGB) features are used in our case, rather than the 1-channel one in (Fan, et al., 2005), 2-D DCT transformation is inapplicable here, and hence we adopt the hash code generation approach proposed by Wang, et al. (2006). However, we extract the 144-dimensional color Correlo-

Figure 6. The hash code generation algorithm proposed in (Fan, et al., 2005)

| Input: feature vector \( Y = (y_i): i \text{ ranges from 0 to } n-1 \) |
| For each element \( y_i \) |
| \begin{align*}
| & \text{If } y_i > 0, \text{ the } i \text{th bit of signature is set to 1} \\
| & \text{Else the } i \text{th bit of signature is set to 0} \\
| \end{align*} |
| End |
| Output: signature |

Figure 7. The hash code generation process proposed in (Wang, et al., 2006)
grams for each image and learn the 144x32 PCA projection model based on the same 11 million iFind images as in (Wang, et al., 2006). Then we use this model to offline map all the 2.4 million photos we collected into hash codes.

**Hash Code Based Image Retrieval.** Each image is now represented by a 32-dimensional bit streams, with the higher bits represent the major information while the lower bits describe the less informative ones. We design and compared four distance measures in our current implementation to check the effectiveness and efficiency of hash code based image retrieval, the results are given in Section V. The four distance measures are listed below:

- **Hash code filtering plus Euclidean distance.** If the highest n bits of an image exactly match those of the query image, then the Euclidean distance of these two images are measured based on their Correlograms. This measure is a tradeoff between search effectiveness and efficiency since the time consuming Euclidean distance computing is required only for a small set of images. On the other hand, mapping Correlograms to hash codes brings information loss, thus calculating image similarities based on the original features may ensure a better effectiveness. In our experiments, n = 20.
- **Hamming distance.** It measures the number of different bits of two hash codes. Obviously this is the most efficient measurement. Its effectiveness depends on that of the hash code generation approach.
- **Weighted Hamming distance.** Hamming distance assume all bits have equal weights. But intuitively, since the higher bits are more important, differences in higher bits should be larger-weighted. We propose a weighting function as below:
  Firstly, we evenly group a 32-bit hash code into 8 bins. From left to right, each bin contains 4-bits. Then we weight the Hamming distance on the ith bin by $2^{(8-i)}$, $1 \leq i \leq 8$. Obviously the weighting function magnifies the difference on the higher bits. It is simple but effective as proved in our experiments.
- **Euclidean distance.** The Euclidean distance based on original Correlogram features is also given as a baseline to assess the performance of the above hash code-based measurements.

We rank the images in the ascending order of the distances respectively and return the top N as search results.

**Efficiently Handling the Large-Scale Database with Distributed Computing**

In part 1), we demonstrate the method of supporting content-based search in real time and proposed a hash code generation approach as well as different distance measures. However this is not enough to support real time image annotation when the database contains millions or billions of images. In this subsection, we describe our design of a distributed system, which as shown in Fig. 3, finishes the search process in 15 milliseconds and the mining process in 563 milliseconds, based on 2.4 million images.

The system architecture is shown in Fig. 8. The key is that the content-base search engine, text search engine and SRC clustering engine are provided as Web services using C# Remoting technique. Each service register a distinct TCP port and listening to it, when there is a service request, the service accepts inputs, does its own work, and sends back outputs. Moreover, by building up this distributed system, both visual and textual features can be kept in memories of servers, and no disk I/O is requested, so obviously it is quite efficient and easily scalable.
The Control of SRC (Zeng, et al., 2004) Outputs

As mentioned in Section IV.A, we use the search result clustering technique named SRC (Zeng, et al., 2004) to learn the topics based on the visually and semantically similar search results.

A left problem is that SRC requires users to provide the number of clusters $|n_{src}|$ wanted, which is difficult to select without observing the data. We provide an experiential algorithm as Eq. 8 below to select $|n_{src}|$:

$$|n_{src}| = \max \left( |\Theta^*| / 200, 4 \right)$$

where $|\Theta^*|$ is the number of retrieved images. 200 is an empirical value of cluster size suggested by SRC to ensure the saliency of learned cluster names. On the other hand, if $|\Theta^*|$ is too small, SRC tends to group all images in one or two clusters and hence images in one cluster may be too diverse to produce meaningful cluster names. To avoid this, we force the algorithm to output at least 4 clusters, while 4 is another empirically selected parameter.

Moreover, because SRC extracts all n-grams $\Theta^*(n \leq 3)$ as its candidate key phrases, if $\Theta^*$ is too large, time cost of the SRC service may be very high. Hence, we set $\max \Theta^* = 200$.

The final problem is annotation rejection. As aforementioned (refer to Section IV.B), it is unsuitable to output all cluster names yield by SRC as the annotations, hence to define criteria for annotation rejection is necessary to ensure the performance. Two criteria are compared in our current implementations:

- **Maximum cluster size criterion.** This is a very simple but effective annotation rejection method, which uses the size of each cluster, i.e. the number of images in each cluster, as the score to rank each cluster name, and the top ranked ones are assigned onto the query image as the final annotations. This is equivalent to the Maximum a Posteriori estimation (MAP) which assumes that “the majority speaks the truth” and is statistically reasonable since the cluster names are learned from search results which are assumed visually and semantically relevant to the query image.

- **Average member image score criterion.** The term “member image” of a cluster A indicates an image that is assigned to A, and the cluster score is measured as the average similarity of its member images to the query image. It is reasonable since the more relevant member images a cluster contains, the more probably the concepts learned from this cluster represents the content of the query image.

We keep the top ranked clusters, merge their cluster names by removing the duplicate words and phrases, and output as the learned annotations, which closes the entire system process.

**EVALUATIONS**

2.4 million photos from several online photo forums are collected as the database, from which the relevant images are retrieved to annotate the query image. These photos are not only of high quality, but also have rich descriptions, as discussed in Section IV.C.

In order to evaluate the effectiveness and efficiency of our proposed approach, we conducted a series of experiments based on two datasets. One is an open dataset, of which the images are collected
from Google image search engine (Google, 2006). 30 Google images from 15 categories (as shown in Table II) were randomly selected. However, to give a more objective evaluation of the effectiveness, we deliberately selected a few vague query keywords, e.g. using “Paris” as the query keyword to annotate a photo of “Sacre Coeur”. Because no ground truth labels are available, we manually evaluated the retrieval performance on this dataset.

The other testing dataset is a benchmark content-based image retrieval database provided by the University of Washington (UW). Images in this dataset have about 5 manually labeled ground truth annotations on average, and the UW folder names were used as the query keywords (see Table III). A problem with this dataset is that for many images, not all objects are annotated. For a fair evaluation, we manually revised the results to accept synonyms and correct annotations that do not appear in UW labels.

**Experiments on Google Images**

**Evaluation Criterion**

The evaluation of image auto-annotation approaches is still an open problem. Typically in many previous works (Blei, et al., 2003, Carneiro and Vasconcelos, 2005, Jeon and Manmatha, 2004), image retrieval technique is used for evaluation. Generally in these approaches, the entire dataset is separated into two parts, one for training and the other for testing such that images in these two datasets share almost the same vocabularies and data distributions, and ground truth annotations are available for the testing.

*Figure 8. System architecture of the proposed modeless and data-driven image annotation approach. The visual search engine, text search engine and SRC engine are implemented as services, which make up of a distributed system using C# Remoting technique.*
Annotating Images by Mining Image Search

dataset. Thus in their evaluation, a number of keywords can be selected from the vocabulary to query
the annotated testing dataset and the precision and recall are calculated for the returned images.

Let \( G \) be the ground truth relevant images and \( L \) be the retrieved images, retrieval precision and
recall are defined as in Eq.9.

\[
\text{Precision} = \frac{|L \cap G|}{|L|}, \quad \text{Recall} = \frac{|L \cap G|}{|G|}
\]  

(9)
i.e. precision evaluates the proportion of relevant images in the retrieval results, while recall calculates
that of the relevant images in all the relevant images contained in the database.

However, Eq.9 suggests only the probability of identifying “correct” annotations. But for an annotation
system, generally the output annotations belong to three categories: “perfect”, “correct”, and “wrong”.
The “perfect” annotations hit the specific objects inside an image, e.g. “tiger” for the first image in
Fig. 4. The “correct” annotations are those that are not-wrong but also not perfect, e.g. “France” for an
image about Eiffel tower. The “wrong” annotations are the incorrect outputs, which have nothing to do
with the contents of the query image. We believe that a comprehensive evaluation should take all these
three categories into consideration. Hence we propose Eq.10 to evaluate the performance. It modifies
the normalized score measure (Barnard, et al., 2003, Li and Wang, 2006), which only categorizes the
annotations into “right” or “wrong”.

\[
E = \frac{p + 0.5 \times r - w}{m}
\]

(10)
where \( m \) denotes the number of annotations predicted. \( p, r, w \) are the number of “perfect”, “correct”,
and “wrong” annotations respectively. Note that to emphasize the preference for “perfect” annotations,
we punish the “correct” ones by a lower weight 0.5. Obviously when all the predictions are “perfect”,
\( E = 1 \), while if all are wrong, \( E = -1 \).

Since there are no ground truth annotations available for this dataset, it is impossible for us to evalu-
ate the recall performance.

System Effectiveness

Fig. 9 shows how annotation precision varies as the similarity weight changes. This weight weights the
average similarity of images retrieved in the content-based search stage, and the product serves as the
threshold to filter out irrelevant images, i.e. images whose similarity to the query image is lower than

Table 2. Queries from google

| Apple, Beach, Beijing, Bird, Butterfly, Clouds, Clownfish, Japan, Liberty, Lighthouse, Louvre, Paris, Sunset, Tiger, Tree |

Table 3. Queries from U.washington

| Australia, Campus, Cannon beach, Cherries, Football, Geneva, Green lake, Indonesia, Iran, Italy, Japan, San juan, Spring flower, Swiss mountain, Yellowstone |

339
Figure 9. Annotation precision measured by Eq. 10. It shows respectively the precision curve of using the four distance measures described in Section IV.D vs. the similarity weight as a threshold to control the number of image search results yield at the search stage.

(a) Precision w.r.t. maximum cluster size criterion

(b) Precision w.r.t. average member image score criterion

this threshold will not be delivered to the SRC mining stage. This parameter determines $\Theta'$ in Eq.8 and thus directly affects the learned clusters and the predicted annotations.

The reason of proposing such a threshold strategy is that, since the similarity of images varies greatly; it is very difficult to select a hard threshold which can promise high system performance for any queries. On the contrary, our soft threshold is query-dependent, it leverages the information provided by content features of the image search results.
The green square curve in Fig. 9 represents the process without the content-based search step, i.e. it relies purely on text-based search to yield near duplicates, and SRC clustering is based on the images that are “semantically” similar but may not visually similar. It serves as the baseline method here. Also, since no visual features are available, annotations predicted by this method uses the maximum cluster size criterion (see Section IV.E).

Fig. 9(a) shows annotation precision curves of the four content-based search distance measures (refer to Section IV.D) under maximum cluster size criterion.

From this figure, we can see that the weighted Hamming distance measure performs the best. This is reasonable because it emphasizes the feature components that captures the important information in an image while at the same time de-emphasize the unimportant ones.

It is interesting that Euclidean distance measure based on the original Correlogram features performs comparatively to the Hamming distance measure in effectiveness. This shows that the information-loss of mapping original features into hash codes (refer to Section IV.D) can be ignored on this dataset, or say, the model for hash code generation is effective.

Another interesting result is that the hash code filtering plus Euclidean distance method performs badly. The reason may be that the higher 20 bits of hash codes learned are too coarse to distinguish relevant images from irrelevant ones.

All the distance measures perform much better than the baseline method. This shows that requiring the visual similarity of clustering images is necessary and important.

Fig. 9 (b) shows the performance with maximum average member image score criterion. It is generally worse than that with the maximum cluster size criterion. A possible reason is due to the semantic gap. Recall that SRC algorithm clusters images purely based on their surrounding texts and ignores image visual features. Thus images in one cluster may have very different visual features even if they belong to the same category. Obviously this will not affect the maximum cluster size criterion but only the average member image score criterion, because the latter one uses visual similarity to score the clusters. Another possible reason may be that the images which share similar descriptions vary greatly in their visual appearances.

Note that the system’s performance jumps when the threshold is too large such that $\Theta^*$ is too small to ensure satisfying SRC clustering performance.

Fig.10 shows a few examples of the annotation results. The boldfaced keywords indicate the queries. It can be seen that our approach is able to find correct and most of the time perfect complementary annotations.

System Efficiency

We have give the readers a sense of the efficiency in Fig. 3, which shows that our annotation process is fulfilled in real time, although the database contains 2.4 million images, which is the largest till now to our knowledge.

In the subsection, we conduct one more experiment to evaluate the efficiency statistically. For all the queries, we collect their search results, which are about 24,000 images on average, and test the system efficiency according to the four distance measures: Hamming Distance, Weighted Hamming Distance, Hash code filtering plus Euclidean distance, and the traditional Euclidean distance based on original Correlogram features. The hardware environment is a computer with one Dual Intel Pentium 4 Xeon hyper-threaded CPU and 2G memory. The time cost for computing the pair-wise distance between each
Figure 10. A few examples of the annotation results yield by the modeless and data-driven image annotation approach

<table>
<thead>
<tr>
<th>Paris</th>
<th>Paris</th>
<th>Paris</th>
<th>Paris</th>
</tr>
</thead>
<tbody>
<tr>
<td>Las vegas,</td>
<td>Sacre coeur,</td>
<td>Eiffel tower,</td>
<td>Eiffel tower,</td>
</tr>
<tr>
<td>eiffel tower,</td>
<td>parish building,</td>
<td>france, sky,</td>
<td>france, sky,</td>
</tr>
<tr>
<td>love paris</td>
<td>eiffel tower</td>
<td>paris nights</td>
<td>paris nights</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sunset</th>
<th>Tiger</th>
<th>Tree</th>
<th>Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lake, tree,</td>
<td>White tiger,</td>
<td>House, flower,</td>
<td>House, flower,</td>
</tr>
<tr>
<td>mountain,</td>
<td>usa, zoo</td>
<td>snow, sky,</td>
<td>snow, sky,</td>
</tr>
<tr>
<td>sky,</td>
<td></td>
<td>tree trunk</td>
<td>tree trunk</td>
</tr>
<tr>
<td>beautiful,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Apple</th>
<th>Beach</th>
<th>Butterfly</th>
<th>Butterfly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studio,</td>
<td>South america,</td>
<td>Flower,</td>
<td>Flower,</td>
</tr>
<tr>
<td>kitchen,</td>
<td>beautiful beach,</td>
<td>butterfly house,</td>
<td>butterfly house,</td>
</tr>
<tr>
<td>fruit,</td>
<td>beach house</td>
<td>beautiful butterfly,</td>
<td>beautiful butterfly,</td>
</tr>
<tr>
<td>color</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Apple</th>
<th>Beach</th>
<th>Butterfly</th>
<th>Clou,d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fruit,</td>
<td>Sky, island,</td>
<td>Yellow</td>
<td>National park,</td>
</tr>
<tr>
<td>apple tree</td>
<td>sun beach,</td>
<td>butterfly,</td>
<td>europe,</td>
</tr>
<tr>
<td></td>
<td>sunrise,</td>
<td>swallowtail,</td>
<td>south america,</td>
</tr>
<tr>
<td></td>
<td>beach island</td>
<td>nature</td>
<td>blue sky</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beach</th>
<th>Beach</th>
<th>Clouds</th>
<th>Clouds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morn, sand,</td>
<td>beach resort,</td>
<td>National park,</td>
<td>National park,</td>
</tr>
<tr>
<td>south beach,</td>
<td>white beach,</td>
<td>europe,</td>
<td>europe,</td>
</tr>
<tr>
<td>sun beach</td>
<td>South beach</td>
<td>south america,</td>
<td>south america,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue sky</td>
<td>blue sky</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clouds</th>
<th>Clownfish</th>
<th>Liberty</th>
<th>Liberty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dark clouds,</td>
<td>Anemone, reef,</td>
<td>York, liberty</td>
<td>York, liberty</td>
</tr>
<tr>
<td>sun, sky,</td>
<td>red sea</td>
<td>statue, sun</td>
<td>statue, sun</td>
</tr>
<tr>
<td>sunrise,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mom</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 11. Search efficiency vs. the four distance measures. Pair-wise distances are measured based on 24,000 images. It proves the efficiency of the proposed idea. Moreover, the Hamming distance is the most efficient, which is also reasonable.

![Time Cost vs. Distance Measure](image_url)

of the 24,000 images and the query image, as well as ranking the images accordingly, is shown in Fig. 11, which cost 0.034, 0.072, 0.051 and 0.122 seconds respectively. We can see that computing Euclidean distance is nearly 4 times slower of calculating Hamming distance.
Annotating Images by Mining Image Search

The hash code filtering plus Euclidean distance measure is the second efficient. The reason is that the hash codes here serve as the inverted indices to identify the relevant images which is of O(1) computation complexity. Time cost for this measure is consumed by the Euclidean distance calculation afterwards.

Note that the above evaluations were conducted with all features loaded into memory. If disk access is required, we can imagine that hash code-based measures will be even faster than the original visual feature-based ones.

Experiments on UW Dataset

In this subsection, we evaluate the performance of our proposed approach based on the benchmark U.Washington database.

Because ground truth is available and this dataset is comparably small, we use precision and recall as the evaluation criteria. Fig. 12 shows the maximum precision and recall of our approach w.r.t. the two annotation rejection criteria vs. the four distance measures. All images in the database are used for the evaluation. Again the weighted Hamming distance measure performs the best.

An interesting point is that the average member image score criterion now works better. This is because few images in our 2.4 million photograph database are visually similar to the UW images, while

Figure 12. Precision and recall yielded corresponding to the four distance measure given the two annotation rejection criteria based on all the images in UW dataset.
the UW images of the same category share similar visual appearance. Moreover, the average image score strategy helps to rank higher the clusters whose member images are more visually relevant to the query thus is less biased by the irrelevant member image descriptions.

It is worth mentioning that the real performance of our system is actually much better than shown in Fig. 12. Since the evaluation shown in Fig.12 did not grant synonyms, e.g. “beach” and “coast”, and semantically relevant keywords, e.g. “Geneva” and “Switzerland”, as correct answers. Moreover, UW ground truth annotations may ignore some contents of an image, so that our current evaluation assumes them to be “wrong” even if the predicted annotations are correct, just because they do not appear in UW ground truth. To prove this, we manually examined the predicted annotations of 100 randomly selected queries. The corrected precision and recall are 38.14% and 22.95% respectively, nearly 12% precision improvement, with weighted Hamming distance measure, the average member image score criterion, and the similarity weight 1.2.

Fig. 13 show four examples of our output, which hit no UW ground truth labels and gives zero precisions in the strict evaluation but are indeed correct answers.

Moreover, since our method is a modeless one which has no supervised learning stage, and the UW images are “outliers” to our database from which few relevant images can be found, the task is much tougher for us than for the previous approaches. On the contrary, in most of the previous works, training data and testing data are selected from the same dataset and the training dataset is usually much larger than the testing dataset, e.g. (Barnard, et al., 2003, Blei and Jordan, 2003) use 4,500 Corel images for training and 500 images for testing, and the performance is around 20-30%. This shows that our system is more effective in predicting annotations, and is robust in handling outsiders.

**DISCUSSIONS**

Compared to the previous works, or traditional computer vision and machine learning ones which built up generative or discriminative annotation models, in this paper, we propose a novel attempt of modeless image annotation, which investigates how effective to a data-driven approach only can be, and suggest mining annotations for an uncaptioned image by mining its search results. It has at least three advantages: (1) no training dataset and supervised learning are required and hence is free from the problem of lack of training data; (2) since no training stage is available, and hence it requires no predefined vocabularies while vocabulary creation is still an open research topic (Naphade, et al., 2006); (3) by leveraging the Web data, we can obtain diverse images for the same semantics which ensure a
Annotating Images by Mining Image Search

high generalization capability or say the practical usage of the proposed annotation approach. This is a noticeable advantage to using the Corel Stock-style database in which images of the same concept have similar visual appearances with clean descriptions. Moreover, our approach is highly scalable and very robust to outsider queries.

CONCLUSION

Compared to the previous annotation approaches which build up generative or discriminative annotation models, in this paper, we propose a novel attempt of modeless image annotation, which investigates how effective a data-driven approach can be, and suggest mining annotations for an uncaptioned image by mining its search results. We collected 2.4 million images with their surrounding texts from a few photo forum websites.

The entire process contains three steps: (1) the search stage, which yields visually and semantically search results given the uncaptioned images as queries; (2) the mining stage, which applies a label-based clustering technique to mining topic labels from the textual descriptions of the search results as candidate annotations; (3) the annotation rejection stage, which filtered the candidate annotations to ensure the preciseness of the final outputs.

Compared with the previous works, our method saves labor both in training data collection and vocabulary creation, and is highly scalable and robust to outliers. And as to our knowledge, it is a real-time approach that handles the largest number of images.

To ensure real time annotation, two key techniques are leveraged—one is to map the high dimensional image visual features into hash codes, the other is to implement it as a distributed system, of which the search and mining processes are provided as Web services. As a typical result, the entire process finishes in less than 1 second.

Experiments conducted on both an open database (Google images) and a benchmark database (UW CBIR database) proved the effectiveness and efficiency of this proposed approach.

There are much room to improve our proposed approach, e.g. as mentioned in Section IV.B, to propose label-based clustering technique which takes also image visual features into consideration is necessary and important for further performance improvement. And we would like to investigate how much performance gain we can obtain by embedding learning approaches into our current implementation.

ACKNOWLEDGMENT

The authors would like to thank Feng Jing, Hua-Jun Zeng, Kefeng Deng, Bin Wang and Le Chen for their sincere helps on technical discussions, assistances, as well as system implementations. Many thanks should also be given to the volunteers that help in manually evaluating our experimental results.

REFERENCES


**ENDNOTES**

1. Note that we do not require it to be perfect, at least in our approach, e.g. given an image about the Eiffel Tower, the “perfect” keyword is “Eiffel” but an accurate one can be “France”.

2. This is a crucial technique for text search engines. It stores the keyword-document relationship into a so-called inverted index file, whose entries are keywords and values are the document ids
to represent which document is indexed by a given keyword, or say which document contains this keyword.

3 Although SRC accepts user-assigned number to indicate how many categories are preferred, it is a suggested number that statistically produces the highest performance.

4 We need to keep in mind that our system should be easily scale up, e.g. to support real-time annotation with billions of images.

5 http://www.cs.washington.edu/research/imagedatabase/groundtruth/