Real-Time Computerized Annotation of Pictures*

Jia Li
Department of Statistics
The Pennsylvania State University
University Park, PA, USA
jiali@psu.edu

James Z. Wang
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA, USA
jwang@ist.psu.edu

ABSTRACT

Automated annotation of digital pictures has been a highly challenging problem for computer scientists since the invention of computers. The capability of annotating pictures by computers can lead to breakthroughs in a wide range of applications including Web image search, online picture-sharing communities, and scientific experiments. In our work, by advancing statistical modeling and optimization techniques, we can train computers about hundreds of semantic concepts using example pictures from each concept. The ALIPR (Automatic Linguistic Indexing of Pictures - Real Time) system of fully automatic and high speed annotation for online pictures has been constructed. Thousands of pictures from an Internet photo-sharing site, unrelated to the source of those pictures used in the training process, have been tested. The experimental results show that a single computer processor can suggest annotation terms in real-time and with good accuracy.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

Image Annotation, Statistical Learning, Modeling, Clustering

*An on-line demonstration is provided at the URL:
`http://alipr.com`.
More information about the research:
`http://riemann.ist.psu.edu`.
Both authors are also affiliated with the Department of Computer Science and Engineering, and the Integrative Biosciences (IBIOS) Program.

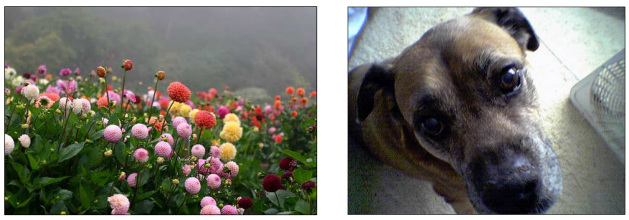
MM'06, October 23–27, 2006, Santa Barbara, California, USA.

1. INTRODUCTION

Image archives on the Internet are growing at a phenomenal rate. With digital cameras becoming increasingly affordable and the widespread use of home computers possessing hundreds of gigabytes of storage, individuals nowadays can easily build sizable personal digital photo collections. Photo sharing through the Internet has become a common practice. According to a report released in June 2005, an Internet photo-sharing startup, flickr.com, has almost one million registered users and hosts 19.5 million photos, with a growth of about 30 percent per month. More specialized online photo-sharing communities, such as photo.net and airliners.net, also have databases in the order of millions of images entirely contributed by the users.

1.1 The Problem

Image search provided by major search engines such as Google, MSN, and Yahoo! relies on textual descriptions of images found on the Web pages containing the images and the file names of the images. These search engines do not analyze the pixel content of images and hence cannot be used to search unannotated image collections. Fully computerized or computer-assisted annotation of images by words is a crucial technology to ensure the "visibility" of images on the Internet, due to the complex and fragmented nature of the networked communities.



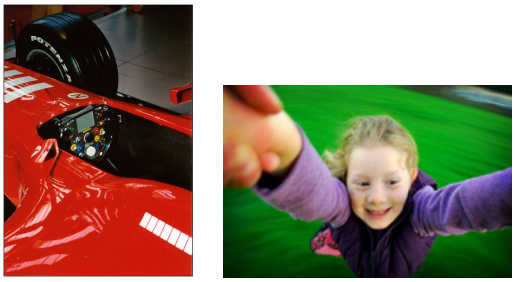
(a) (b)

Figure 1: Example pictures from the Website flickr.com. User-supplied tags: (a) 'dahlia', 'golden', 'gate', 'park', 'flower', and 'fog'; (b) 'cameraphone', 'animal', 'dog', and 'tyson'.

Although owners of digital images can be requested to provide some descriptive words when depositing the images, the annotation tends to be highly subjective. Take an example of the pictures shown in Figure 1. The users on flickr.com annotated the first picture by the tags 'dahlia', 'golden', 'gate', 'park', 'flower', and 'fog' and the second

picture by 'cameraphone', 'animal', 'dog', and 'tyson'. While the first picture was taken at the Golden Gate Park near San Francisco according to the photographer, this set of annotation words can be a problem because this picture may show up when other users are searching for images of gates. The second picture may show up when users search for photos of various camera phones.

A computerized system that accurately suggests annotation tags to users can be very useful. If a user is too busy, he or she can simply check off those relevant words and type in other words. The system can also allow trained personnel to check the words with the image content at the time a text-based query is processed. However, automatic annotation of images with a large number of concepts is extremely challenging, a major reason that real-world applications have not appeared.



(a) race car (b) spinning

Figure 2: Human beings can imagine objects, parts of objects, or concepts not captured in the image. The images were obtained from flickr.com.

Human beings use a lot of background knowledge when we interpret an image. With the endowed capability of imagination, we can often *see* what is not captured in the image itself. For example, when we look at the picture in Figure 2(a), we know it is a race car although only a small portion of the car is shown. We can imagine in our mind the race car in three dimensions. If an individual has never seen a car or been told about cars in the past, he is unlikely to understand what this picture is about, even if he has the ability to imagine. Based on the shining paint and the color of the rubber tire, we can conclude that the race car is of very high quality. Similarly, we realize that the girl in Figure 2(b) is spinning based on the perceived movements with respect to the background grass land and her posture. Human beings are not always correct in image interpretation. For example, a nice toy race car may generate the same photograph as in Figure 2(a). Computer graphics techniques can also produce a picture just like that.

Without a doubt, it is very difficult, if at all possible, to empower computers with the capability of imagining what is absent in a picture. However, we can potentially train computers by examples to recognize certain objects and concepts. Such training techniques will enable computers to annotate not only photographic images taken by home digital cameras but also the ever increasing digital images in scientific research experiments. In biomedicine, for instance, modern imaging technologies reveal to us tissues and portions of our body in finer and finer details, and with different modalities. With the vast amount of image data we generate, it has become a serious problem to examine all the data manually. Statistical/machine learning based technologies can potentially allow computers to screen such images before scientists spend their precious time on them.

1.2 Prior Related Work

The problem of automatic image annotation is closely related to that of content-based image retrieval. Since the early 1990s, numerous approaches, both from academia and the industry, have been proposed to index images using numerical features automatically-extracted from the images. Smith and Chang developed of a Web image retrieval system [19]. In 2000, Smeulders et al. published a comprehensive survey of the field [18]. Progresses made in the field after 2000 is documented in a recent survey article [5]. Due to space limitation, we review some work closely related to ours. The references listed below are to be taken as examples only. Readers are urged to refer to survey articles for more complete references of the field.

Some initial efforts have recently been devoted to automatically annotating pictures, leveraging decades of research in computer vision, image understanding, image processing, and statistical learning [2, 8, 9]. Generative modeling [1, 13], statistical boosting [20], visual templates [4], Support Vector Machines [22], multiple instance learning, active learning [26, 10], latent space models [15], spatial context models [17], feedback learning [16] and manifold learning [23, 11] have been applied image classification, annotation, and retrieval.

Our work is closely related to generative modeling approaches. In 2002, we developed the ALIP annotation system by profiling categories of images using the 2-D Multiresolution Hidden Markov Model (MHMM) [13, 25]. Images in every category focus on a semantic theme and are described collectively by several words, e.g., "sail, boat, ocean" and "vineyard, plant, food, grape". A category of images is consequently referred to as a *semantic concept*. That is, a concept in our system is described by a set of annotation words. In our experiments, the term concept can be interchangeable with the term category. To annotate a new image, its likelihood under the profiling model of each concept is computed. Descriptive words for top concepts ranked according to likelihoods are pooled and passed through a selection procedure to yield the final annotation.

Barnard et al. [1] aimed at modeling the relationship between segmented regions in images and annotation words. A generative model for producing image segments and words is built based on individually annotated images. Given a segmented image, words are ranked and chosen according to their posterior probabilities under the estimated model. Several forms of the generative model were experimented with and compared against each other.

The early research has not investigated real-time automatic annotation of images with a vocabulary of several hundred words. For example, as reported [13], the system takes about 15-20 minutes to annotate an image on a 1.7 GHz Intel-based processor, prohibiting its deployment in the real-world for Web-scale image annotation applications.

1.3 Contributions of the Work

We have developed a new annotation method that achieves real-time operation and better optimization properties while preserving the architectural advantages of

the generative modeling approach. Models are established for a large collection of semantic concepts. The approach is inherently cumulative because when images of new concepts are added, the computer only needs to learn from the new images. What has been learned about previous concepts is stored in the form of profiling models and needs no re-training.

The breakthrough in computational efficiency results from a fundamental change in the modeling approach. In ALIP [13], every image is characterized by a set of feature vectors residing on grids at several resolutions. The profiling model of each concept is the probability law governing the generation of feature vectors on 2-D grids. Under the new approach, every image is characterized by a statistical distribution. The profiling model specifies a probability law for distributions directly.

We show that by exploiting statistical relationships between images and words, without recognizing individual objects in images, the computer can automatically annotate images in real-time and provide more than 98% images with at least one correct annotation out of the top 15 selected words. The highest ranked annotation word for each image is accurate with a rate above 51%. These quantitative results of performance are based on human subject evaluation of computer annotation for over $5,400$ general-purpose photographs.

A real-time annotation demonstration system, ALIPR (Automatic Linguistic Indexing of Pictures - Real Time), is provided online [1]. The system annotates any online image specified by its URL. The annotation is based only on the pixel information stored in the image. With an average of about 1.4 seconds on a 3.0 GHz Intel processor, annotation words are created for each picture.

The contribution of our work is multifold:

- We have developed a real-time automatic image annotation system. This system has been tested on Web images acquired completely independently from the training images. Rigorous evaluation has been conducted. To our best knowledge, this work is the first attempt to manually assess the performance of an image annotation system at a large scale. Data acquired in the experiments will set yardsticks for related future technologies and for the mere interest of understanding the potential of artificial intelligence.

- We have developed a novel clustering algorithm for objects represented by discrete distributions, or bags of weighted vectors. This new algorithm minimizes the total within cluster distance for a data form more general than vectors. We call the algorithm D2-clustering where D2 stands for discrete distribution. A new mixture modeling method has been developed to construct a probability measure on the space of discrete distributions. Both the clustering algorithm and the modeling method can be applied broadly to problems involving data other than images.

1.4 Outline of the Paper

The remainder of the paper is organized as follows: In Sections 2 and 3, we provide details of the training and annotation algorithms, respectively. The experimental results are provided in Section 4. We conclude and suggest future work in Section 5.

[1]Demonstration URL: `http://alipr.com`

2. THE TRAINING ALGORITHM

The training procedure is composed of the following steps. An outline is provided before we present each step in details. Label the concept categories by $\{1, 2, ..., M\}$. For the experiments, to be explained, using the Corel database as training data, $M = 599$. Denote the concept to which image i belongs by g_i, $g_i \in \{1, 2, ..., M\}$.

1. Extract a signature for each image i, $i \in \{1, 2, ..., N\}$. Denote the signature by s_i, $s_i \in \Omega$. The signature consists of two discrete distributions, one of color features, and the other of texture features. The distributions on each type of features across different images have different supports.

2. For each concept $m \in \{1, 2, ..., M\}$, construct a profiling model $\mathcal{M}_m$ using the signatures of images belonging to concept m: $\{s_i : g_i = m, 1 \leq i \leq N\}$. Denote the probability density function under model $\mathcal{M}_m$ by $f(s \mid \mathcal{M}_m)$, $s \in \Omega$.

Figure 3 illustrates this training process. The annotation process based upon the models will be described in Section 3.

2.1 The Training Database

It is well known that applying learning results to unseen data can be significantly harder than applying to training data [21]. In our work, we used completely different databases for training the system and for testing the performance.

The Corel image database, used also in the development of SIMPLIcity [24] and ALIP [13], containing close to $60,000$ general-purpose photographs is used to learn the statistical relationships between images and words. This database was categorized into 599 semantic concepts by Corel during image acquisition. Each concept, containing roughly 100 images, is described by several words, e.g., "landscape, mountain, ice, glacier, lake", "space, planet, star." A total of 332 distinct words are used for all the concepts. We created most of the descriptive words by browsing through images in every concept. A small portion of the words come from the category names given by the vendor. We used 80 images in each concept to build profiling models.

2.2 Preliminaries

To form the signature of an image, two types of features are extracted: color and texture. The RGB color components of each pixel are converted to the LUV color components. We use wavelet coefficients in high frequency bands to form texture features. A Daubechies 4 wavelet transform [6] is applied to the L component (intensity) of each image. The LH, HL, and HH band wavelet coefficients (in absolute values) corresponding to the same spatial position in the image are grouped into one three dimensional texture feature vector. The three dimensional color feature vectors and texture feature vectors are clustered respectively by k-means. The number of clusters in k-means is determined dynamically by thresholding the average within cluster variation. Arranging the cluster labels of the pixels into an image according to the pixel positions, we obtain a segmentation of the image. We

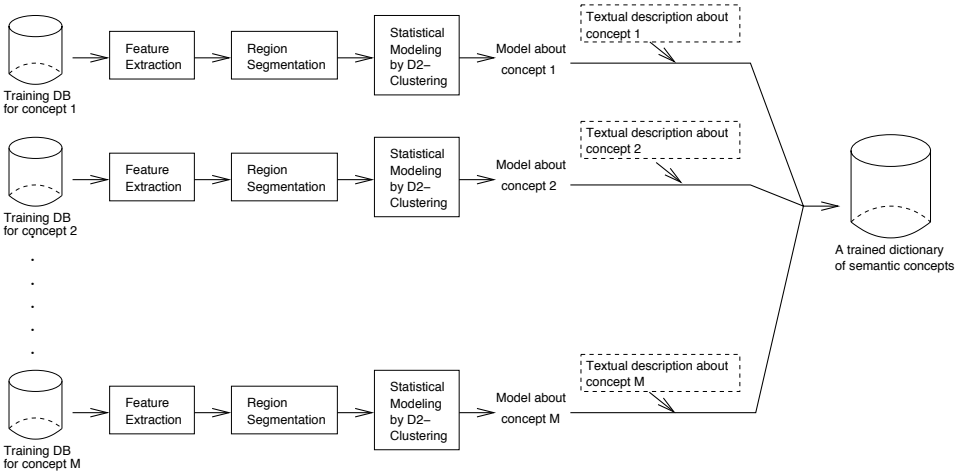


Figure 3: The training process of the ALIPR system.

refer to the collection of pixels mapped to the same cluster as a region. For each region, the average color (or texture) vector and the percentage of pixels it contains with respect to the whole image are computed. The color information is thus formulated as a discrete distribution $\{(v^{(1)}, p^{(1)}), (v^{(2)}, p^{(2)}), ..., (v^{(m)}, p^{(m)})\}$, where $v^{(j)}$ is the mean color vector, $p^{(j)}$ is the associated probability, and m is the number of regions. Similarly, the texture information is cast into a discrete distribution. This feature extraction process is similar to that of the SIMPLIcity system [24].

In general, let us denote images in the database by $\{\beta_1, \beta_2, ..., \beta_n\}$. Suppose every image is mathematically an array of discrete distributions, $\beta_i = (\beta_{i,1}, \beta_{i,2}, ..., \beta_{i,d})$. Denote the space of $\beta_{i,l}$ by Ω_l, $\beta_{i,l} \in \Omega_l$, $l = 1, 2, ..., d$. Then the space of β_i is the Cartesian product space

$$\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_d \ .$$

The dimension d of Ω, i.e., the number of distributions contained in β_i, is referred to as the *super-dimension* to distinguish from the dimensions of vector spaces on which these distributions are defined. For a fixed super-dimension j, the distributions $\beta_{i,j}$, $i = 1, ..., n$, are defined on the same vector space, $\mathcal{R}^{d_j}$, where d_j is the dimension of the jth sample space. Denote distribution $\beta_{i,j}$ by

$$\beta_{i,j} = \{(v_{i,j}^{(1)}, p_{i,j}^{(1)}), (v_{i,j}^{(2)}, p_{i,j}^{(2)}), ..., (v_{i,j}^{(m_{i,j})}, p_{i,j}^{(m_{i,j})})\} \ , \qquad (1)$$

where $v_{i,j}^{(k)} \in \mathcal{R}^{d_j}$, $k = 1, ..., m_{i,j}$, are vectors on which the distribution $\beta_{i,j}$ takes positive probability $p_{i,j}^{(k)}$. The cardinality of the support set for $\beta_{i,j}$ is $m_{i,j}$ which varies with both the image and the super-dimension.

To further clarify the notation, consider the following example. Suppose images are segmented into regions by clustering 3-dimensional color features and 3-dimensional texture features respectively. Suppose a region formed by segmentation with either type of features is characterized by the corresponding mean feature vector. For brevity, suppose the regions have equal weights. Since two sets of regions are obtained for each image, the super-dimension is $d = 2$. Let the first super-dimension correspond to color regions and the second to texture regions. Suppose an image i has 4 color regions and 5 texture regions. Then

$$\beta_{i,1} = \{(v_{i,1}^{(1)}, \frac{1}{4}), (v_{i,1}^{(2)}, \frac{1}{4}), ..., (v_{i,1}^{(4)}, \frac{1}{4})\}, \ v_{i,1}^{(k)} \in \mathcal{R}^3;$$

$$\beta_{i,2} = \{(v_{i,2}^{(1)}, \frac{1}{5}), (v_{i,2}^{(2)}, \frac{1}{5}), ..., (v_{i,2}^{(5)}, \frac{1}{5})\}, \ v_{i,2}^{(k)} \in \mathcal{R}^3.$$

A different image i' may have 6 color regions and 3 texture regions. In contrast to image i, for which $m_{i,1} = 4$ and $m_{i,2} = 5$, we now have $m_{i',1} = 6$ and $m_{i',2} = 3$. However, the sample space where $v_{i,1}^{(k)}$ and $v_{i',1}^{(k')}$ (or $v_{i,2}^{(k)}$ vs. $v_{i',2}^{(k')}$) reside is the same, specifically, $\mathcal{R}^3$.

Existing methods of multivariate statistical modeling are not applicable to build models on Ω because Ω is not a Euclidean space. Lacking algebraic properties, we have to rely solely on a distance defined in Ω. Consequently, we adopt a prototype modeling approach.

2.3 Mallows Distance between Distributions

To compute the distance $D(\gamma_1, \gamma_2)$ between two distributions γ_1 and γ_2, we use the Mallows distance [14, 12] introduced in 1972. Suppose random variable $X \in \mathcal{R}^k$ follow the distribution γ_1 and $Y \in \mathcal{R}^k$ follow γ_2. Let $\Upsilon(\gamma_1, \gamma_2)$ be the set of joint distributions over X and Y with marginal distributions of X and Y constrained to γ_1 and γ_2 respectively. Specifically, if $\zeta \in \Upsilon(\gamma_1, \gamma_2)$, then ζ has sample space $\mathcal{R}^k \times \mathcal{R}^k$ and its marginals $\zeta_X = \gamma_1$ and $\zeta_Y = \gamma_2$. The Mallows distance is defined as the minimum expected distance between X and Y optimized over all joint distributions $\zeta \in \Upsilon(\gamma_1, \gamma_2)$:

$$D(\gamma_1, \gamma_2) \triangleq \min_{\zeta \in \Upsilon(\gamma_1, \gamma_2)} (E \parallel X - Y \parallel^p)^{1/p} \ , \qquad (2)$$

where $\| \cdot \|$ denotes the L_p distance between two vectors. In our discussion, we use the L_2 distance, i.e., $p = 2$. The Mallows distance is proven to be a true metric [3].

For discrete distributions, the optimization involved in computing the Mallows distance can be solved by linear programming. Let the two discrete distributions be

$$\gamma_i = \{(z_i^{(1)}, q_i^{(1)}), (z_i^{(2)}, q_i^{(2)}), ..., (z_i^{(m_i)}, q_i^{(m_i)})\}, i = 1, 2 \, .$$

Then Equation (2) is equivalent to the following optimiza-

tion problem:

$$D^2(\gamma_1,\gamma_2) = \min_{\{w_{i,j}\}} \sum_{i=1}^{m_1}\sum_{j=1}^{m_2} w_{i,j} \parallel z_1^{(i)} - z_2^{(j)} \parallel^2 \quad (3)$$

subject to $\sum_{j=1}^{m_2} w_{i,j} = q_1^{(i)}$, $i = 1, ..., m_1$, $\sum_{i=1}^{m_1} w_{i,j} = q_2^{(j)}$, $j = 1, ..., m_2$, $w_{i,j} \geq 0$, $i = 1, ..., m_1$, $j = 1, ..., m_2$.

The above optimization problem suggests that the squared Mallows distance is a weighted sum of pairwise squared L_2 distances between any support vector of γ_1 and any of γ_2. With an objective to minimize the aggregated distance, the optimization is over the matching weights between support vectors in the two distributions. The weights $w_{i,j}$ are restricted to be nonnegative and the weights emitting from any vector $z_i^{(j)}$ sum up to its probability $q_i^{(j)}$. Thus $q_i^{(j)}$ sets the amount of influence from $z_i^{(j)}$ on the overall distribution distance.

2.4 Discrete Distribution (D2-) Clustering

Since elements in Ω each contain multiple discrete distributions, we measure their distances by the sum of squared Mallows distances between individual distributions. Denote the distance by $\tilde{D}(\beta_i, \beta_j)$, $\beta_i, \beta_j \in \Omega$, then

$$\tilde{D}(\beta_i, \beta_j) \triangleq \sum_{l=1}^{d} D^2(\beta_{i,l}, \beta_{j,l}) \ .$$

Recall that d is the super-dimension of Ω.

To determine a set of prototypes $A = \{\alpha_i : \alpha_i \in \Omega, i = 1, ..., \bar{m}\}$ for an image set $B = \{\beta_i : \beta_i \in \Omega, i = 1, ..., n\}$, we propose the following optimization criterion:

$$L(B, A^*) = \min_A \sum_{i=1}^{n} \min_{j=1,...,\bar{m}} \tilde{D}(\beta_i, \alpha_j) \ . \quad (4)$$

The objective function (4) entails that the optimal set of prototypes, A^*, should minimize the sum of distances between images and their closest prototypes. This is a natural criterion to employ for clustering and is in the same spirit as the optimization criterion used by k-means. However, as Ω is more complicated than the Euclidean space and the Mallows distance itself requires optimization to compute, the optimization problem of (4) is substantially more difficult than that faced by k-means.

For the convenience of discussion, we introduce a prototype assignment function $c(i) \in \{1, 2, ..., \bar{m}\}$, for $i = 1, ..., n$. Let $L(B, A, c) = \sum_{i=1}^{n} \tilde{D}(\beta_i, \alpha_{c(i)})$. With A fixed, $L(B, A, c)$ is minimized by $c(i) = \arg\min_{j=1,...,\bar{m}} \tilde{D}(\beta_i, \alpha_j)$. Hence, $L(B, A^*) = \min_A \min_c L(B, A, c)$ according to (4). The optimization problem of (4) is thus equivalent to the following:

$$L(B, A^*, c^*) = \min_A \min_c \sum_{i=1}^{n} \tilde{D}(\beta_i, \alpha_{c(i)}) \quad (5)$$

To minimize $L(B, A, c)$, we iterate the optimization of c given A and the optimization of A given c as follows. We assume that A and c are initialized. The initialization will be discussed later. From clustering perspective, the partition of images to the prototypes and optimization of the prototypes are alternated.

1. For every image i, set $c(i) = \arg\min_{j=1,...,\bar{m}} \tilde{D}(\beta_i, \alpha_j)$.

2. Let $\mathcal{C}_j = \{i : c(i) = j\}$, $j = 1, ..., \bar{m}$. That is, $\mathcal{C}_j$ contains indices of images assigned to prototype j. For each prototype j, let $\alpha_j = \arg\min_{\alpha\in\Omega} \sum_{i\in\mathcal{C}_j} \tilde{D}(\beta_i, \alpha)$.

The update of $c(i)$ in Step 1 can be obtained by exhaustive search. The update of α_j cannot be achieved analytically and is the core of the algorithm. Note that

$$\begin{aligned} \alpha_j &= \arg\min_{\alpha\in\Omega} \sum_{i\in\mathcal{C}_j} \tilde{D}(\beta_i, \alpha) \\ &= \arg\min_{\alpha\in\Omega} \sum_{i\in\mathcal{C}_j} \sum_{l=1}^{d} D^2(\beta_{i,l}, \alpha_{\cdot,l}) \\ &= \sum_{l=1}^{d} \arg\min_{\alpha_{\cdot,l}\in\Omega_l} \sum_{i\in\mathcal{C}_j} D^2(\beta_{i,l}, \alpha_{\cdot,l}) \quad (6) \end{aligned}$$

Equation (6) indicates that each super-dimension $\alpha_{j,l}$ in α_j can be optimized separately. Due to the lack of space, we omit the derivation of the optimization algorithm that solves (6). We now summarize the D2-clustering algorithm, assuming the prototypes are initialized.

1. For every image i, set $c(i) = \arg\min_{j=1,...,\bar{m}} \tilde{D}(\beta_i, \alpha_j)$.

2. Let $\mathcal{C}_\eta = \{i : c(i) = \eta\}$, $\eta = 1, ..., \bar{m}$. Update each $\alpha_{\eta,l}$, $\eta = 1, ..., \bar{m}$, $l = 1, ..., d$, individually by the following steps. Denote

$$\alpha_{\eta,l} = \{(z_{\eta,l}^{(1)}, q_{\eta,l}^{(1)}), (z_{\eta,l}^{(2)}, q_{\eta,l}^{(2)}), ..., (z_{\eta,l}^{(m'_{\eta,l})}, q_{\eta,l}^{(m'_{\eta,l})})\} \ .$$

   (a) Fix $z_{\eta,l}^{(k)}$, $k = 1, ..., m'_{\eta,l}$. Update $q_{\eta,l}^{(k)}$, $w_{k,j}^{(i)}$, $i \in \mathcal{C}_\eta$, $k = 1, ..., m'_{\eta,l}$, $j = 1, ..., m_{i,l}$ by solving the linear programming problem:

$$\min_{q_{\eta,l}^{(k)}} \sum_{i\in\mathcal{C}_\eta} \min_{w_{k,j}^{(i)}} \sum_{k=1}^{m'_{\eta,l}} \sum_{j=1}^{m_{i,l}} w_{k,j}^{(i)} \parallel z_{\eta,l}^{(k)} - v_{i,l}^{(j)} \parallel^2 ,$$

   subject to $\sum_{k=1}^{m'_{\eta,l}} q_{\eta,l}^{(k)} = 1$; $q_{\eta,l}^{(k)} \geq 0$, $k = 1, ..., m'_{\eta,l}$; $\sum_{j=1}^{m_{i,l}} w_{k,j}^{(i)} = q_{\eta,l}^{(k)}$, $i \in \mathcal{C}_\eta$, $k = 1, ..., m'_{\eta,l}$; $\sum_{k=1}^{m'_{\eta,l}} w_{k,j}^{(i)} = p_{i,l}^{(j)}$, $i \in \mathcal{C}_\eta$, $j = 1, ..., m_{i,l}$; $w_{k,j}^{(i)} \geq 0$, $i \in \mathcal{C}_\eta$, $k = 1, ..., m'_{\eta,l}$, $j = 1, ..., m_{i,l}$.

   (b) Fix $q_{\eta,l}^{(k)}$, $w_{k,j}^{(i)}$, $i \in \mathcal{C}_\eta$, $1 \leq k \leq m'_{\eta,l}$, $1 \leq j \leq m_{i,l}$. Update $z_{\eta,l}^{(k)}$, $k = 1, ..., m'_{\eta,l}$ by

$$z_{\eta,l}^{(k)} = \frac{\sum_{i\in\mathcal{C}_\eta} \sum_{j=1}^{m_{i,l}} w_{k,j}^{(i)} v_{i,l}^{(j)}}{\sum_{i\in\mathcal{C}_\eta} \sum_{j=1}^{m_{i,l}} w_{k,j}^{(i)}} \ .$$

   (c) Compute

$$\sum_{k=1}^{m'_{\eta,l}} \sum_{j=1}^{m_{i,l}} w_{k,j}^{(i)} \parallel z_{\eta,l}^{(k)} - v_{i,l}^{(j)} \parallel^2 \ .$$

   If the rate of decrease from the previous iteration is below a threshold, go to Step 3; otherwise, go to Step 2a.

3. Compute $L(B, A, c)$. If the rate of decrease from the previous iteration is below a threshold, stop; otherwise, go back to Step 1.

The initial prototypes are generated by tree structured clustering. At each iteration, the leaf node with the maximum sum of within cluster distances is chosen to split. The prototype of the leaf node obtained from previous computation serves as one seed and a randomly chosen image in the leaf node serves as the second. Starting with the seeds, the two prototypes are optimized by the D2-clustering algorithm, yielding two new leaf nodes at the end.

The number of prototypes $\bar{m}$ is determined adaptively for different concepts of images. Specifically, the value of $\bar{m}$ is increased gradually until the loss function is below a given threshold or $\bar{m}$ reaches an upper limit. In our experiment, the upper limit is set to 20, which ensures that on average, every prototype is associated with 4 training images. Concepts with higher diversity among images tend to require more prototypes. The histogram for the number of prototypes in each concept, shown in Figure 4(a), demonstrates the wide variation in the level of image diversity within one concept.

2.5 Modeling

With the prototypes determined, we employ a mixture modeling approach to construct a probability measure on Ω. Every prototype is regarded as the centroid of a mixture component. Here, the term *component* refers to a cluster formed by partitioning a set of images. Each element in the component is the signature of an image. For an image generated by a component, the further it is from the corresponding prototype, the lower the likelihood of the image under this component.

Figure 4 (b) shows the histogram of distances between images and their closest prototypes in one experiment. The curves overlaid on it are the probability density functions (pdf) of two fitted Gamma distributions. The pdf function is scaled so that it is at the same scale as the histogram. Denote a Gamma distribution by $(\gamma : b, s)$, where b is the scale parameter and s is the shape parameter. The pdf of $(\gamma : b, s)$ is [7]:

$$f(u) = \frac{(\frac{u}{b})^{s-1} e^{-u/b}}{b\Gamma(s)}, \quad u \geq 0$$

where $\Gamma(\cdot)$ is the Gamma function [7]. Consider multivariate random vector $X = (X_1, X_2, ..., X_k) \in \mathcal{R}^k$ that follows a normal distribution with mean $\mu = (\mu_1, ..., \mu_k)$ and a covariance matrix $\Sigma = \sigma^2 I$, where I is the identity matrix. Then the squared Euclidean distance between X and the mean μ, $\| X - \mu \|^2$, follows a Gamma distribution $(\gamma : \frac{k}{2}, 2\sigma^2)$. Based on this fact, we assume that the neighborhood around each prototype in Ω can be locally approximated by $\mathcal{R}^k$, where $k = 2s$ and $\sigma^2 = b/2$. The parameters s and b are estimated from the distances between images and their closest prototypes. In the local conjectural space $\mathcal{R}^k$, images belonging to a given prototype are assumed to be generated by a multivariate normal distribution with a mean vector being the map of the prototype in $\mathcal{R}^k$. The pdf for a multivariate normal distribution $N(\mu, \sigma^2 I)$ is:

$$\varphi(x) = (\frac{1}{\sqrt{2\pi\sigma^2}})^k e^{-\frac{\|x-\mu\|^2}{2\sigma^2}}.$$

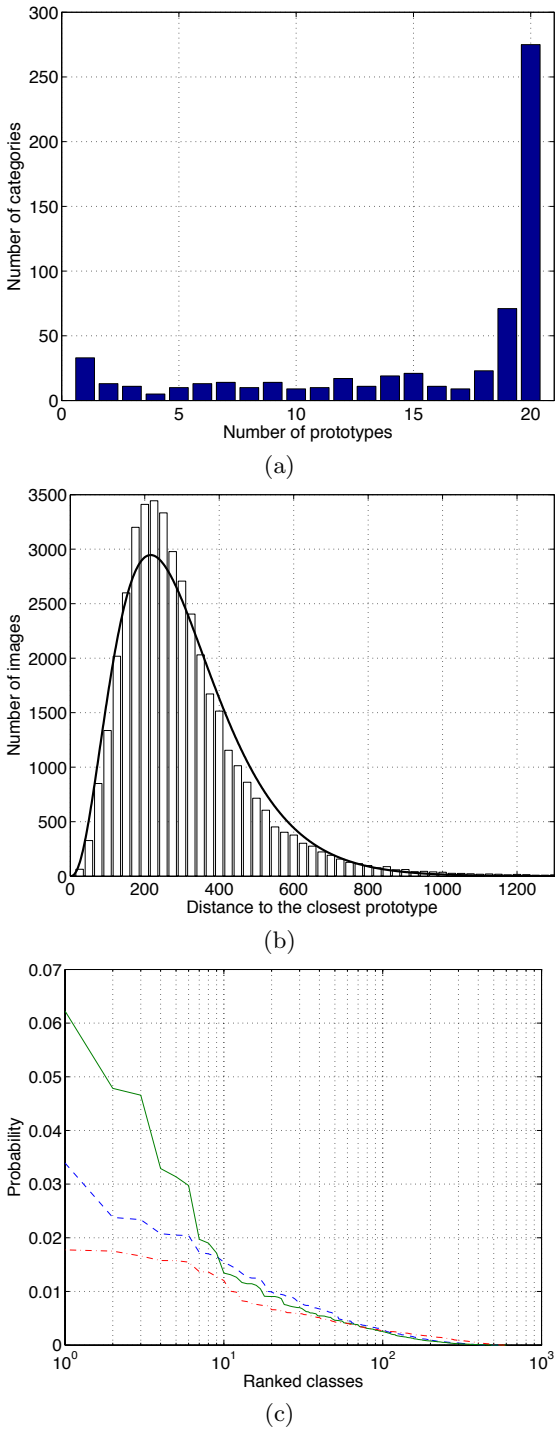


Figure 4: Statistical modeling results. (a) Histogram for the number of prototypes in each class. (b) Fitting a Gamma distribution to the distance between an image and its closest prototype: the histogram of the distances is shown with the correspondingly scaled probability density function of an estimated Gamma distribution. (c) The ranked concept posterior probabilities for three example images.

Formulating the component distribution back in Ω, we note that $\| x - \mu \|^2$ is correspondingly the $\tilde{D}$ distance between an image and its prototype. Let the prototype be α and the image be β. Also express k and σ^2 in terms of the Gamma distribution parameters b and s. The component distribution around α is:

$$g(\beta) = (\frac{1}{\sqrt{\pi b}})^{2s} e^{-\frac{\tilde{D}(\beta,\alpha)}{b}} .$$

For an M component mixture model in Ω with prototypes $\{\alpha_1, \alpha_2, ..., \alpha_M\}$, let the prior probabilities for the components be ω_η, $\eta = 1, ..., M$, $\sum_{\eta=1}^{M} \omega_\eta = 1$. The overall model for Ω is then:

$$\phi(\beta) = \sum_{\eta=1}^{M} \omega_\eta (\frac{1}{\sqrt{\pi b}})^{2s} e^{-\frac{\tilde{D}(\beta,\alpha_\eta)}{b}} . \quad (7)$$

The prior probabilities ω_η can be estimated by the percentage of images partitioned into prototype α_η, i.e., for which α_η is their closest prototype.

Next, we discuss the estimation of b and s. Let the set of distances be $\{u_1, u_2, ..., u_n\}$. Denote the mean $\bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i$. The maximum likelihood (ML) estimators $\hat{b}$ and $\hat{s}$ are solutions of the equations:

$$\begin{cases} \log \hat{s} - \psi(\hat{s}) = \log \left[\bar{u} / (\prod_{i=1}^{n} u_i)^{1/n} \right] \\ \hat{b} = \bar{u} / \hat{s} \end{cases}$$

where $\psi(\cdot)$ is the di-gamma function [7]:

$$\psi(s) = \frac{d \log \Gamma(s)}{ds}, \ s > 0 .$$

The above set of equations are solved by numerical methods. As $2s = k$, the dimension of the conjectural space, needs to be an integer, we adjust the ML estimation $\hat{s}$ to $s^* = \lfloor 2\hat{s} + 0.5 \rfloor / 2$, where $\lfloor \cdot \rfloor$ is the floor function. The ML estimator for b with s^* given is $b^* = \bar{u}/s^*$. Based on the training data we used, the histogram of the distances and the fitted Gamma distribution are shown in Figure 4(b). The Gamma distribution estimated is $(\gamma : 3.5, 86.34)$, indicating that the conjectural space is of dimension 7.

In summary, the modeling process comprises the following steps: (1) For each image category, find a set of prototypes, partition images into these prototypes, and compute the distance between each image and the prototype it belongs to. (2) Collect the distances in all image categories and estimate a Gamma distribution parameterized by s^* and b^*. For robustness, if a prototype contains too few images, for instance, less than 3 images in our system, distances computed from these images are not used. (3) Construct a mixture model for each image category using Equation (7).

3. THE ANNOTATION METHOD

Let the set of distinct annotation words for the M concepts be $\mathcal{W} = \{w_1, w_2, ..., w_K\}$. In the experiment with the Corel database as training data, $K = 332$. Denote the set of concepts that contain word w_i in their annotations by $\mathcal{C}(w_i)$. For instance, the word 'castle' is among the description of concept 160, 404, and 405. Then $\mathcal{C}(castle) = \{160, 404, 405\}$.

To annotate an image, its signature s is extracted first. We then compute the probability for the image being in each concept m:

$$p_m(s) = \frac{\rho_m f(s \mid \mathcal{M}_m)}{\sum_{l=1}^{M} \rho_l f(s \mid \mathcal{M}_l)} , \quad m = 1, 2, ..., M ,$$

where ρ_m are the prior probabilities for the concepts and are set uniform. The probability for each word w_i, $i = 1, ..., K$, to be associated with the image is

$$q(s, w_i) = \sum_{m: m \in \mathcal{C}(w_i)} p_m(s) .$$

We then sort $\{q(s, w_1), q(s, w_2), ..., q(s, w_K)\}$ in descending order and select top ranked words. Figure 4(c) shows the sorted posterior probabilities of the 599 semantic concepts given each of three example images. The posterior probability decreases slowly across the concepts, suggesting that the most likely concept for each image is not strongly favored over the others. It is therefore important to quantify the posterior probabilities rather than simply classifying an image into one concept.

To further improve computational efficiency for real-time annotation, the Mallows distance between an image to be annotated and each prototype is first approximated by the IRM distance [24], which is much faster to compute. Then, for a certain number of prototypes that are closest to the image according to IRM, the Mallows distances are computed precisely. Thus, the IRM distance is used as a screening mechanism rather than a simple replacement of the more complex distance. Invoking this speed-up method causes negligible change on annotation results.

4. EXPERIMENTAL RESULTS

The training process takes an average of 109 seconds CPU time, with a standard deviation of 145 seconds, for each category of 80 training images on a 2.4 GHz AMD processor.

Annotation results for more than 54,700 images created by users of flickr.com are viewable at the Website: `alipr.com`. This site also hosts the ALIPR demonstration system that performs real-time annotation for any online image specified by its URL. Annotation words for 12 images downloaded from the Internet are obtained by the online system and are displayed in Figure 5. Six of the images are photographs and the others are digitized impressionism paintings. For these example images, it takes a 3.0 GHz Intel processor an average of 1.4 seconds to convert each from the JPEG to raw format, abstract the image into a signature, and find the annotation words.

It is not easy to find completely failed examples. However, we picked some unsuccessful examples, as shown in Figure 6. In general, the computer does poorly (a) when the way an object is taken in the picture is very different from those in the training, (b) when the picture is fuzzy or of extremely low resolution or low contrast, (c) if the object is shown partially, (d) if the white balance is significantly off, and (e) if the object or the concept has not been learned.

To numerically assess the annotation system, we manually examined the annotation results for 5,411 digital photos deposited by random users at flickr.com. Due to limited space, we will focus on reporting the results on these images.

Although several prototype annotation systems have been developed previously, a quantitative study on how accurate a computer can annotate images in the real-world has never been conducted. The existing assessment of annotation

people, man-made, car,
landscape, bus, boat,
sport, royal guard, ocean

flower, plant, rose,
cactus, flora, grass,
landscape, water, perennial

indoor, food, dessert,
man-made, bath, kitchen,
texture, landscape, bead

landscape, building, historical,
mountain, man-made, indoor,
people, lake, animal

grass, people, animal,
horse, rural, dog,
landscape, tribal, plant

grass, animal, wild life,
sport, people, rock,
tree, horse, polo

texture, indoor, food,
natural, people, animal,
landscape, rock, man-made

landscape, indoor, color,
sky, sunset, sun,
bath, kitchen, mountain

grass, landscape, house,
rural, horse, animal,
people, plant, flower

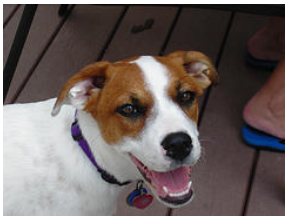
people, landscape, animal,
cloth, female, painting,
face, male, man-made

man-made, indoor, painting,
people, food, fruit,
mural, old, poster

grass, landscape, tree,
lake, autumn, people,
rural, texture, natural

Figure 5: Automatic annotation for photographs and paintings. The words are ordered according to estimated likelihoods. The photographic images were obtained from flickr.com. The paintings were obtained from online Websites.

accuracy is limited in two ways. First, because the computation of accuracy requires human judgment on the appropriateness of each annotation word for each image, the enormous amount of manual work has prevented researchers from calculating accuracy directly and precisely. Lower bounds [13] and various heuristics [1] are used as substitutes. Second, test images and training images are from the same benchmark database. Because many images in the database are highly similar to each other, it is unclear whether the models established are equally effective for general images.



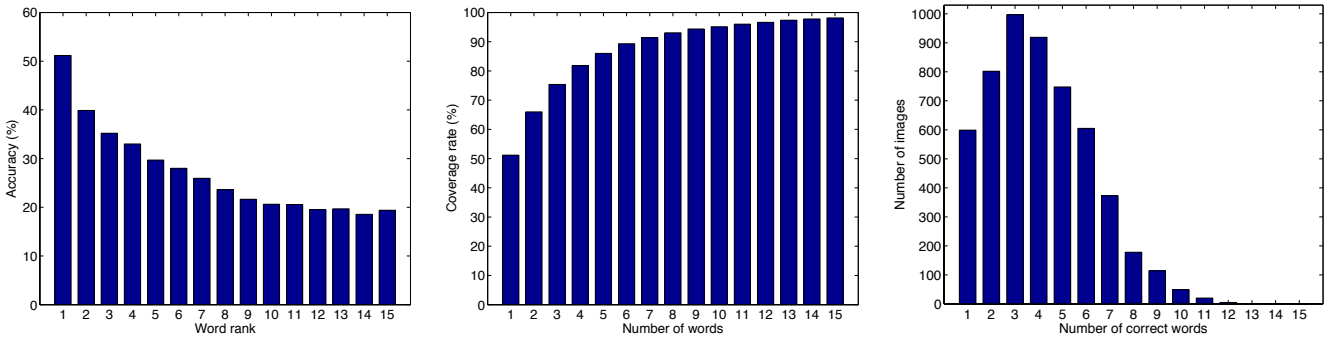
(a) building, people, water, modern, city, work, historical, cloth, horse
User annotation: photo, unfound, molly, dog, animal

(b) texture, indoor, food, natural, cuisine, man-made, fruit, vegetable, dessert
User annotation: phonecamera, car

(c) texture, natural, flower, sea, micro_image, fruit food, vegetable, indoor
User annotation: me, selfportrait, orange, mirror

(d) texture, painting, flower, landscape, rural, pastoral, plant, grass, natural
User annotation: 911, records, money, green, n2o

Figure 6: Unsuccessful cases of automatic annotation. The words are ordered according to estimated likelihoods. The photographic images were obtained from flickr.com. Underlined words are considered reasonable annotation words. Suspected problems: (a) object with an unusual background, (b) fuzzy shot, (c) partial object, wrong white balance, and (d) unlearned object or concept.



(a) (b) (c)

Figure 7: Annotation performance based on manual evaluation of $5,411$ flickr.com images. (a) Percentages of images correctly annotated by the nth word. (b) Percentages of images correctly annotated by at least one word among the top n words. (c) Histogram of the numbers of correct annotation words for each image among the top 15 words assigned to it.

Our evaluation experiments, designed in a realistic manner, will shed light on the level of intelligence a computer can achieve for describing images.

A Web-based evaluation system is developed to record human decision on the appropriateness of each annotation word provided by the system. Each image is shown together with 15 computer-assigned words in a browser. A trained person, who did not participate in the development of the training database or the system itself, examines every word against the image and checks a word if it is judged as correct. For words that are object names, they are considered correct if the corresponding objects appear in an image. For more abstract concepts, e.g., 'city' and 'sport', a word is correct if the image is relevant to the concept. For instance, 'sport' is appropriate for a picture showing a polo game or golf, but not for a picture of dogs. Manual assessment is collected for $5,411$ images at flickr.com. Optimism in performance evaluation is avoided by employing independently acquired training and testing images.

Annotation performance is reported from several aspects in Figure 7. Each image is assigned with 15 words listed in the descending order of the likelihood of being relevant. Figure 7(a) shows the accuracies, that is, the percentages of images correctly annotated by the nth annotation word, $n = 1, 2, ..., 15$. The first word achieves an accuracy of 51.17%. The accuracy decreases gradually with n except for minor fluctuation with the last three words. This reflects that the ranking of the words by the system is on average consistent with the true level of accuracy. Figure 7(b) shows the coverage rate versus the number of annotation words used. Here, coverage rate is defined as the percentage of images that are correctly annotated by at least one word among a given number of words. To achieve 80% coverage, we only need to use the top 4 annotation words. The top 7 and top 15 words achieve respectively a coverage rate of 91.37% and 98.13%. The histogram of the numbers of correct annotation words among the top 15 words is provided in Figure 7(c). On average, 4.1 words are correct for each image.

5. CONCLUSIONS AND FUTURE WORK

Images are a major media on the Internet. To ensure easy sharing of and effective searching over a huge and fast growing number of online images, real-time automatic annotation by words is an imperative but highly challenging task. We have developed and evaluated the ALIPR, Automatic Linguistic Indexing of Pictures - Real Time, system. Our work has shown that the computer can

annotate general photographs with substantial accuracy by learning from a large collection of example images. Novel statistical modeling and optimization methods have been developed for establishing probabilistic associations between images and words. By manually examining annotation results for over 5,400 real-world pictures, we have shown that more than 51% of the images are correctly annotated by their highest ranked words alone. When the top 15 words are used to describe each image, above 98% of the images are correctly annotated by some words.

There are many directions future work can take to improve the accuracy of the system. First, the incorporation of 3-D information in the learning process can potentially improve the models. This can be done through learning via stereo images or 3-D images. Shape information can be utilized to improve the modeling process. Second, better and larger amount of training images per semantic concept may produce more robust models. Contextual information may also help in the modeling and annotation process. Third, the applications of this method to various application domains including biomedicine can be interesting. Finally, the system can be integrated with other retrieval methods to improve the usability.

Acknowledgments: The research is supported in part by the US National Science Foundation under Grant Nos. 0219272 and 0202007. We thank Diane Flowers for providing manual evaluation on annotation results, Dhiraj Joshi for designing a Web-based evaluation system, David M. Pennock at Yahoo! for providing test images, and Hongyuan Zha for useful discussions. We would also like to acknowledge the comments and constructive suggestions from reviewers.

6. REFERENCES

[1] K. Barnard, P. Duygulu, N. de Freitas, D. A. Forsyth, D. M. Blei, and M. I. Jordan, "Matching words and pictures," *Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[2] D. Beymer and T. Poggio, "Image representations for visual learning," *Science*, vol. 272, pp. 1905–1909, 1996.

[3] P. J. Bickel and D. A. Freedman, "Some asymptotic theory for the bootstrap," *Annals of Statistics*, vol. 9, pp. 1196–1217, 1981.

[4] S.-F. Chang, W. Chen, and H. Sundaram, "Semantic visual templates: Linking visual features to semantics," In *Proc. Int. Conf. on Image Processing,* vol. 3, pp. 531–535, Chicago, IL, 1998.

[5] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval - Approaches and trends of the new age," In *Proc. Int. Workshop on Multimedia Information Retrieval*, pp. 253–262, 2005.

[6] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, 1992.

[7] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, 3rd ed., John Wiley & Sons, Inc., 2000.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed., Prentice Hall, 2002.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inferences, and Prediction*, Springer-Verlag, New York, 2001.

[10] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Mean version space: A new active learning method for content-based image retrieval," In *Proc. Multimedia Information Retrieval Workshop*, 2004.

[11] X. He, W.-Y. Ma, and H.-J. Zhang. "Learning an image manifold for retrieval," In *Proc. ACM Multimedia Conf.*, 2004.

[12] E. Levina and P. Bickel, "The earth mover's distance is the Mallows distance: Some insights from statistics," In *Proc. Int. Conf. on Computer Vision*, pp. 251–256, Vancouver, Canada, 2001.

[13] J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075–1088, 2003.

[14] C. L. Mallows, "A not on asymptotic joint normality," *Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 508–515,1972.

[15] F. Monay and D. Gatica-Perez, "On image auto-annotation with latent space models," In *Proc. ACM Multimedia Conf.*, 2003.

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

[17] A. Singhal, J. Luo, and W. Zhu, "Probabilistic spatial context models for scene content understanding," In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2003.

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

[19] J. R. Smith and S.-F. Chang, "VisualSEEk: A fully automated content-based image query system," In *Proc. ACM Multimedia Conf.*, 1996.

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

[21] C. Tomasi, "Past performance and future results," *Nature*, vol. 428, page 378, March 2004.

[22] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," In *Proc. ACM Multimedia Conf.*, pp. 107–118, 2001.

[23] N. Vasconcelos and A. Lippman, "A multiresolution manifold distance for invariant image similarity," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 127–142, 2005.

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

[25] J. Z. Wang and J. Li, "Learning-based linguistic indexing of pictures with 2-D MHMMs," In *Proc. ACM Multimedia Conf.*, pp. 436-445, Juan Les Pins, France, ACM, 2002.

[26] C. Zhang and T. Chen, "An active learning framework for content-based information retrieval," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 260–268, 2002.