Real-Time Computerized Annotation of Pictures

Jia Li, *Senior Member, IEEE,* and James Z. Wang, *Senior Member, IEEE*

***Abstract*— Developing effective methods for automated annotation of digital pictures continues to challenge computer scientists. 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 this work, the authors developed new optimization and estimation techniques to address two fundamental problems in machine learning. These new techniques serve as the basis for the Automatic Linguistic Indexing of Pictures - Real Time (ALIPR) system of fully automatic and high speed annotation for online pictures. In particular, the D2-clustering method, in the same spirit as k-means for vectors, is developed to group objects represented by bags of weighted vectors. Moreover, a generalized mixture modeling technique (kernel smoothing as a special case) for non-vector data is developed using the novel concept of Hypothetical Local Mapping (HLM). ALIPR has been tested by thousands of pictures from an Internet photo-sharing site, unrelated to the source of those pictures used in the training process. Its performance has also been studied at an online demonstration site where arbitrary users provide pictures of their choices and indicate the correctness of each annotation word. The experimental results show that a single computer processor can suggest annotation terms in real-time and with good accuracy.**

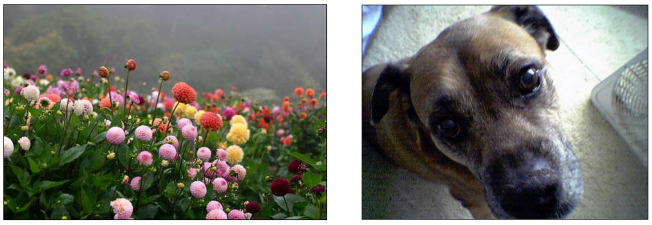
***Index Terms*— Image Annotation, Tagging, Statistical Learning, Modeling, Clustering**

I. 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 reports released in 2007, an Internet photo-sharing startup, flickr.com, has 40 million monthly visitors and hosts two billion photos, with new photos in the order of millions being added on a daily basis. More specialized online photo-sharing communities, such as photo.net and airliners.net, also have databases in the order of millions of images contributed entirely by the users.

A. 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 for unannotated image collections. The complex and fragmented nature of the networked communities makes fully computerized or computer-assisted annotation of images by words a crucial technology to ensure the "visibility" of images on the Internet.



(a) (b)

Fig. 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 photo-sharing communities can request that owners of digital images provide some descriptive words when depositing the images, such annotations tend to be highly subjective. For example in the pictures shown in Figure 1, the users on flickr.com annotated the first picture with the tags *dahlia*, *golden*, *gate*, *park*, *flower*, and *fog* and the second picture by *cameraphone*, *animal*, *dog*, and *tyson*. According to the photographer, the first picture was taken at the Golden Gate Park near San Francisco. This set of annotation words could be a problem because this picture may show up when other users search for images of gates. Similarly, 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 could assist those labeling as well as those searching images. Busy users can simply select relevant words and, optionally, type in other words. The system can also be used to check the user-supplied tags against the image content, by using a semantic network, to improve the accuracy of keyword-based searching. In computer security, such a system can assist trained personnel to filter unwanted materials. No real-world applications of automatic annotation or tagging of images with a large number of concepts exist largely because creating a competent system is extremely challenging.

B. 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 [27]. In 2000, Smeulders et al. published a comprehensive survey of the field [26]. Progresses made in the field after 2000 is documented in a recent survey article [8]. We review here some work closely related to ours. The references listed below are to be taken as

J. Li is with the Department of Statistics, The Pennsylvania State University, University Park, PA 16802. Email: jiali@stat.psu.edu.

J. Z. Wang is with the College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802. He is also affiliated with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213. Email: jwang@ist.psu.edu.

An on-line demonstration is provided at the URL: `http://alipr.com`. More information about the research: `http://riemann.ist.psu.edu`.

Manuscript accepted 19 Dec. 2007.

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 [3], [11], [12]. Generative modeling [2], [16], statistical boosting [28], visual templates [6], Support Vector Machines [30], multiple instance learning, active learning [34], [13], latent space models [20], spatial context models [25], feedback learning [24] and manifold learning [31], [14] have been applied to 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) [16], [33]. 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 (or class). 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. If the layer of word selection is omitted, ALIP essentially conducts multiple classification, where the classes are hundreds of semantic concepts.

Classifying images into a large number of categories has also been explored recently by Chen et al. [7] for the purpose of pure classification and Carneiro et al. [5] for annotation using multiple instance learning. Barnard et al. [2] 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 in [16], 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. Existing systems also lack performance evaluation in real-world deployment, leaving the practical potential of automatic annotation largely unaddressed. In fact, most systems have been tested using images in the same collection as the training images, resulting in bias in evaluation. In addition, because direct measurement of annotation accuracy involves labor intensive examination, substitutive quantities related to accuracy have often been used instead.

C. 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. Statistical 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 the computer needs no re-training.

The breakthrough in computational efficiency results from a fundamental change in the modeling approach. In ALIP [16], 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, and the profiling model specifies a probability law for distributions directly.

A real-time annotation demonstration system, *ALIPR* (*Automatic Linguistic Indexing of Pictures - Real Time*), is provided online at `http://alipr.com`. 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, the system identifies annotation words for each picture.

The contribution of our work is multifold:

- We have developed a real-time automatic image annotation system. To our knowledge, this work is the first to achieve real-time performance with a level of accuracy useful in certain real applications. It is also the first attempt to manually assess the large scale performance of an image annotation system. This system has been through rigorous evaluation, including extensive tests using Web images completely independent from the training images. The system performance has also been assessed based on the input of thousands of online users. Data from these experiments will establish benchmarks for related future technologies as well as for the mere interest of understanding the potential of artificial intelligence. Our research sheds light on the expectation of arbitrary real-world users, an area that has been nearly unexplored.
- We have developed new generally-applicable methods for clustering and mixture modeling, and we expect these methods to be useful for problems involving data other than images. First, we have designed a novel clustering algorithm for objects represented by discrete distributions, i.e., bags of weighted vectors. This new algorithm minimizes the total within cluster distance, a criterion used by the k-means algorithm. We call the algorithm *D2-clustering*, where D2 stands for discrete distribution. D2-clustering generalizes the k-means algorithm from the data form of vectors to sets of weighted vectors. Although under the same spirit as k-means, D2-clustering involves much more sophisticated optimization techniques. Second, we have constructed a new mixture modeling method, namely the hypothetical local mapping (HLM) method, to efficiently build a probability measure on the space of discrete distributions.

D. Outline of the Paper

The remainder of the paper is organized as follows: In Section II, we provide an outline of our approach and preliminaries. The D2-clustering method is described in Section III. The mixture modeling approach is presented in Sections IV. The assignment of annotation words and measures for improving computational efficiency are described in

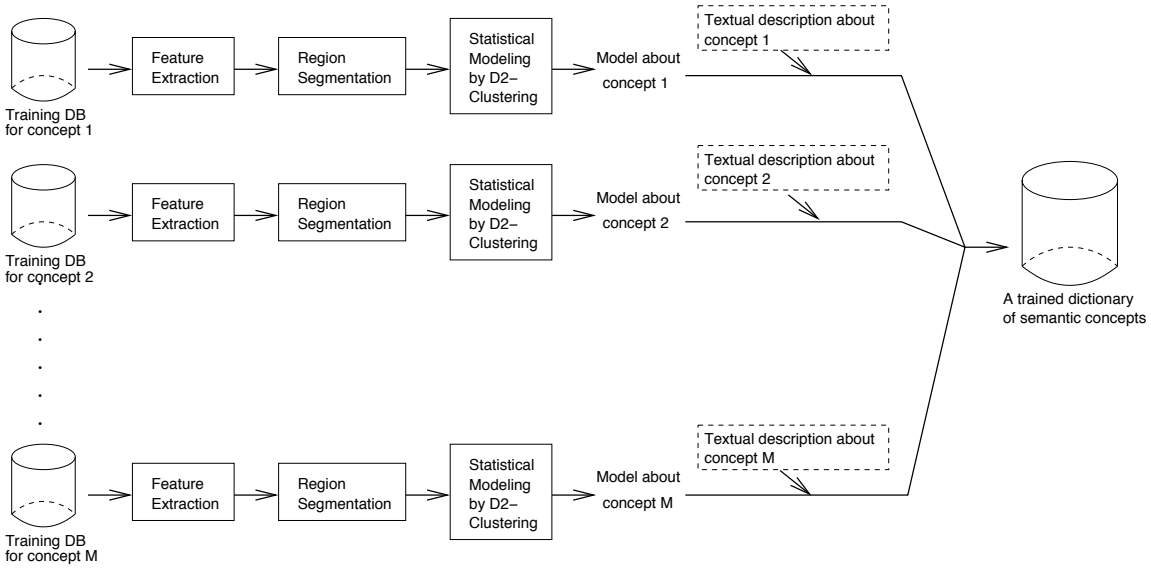


Fig. 2. The training process of the Automatic Linguistic Indexing of Pictures - Real Time (ALIPR) system.

Section V. The experimental results are provided in Section VI. We conclude and suggest future work in Section VII.

II. Preliminaries

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, the Corel database is used for training with $M = 599$. 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 β_i, $\beta_i \in \Omega$. 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: $\{\beta_i : g_i = m, 1 \leq i \leq N\}$. Denote the probability density function under model $\mathcal{M}_m$ by $\phi(s \mid \mathcal{M}_m)$, $s \in \Omega$.

Figure 2 illustrates this training process. The annotation process based upon the models will be described in Section V.

A. The Training Database

It is well known that applying learning results to unseen data can be significantly harder than applying to training data [29]. In our work, we used completely different databases for training the system and for testing the performance.

The Corel image database, containing close to $60,000$ general-purpose photographs, is used to learn the statistical relationships between images and words. This same database was also exploited in the development of SIMPLIcity [32] and ALIP [16]. A large portion of images in the database are scene photographs. The rest includes man-made objects with smooth background, fractals, texture patches, synthetic graphs, drawings, etc. 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.

We clarify that "general-purpose" photographs refer to pictures taken in daily life in contrast to special domain, such as medical or satellite, images. Although our annotation system is training based and is potentially applicable to images in other domains, different designs of image signatures are expected for optimal performance.

B. The Selection of Features and Modeling Methods

In order to achieve the goal of real-time computerized suggestions for picture tags, the combination of feature extraction and statistical matching with hundreds of trained models must be confined to about a second in execution time on a typical computer processor. This stringent speed requirement severely limits the choices of features and methods we could use in this work. As indicated in our recent survey article, many local and global visual feature extraction methods are available [8]. In general, however, there is a trade-off between the number of features we incorporate in the signature and the time it takes to extract the features and to match them against the models. For instance, the earlier ALIP system, which uses block-level wavelet-based descriptors and a spatial statistical modeling method, takes more than ten minutes on a single processor to suggest tags for each picture [16].

To reduce from an order of ten minutes to a second (which represents a three order of magnitude cutback), substantial reduction in both the feature complexity and modeling complexity must be accomplished while maintaining a reasonable level of accuracy for practical use in online tasks. The integration of region segmentation and extracting representative color and texture features of the segments is a suitable time-reduction strategy; however, sophisticated region segmentation methods themselves are often not in real time. Borrowing from the

experiences gained in large-scale visual similarity search, we use a fast image segmentation method based on wavelets and k-means clustering [32].

The low complexity of this segmentation method makes it an attractive option for processing large amounts of images. Unfortunately, this method is more suitable for recognizing scenes, and thus, we expect the method will be insufficient for recognizing individual objects, given the great variations a type of objects (e.g., dogs) can appear in pictures. Although object names are often assigned by the system, the selection is mostly based on statistical correlation with scenes. On the other hand, as pointed out by one reviewer, different levels of performance may be possible under a more controlled image set, such as various types of the same object or images of the same domain. We will explore this in the future.

After the region-based signatures are extracted from the pictures, we encounter the essential obstacle: the segmentation-based signatures are of arbitrary lengths across the picture collection, primarily because the number of regions used to represent a picture often depends on how complicated the composition of the picture is. No existing statistical tools can handle the modeling in this scenario. The key challenge to us, therefore, is to develop new statistical methods for feature modeling and model matching when the signatures are in the form of discrete distributions. The details on these are provided in the following sections.

C. Image Signature

To form the signature of an image, two types of features are extracted: color and texture. To extract the color part of the signature, the RGB color components of each pixel are converted to the LUV color components. The 3-D color vectors at all the pixels are clustered by k-means. The number of clusters in k-means is determined dynamically by thresholding the average within cluster distances. Arranging the cluster labels of the pixels into an image according to the pixel positions, we obtain a segmentation of the image. We refer to the collection of pixels mapped to the same cluster as a region. For each region, its average color vector and the percentage of pixels it contains with respect to the whole image are computed. The color signature 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.

We use wavelet coefficients in high frequency bands to form texture features. A Daubechies-4 wavelet transform [9] is applied to the L component (intensity) of each image. The transform decomposes an image into four frequency bands: LL, LH, HL, HH. The LH, HL, and HH band wavelet coefficients corresponding to the same spatial position in the image are grouped into one 3-D texture feature vector. If an image contains $n_r \times n_c$ pixels, the total number of texture feature vectors is $\frac{n_r}{2} \times \frac{n_c}{2}$ due to the subsampling of the wavelet transform. When forming the texture features, the absolute values of the wavelet coefficients are used. K-means clustering is applied to the texture feature vectors to extract the major modes of these vectors. Again, the number of clusters is decided adaptively by thresholding the average within cluster distances. Similarly as color, the texture signature is cast into a discrete distribution.

Although we only involve color and texture in the current image signature, other types of image features such as shape and salient points can also be formulated into discrete distributions, i.e., bags of weighted vectors. For instance, bags of SIFT features [17] are used to characterize and subsequently detect advertisement logos in video frames [1]. As expected, our current image signature are not sensitive to shape patterns. We choose to use color and texture features because they are relatively robust for digital photos generated by Internet users. Shape or salient point features may be more appealing for recognizing objects. However, these features are highly prone to corruption when the background is noisy, object viewing angle varies, or occlusion occurs, as is usually the case. Moreover, semantics of an image sometimes cannot be adequately expressed by a collection of object names. Deriving image signatures that are robust as well as strong for semantic recognition is itself a deep research problem which we would like to explore in the future.

In general, let us denote images in the database by $\{\beta_1, \beta_2, ..., \beta_N\}$. Suppose every image is represented by 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-D color features and 3-D 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-dimensionality is $d = 2$. Let the first super-dimension correspond to color based regions and the second to texture based 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 to be explained in Section III and IV.

D. 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 [19], [15] 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 $\zeta \in \Upsilon(\gamma_1, \gamma_2)$:

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

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 proved to be a true metric [4].

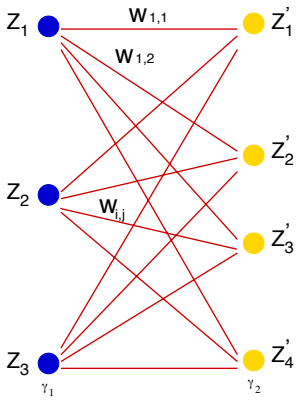


Fig. 3. Matching for computing the Mallows distance.

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 optimization problem:

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

subject to

$$\begin{aligned} \sum_{j=1}^{m_2} w_{i,j} &= q_1^{(i)}, \ i = 1, ..., m_1; \\ \sum_{i=1}^{m_1} w_{i,j} &= q_2^{(j)}, \ j = 1, ..., m_2; \\ w_{i,j} &\geq 0, \ i = 1, ..., m_1, j = 1, ..., m_2 . \end{aligned} \tag{4}$$

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. Hence, as shown in Figure 3, computing the Mallows distance is essentially optimizing matching weights between support vectors in the two distributions so that the aggregated distance is minimized. The matching 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.

The optimization problem involved in computing the Mallows distance is the same as that for solving the mass transportation problem [22]. A well-known image distance used in retrieval, namely the Earth Mover's Distance (EMD) [23] is closely related to the Mallows distance. In fact, as discussed in [15], EMD is equivalent to the Mallows distance when the same total mass is assigned to both distributions.

III. 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

$$A = \{\alpha_\eta : \alpha_\eta \in \Omega, \eta = 1, ..., \bar{m}\}$$

for an image set

$$B = \{\beta_i : \beta_i \in \Omega, i = 1, ..., n\},$$

we propose the following optimization criterion:

$$L(B, A^*) = \min_A \sum_{i=1}^{n} \min_{\eta = 1, ..., \bar{m}} \tilde{D}(\beta_i, \alpha_\eta) . \tag{5}$$

The objective function (5) 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 (5) 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 $c(i) = \text{argmin}_{\eta=1,...,\bar{m}} \tilde{D}(\beta_i, \alpha_\eta)$. Hence, $L(B, A^*) = \min_A \min_c L(B, A, c)$ according to (5). The optimization problem of (5) is thus equivalent to the following:

$$L(B, A^*, c^*) = \min_A \min_c \sum_{i=1}^{n} \tilde{D}(\beta_i, \alpha_{c(i)}) . \tag{6}$$

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 $c(i) = \text{argmin}_{\eta=1,...,\bar{m}} \tilde{D}(\beta_i, \alpha_\eta)$.
2) Let $\mathcal{C}_\eta = \{i : c(i) = \eta\}$, $\eta = 1, ..., \bar{m}$. That is, $\mathcal{C}_\eta$ contains indices of images assigned to prototype η. For each prototype η, let $\alpha_\eta = \text{argmin}_{\alpha \in \Omega} \sum_{i \in \mathcal{C}_\eta} \tilde{D}(\beta_i, \alpha)$.

The update of $c(i)$ in Step 1 can be obtained by exhaustive search. The update of α_η cannot be achieved analytically and is the core of the algorithm. Use the notation $\alpha = (\alpha_{\cdot,1}, \alpha_{\cdot,2}, ..., \alpha_{\cdot,d})$. Note that

$$\begin{aligned}\alpha_\eta &= \operatorname*{argmin}_{\alpha\in\Omega} \sum_{i\in\mathcal{C}_\eta} \tilde{D}(\beta_i, \alpha) = \operatorname*{argmin}_{\alpha\in\Omega} \sum_{i\in\mathcal{C}_\eta} \sum_{l=1}^{d} D^2(\beta_{i,l}, \alpha_{\cdot,l}) \\ &= \sum_{l=1}^{d} \operatorname*{argmin}_{\alpha_{\cdot,l}\in\Omega_l} \sum_{i\in\mathcal{C}_\eta} D^2(\beta_{i,l}, \alpha_{\cdot,l}) \qquad (7)\end{aligned}$$

Equation (7) indicates that each super-dimension $\alpha_{\eta,l}$ in α_η can be optimized separately. For brevity of notation and without loss of generality, let us consider the optimization of $\alpha_{1,1}$. Also assume that $\mathcal{C}_1 = \{1, 2, ..., n'\}$. Let

$$\alpha_{\cdot,1} = \{(z^{(1)}, q^{(1)}), (z^{(2)}, q^{(2)}), ..., (z^{(m)}, q^{(m)})\} ,$$

where $\sum_{k=1}^{m} q^{(m)} = 1$, $z^{(k)} \in \mathcal{R}^{d_1}$. The number of vectors, m, can be preselected. If $\alpha_{\cdot,1}$ contains a smaller number of vectors than m, it can be considered as a special case with some $q^{(k)}$'s being zero. On the other hand, a large m requires more computation. The goal is to optimize over $z^{(k)}$ and $q^{(k)}$, $k = 1, ..., m$, so that $\sum_{i=1}^{n'} D^2(\beta_{i,1}, \alpha_{\cdot,1})$ is minimized. Recall the expansion of $\beta_{i,j}$ in (1). Applying the definition of the Mallows distance, we have

$$\begin{aligned}&\min_{\alpha_{\cdot,1}\in\Omega_1} \sum_{i=1}^{n'} D^2(\beta_{i,1}, \alpha_{\cdot,1}) \\ =\ &\min_{z^{(k)}, q^{(k)}} \sum_{i=1}^{n'} \min_{w_{k,j}^{(i)}} \sum_{k=1}^{m} \sum_{j=1}^{m_{i,1}} w_{k,j}^{(i)} \parallel z^{(k)} - v_{i,1}^{(j)} \parallel^2 . \qquad (8)\end{aligned}$$

The optimization is over $z^{(k)}$, $q^{(k)}$, $k = 1, ..., m$, and $w_{k,j}^{(i)}$, $i = 1, ..., n'$, $k = 1, ..., m$, $j = 1, ..., m_{i,1}$. Probabilities $q^{(k)}$'s are not explicitly in the objective function, but they affect the optimization by posing as constraints. The constraints for the optimization are:

$$\sum_{k=1}^{m} q^{(k)} = 1$$

$$q^{(k)} \geq 0 , \text{ for any } k = 1, ..., m$$

$$\sum_{j=1}^{m_{i,1}} w_{k,j}^{(i)} = q^{(k)} , \text{ for any } i = 1, ..., n', k = 1, ..., m$$

$$\sum_{k=1}^{m} w_{k,j}^{(i)} = p_{i,1}^{(j)} , \text{ for any } i = 1, ..., n', j = 1, ..., m_{i,1}$$

$$w_{k,j}^{(i)} \geq 0 , \text{ for any } i = 1, ..., n', k = 1, ..., m, j = 1, ..., m_{i,1} .$$

A key observation for solving the above optimization is that with fixed $z^{(k)}$, $k = 1, ..., m$, the objective function over $q^{(k)}$'s and $w_{k,j}^{(i)}$'s is linear and all the constraints are linear. Hence, with $z^{(k)}$'s fixed, $q^{(k)}$, $w_{k,j}^{(i)}$ can be solved by linear programming. It is worthy to note the difference between this linear optimization and that involved in computing the Mallows distance. If $q^{(k)}$'s are known, the objective function in (8) is minimized simply by finding the Mallows distance matching weights between the prototype and each image. The minimization can be performed separately for every image. When $q^{(k)}$'s are part of the optimization variables, the Mallows distance matching weights $w_{k,j}^{(i)}$ have to be optimized simultaneously for all the images $i \in \mathcal{C}_1$ because they affect each other through the constraint $\sum_{j=1}^{m_{i,1}} w_{k,j}^{(i)} = q^{(k)}$, for any $i = 1, ..., n'$.

When $q^{(k)}$'s and $w_{k,j}^{(i)}$'s are fixed, Equation (8) is simply a weighted sum of squares in terms of $z^{(k)}$'s and is minimized by the following formula:

$$z^{(k)} = \frac{\sum_{i=1}^{n'} \sum_{j=1}^{m_{i,1}} w_{k,j}^{(i)} v_{i,1}^{(j)}}{\sum_{i=1}^{n'} \sum_{j=1}^{m_{i,1}} w_{k,j}^{(i)}} , \quad k = 1, ..., m . \qquad (9)$$

We now summarize the D2-clustering algorithm, assuming the prototypes are initialized.

1) For every image i, set $c(i) = \operatorname{argmin}_{\eta=1,...,\bar{m}} \tilde{D}(\beta_i, \alpha_\eta)$.
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
   $$\sum_{i\in\mathcal{C}_\eta} \sum_{k=1}^{m'_{\eta,l}} \sum_{j=1}^{m_{i,l}} w_{k,j}^{(i)} \parallel z_{\eta,l}^{(k)} - v_{i,l}^{(j)} \parallel^2 .$$
   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 recursive splitting. As shown in Figure 4, suppose there are currently $\bar{m}'$ prototypes formed. For each prototype, the average $\tilde{D}$ distance between this prototype and all the images assigned to it is computed. The prototype with the maximum average distance is split to create the $\bar{m}' + 1$st prototype. The split is conducted in the following way. Suppose the prototype to be split is α_η, $1 \leq \eta \leq \bar{m}'$. An image assigned to α_η is randomly chosen, for instance, image β_i. Then we set $\alpha_{\bar{m}'+1} = \beta_i$. Note that α_η has already existed. We then treat the current value of α_η and $\alpha_{\bar{m}'+1}$ as initial values, and optimize them by applying the D2-

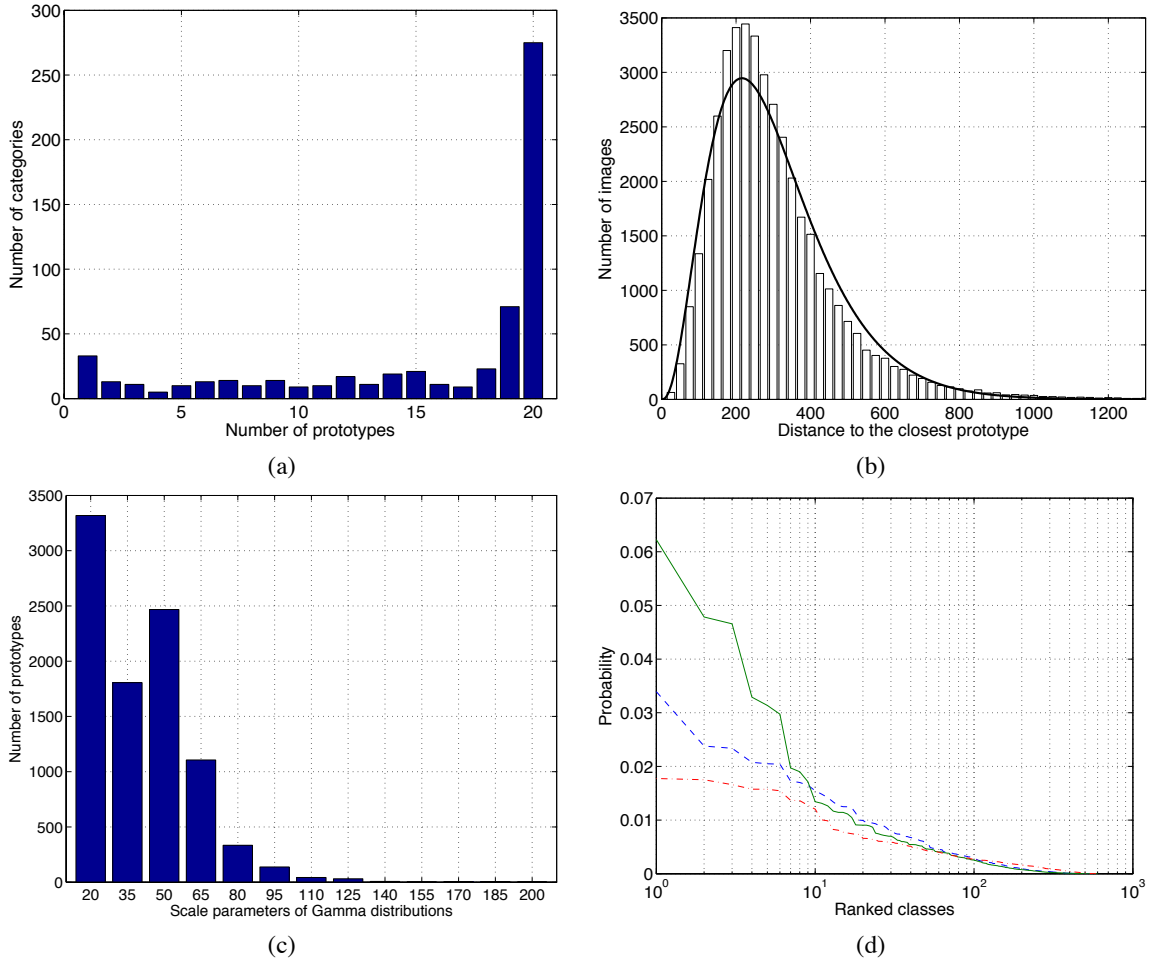


Fig. 5. Statistical modeling results. (a) Histogram of the number of prototypes in each class. (b) Fitting a Gamma distribution to the distance between an image and its closest prototype: the histogram of the distances is shown with the correspondingly scaled probability density function of an estimated Gamma distribution. (c) Histogram of the scale parameters of the Gamma distributions for all the prototypes formed from the training data. (d) The ranked concept posterior probabilities for three example images.

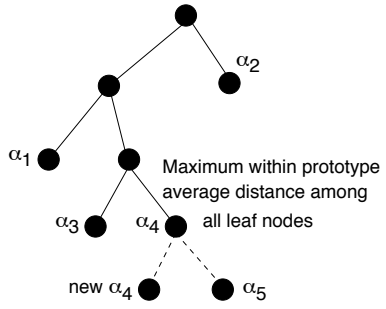


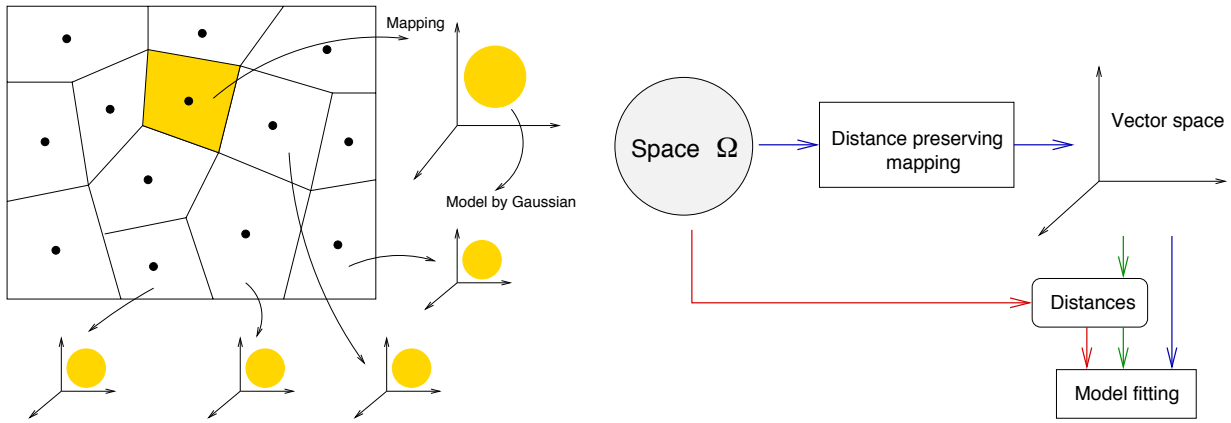
Fig. 4. Tree structured recursive split for initialization.

clustering only to images assigned to α_η at the beginning of the split. At the end of the D2-clustering, we have updated α_η and $\alpha_{\bar{m}'+1}$ and obtained a partition into the two prototypes for images originally in α_η. The splitting procedure is recursively applied to the prototype currently with maximum average distance until the maximum average distance is below a threshold or the number of prototypes exceeds a given threshold. During initialization, the probabilities $q_{\eta,l}^{(k)}$ in each $\alpha_{\eta,l}$ are set uniform for simplicity. Therefore, in Step 2a of the above algorithm, optimization can be done only over the matching weights $w_{k,j}^{(i)}$, and $w_{k,j}^{(i)}$ can be computed separately for each image.

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 5(a), demonstrates the wide variation in the level of image diversity within one concept.

IV. Mixture 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. When the context is clear, we may use component and cluster exchangeably because every mixture component is estimated using image signatures in one cluster. The likelihood of a signature under a given component reduces when the signature is further away from the corresponding prototype (i.e., component center).



(a) (b)

Fig. 6. Mixture modeling via hypothetical local mapping for space Ω. (a) Local mapping of clusters generated by D2-clustering in Ω. (b) Bypassing mapping in model estimation.

A. Modeling via Hypothetical Local Mapping (HLM)

Figure 5(b) shows the histogram of distances between images and their closest prototypes in one experiment. The curve overlaid on it is the probability density functions (pdf) of a fitted Gamma distribution. The pdf function is scaled so that it is at the same scale as the histogram. This plot only reflects local characteristics of image distances inside a cluster. We remind readers that distances between arbitrary images in a database are expected to follow more complex distributions. Examples of histograms of image distances based on different features and over large datasets are provided in [21]. 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 [10]:

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

where $\Gamma(\cdot)$ is the Gamma function [10].

Consider multivariate random vector $X = (X_1, X_2, ..., X_k)^t \in \mathcal{R}^k$ that follows a normal distribution with mean $\mu = (\mu_1, ..., \mu_k)^t$ 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, that is, the cluster associated with this prototype, can be locally approximated by $\mathcal{R}^k$, where $k = 2s$ and $\sigma^2 = b/2$. Here, approximation means there is a one to one mapping between points in Ω and in $\mathcal{R}^k$ that maximumly preserves all the pairwise distances between the points. The parameters s and b are estimated from the distances between images and their closest prototypes. In the local hypothetical space $\mathcal{R}^k$, images belonging to a given prototype are assumed to be generated by a multivariate normal distribution, the mean vector of which is the map of the prototype in $\mathcal{R}^k$. The pdf for a multivariate normal distribution $N(\mu, \sigma^2 I)$ is:

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

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:

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

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:

$$\phi(\beta) = \sum_{\eta=1}^{m} \omega_\eta \left(\frac{1}{\sqrt{\pi b}}\right)^{2s} e^{-\frac{\tilde{D}(\beta,\alpha_\eta)}{b}}. \quad (10)$$

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. Note that the mixture model can be made increasingly flexible by adding more components, and is not restricted by the rigid shape of a single Gaussian distribution. Here, the Gaussian distribution plays a similar role as a kernel in nonparametric density estimate. The fact that the histogram of within-cluster distances well fits a Gamma distribution, as shown in Figure 5(b), also supports the Gaussian assumption for individual clusters.

We call the above mixture modeling approach the *hypothetical local mapping (HLM)* method. In a nutshell, as illustrated in Figure 6(a), the metric space Ω is carved into cells via D2-clustering. Each cell is a neighborhood (or cluster) around its center, i.e., the prototype. Locally, every cluster is mapped to a Euclidean space that preserves pairwise distances. In the mapped space, data are modeled by a Gaussian distribution. It is assumed that the mapped spaces of the cells have the same dimensionality but possibly different variances. Due to the relationship between the Gaussian and Gamma distributions, parameters of the Gaussian distributions and the dimension of the mapped spaces can be estimated using only distances between each data point and its corresponding prototype. This implies that the actual mapping into $\mathcal{R}^k$ is unnecessary because the original distances between images and their corresponding prototypes, preserved in mapping, can be used directly. This argument is also illustrated in Figure 6(b). The local mapping from Ω to $\mathcal{R}^k$ is thus hypothetical and serves merely as a conceptual tool for constructing a probability measure on Ω.

Mixture modeling is effective for capturing the nonhomogeneity of data, and is a widely embraced method for classification and

clustering [12]. The main difficulty encountered here is the unusual nature of space Ω. Our approach is inspired by the intrinsic connection between k-means clustering and mixture modeling. It is known that under certain constraints on the parameters of component distributions, the classification EM (CEM) algorithm [18] used to estimate a mixture model is essentially the k-means algorithm. We thus generalize k-means to D2-clustering and form a mixture model based on clustering. This way of constructing a mixture model allows us to capture the clustering structure of images in the original space of Ω. Furthermore, the method is computationally efficient because the local mapping of clusters can be bypassed in calculation.

B. Parameter Estimation

Next, we discuss the estimation of the Gamma distribution parameters 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 [10]:

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

The above set of equations are solved by numerical methods. Because $2s = k$ and the dimension of the hypothetical space, k, 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 estimation for b with s^* given is $b^* = \bar{u}/s^*$. As an example, we show the histogram of the distances obtained from the training images and the fitted Gamma distribution with parameter $(\gamma : 3.5, 86.34)$ in Figure 5(b).

In our system, we assume that the shape parameter s of all the mixture components in all the image concept classes is common while the scale parameter b varies with each component. That is, the clusters around every prototype are mapped hypothetically to the same dimensional Euclidean space, but the spreadness of the distribution in the mapped space varies with the clusters. Suppose the total number of prototypes is $\bar{M} = \sum_k M_k$, where M_k is the number of prototypes for the kth image category, $k = 1, 2, ..., M$. Let $\mathcal{C}_j$, $j = 1, ..., \bar{M}$, be the index set of images assigned to prototype j. Note that the assignment of images to prototypes is conducted separately for every image class because D2-clustering is applied individually to every class, and the assignment naturally results from clustering. Let the mean of the distances in cluster j be $\bar{u}_j = \frac{1}{|\mathcal{C}_j|}\sum_{i\in\mathcal{C}_j} u_j$. It is proved in Appendix A that the maximum likelihood estimation for s and b_j, $j = 1, ..., \bar{M}$, is solved by the following equation:

$$\begin{cases} \log \hat{s} - \psi(\hat{s}) = \log\left[\prod_{j=1}^{\bar{M}} \bar{u}_j^{|\mathcal{C}_j|/N}/(\prod_{i=1}^{N} u_i)^{1/N}\right] \\ \hat{b}_j = \bar{u}_j/\hat{s}\ , j = 1, 2, ..., \bar{M} \end{cases} \tag{11}$$

The above equation assumes that $u_i > 0$ for every i. Theoretically, this is true with probability one. In practice, however, due to limited data, we may obtain clusters containing a single image, and hence some u_i's are zero. We resolve this issue by discarding distances acquired from clusters including only one image. In addition, we modify $\hat{b}_j = \bar{u}_j/\hat{s}$ slightly to

$$\hat{b}_j = \lambda\frac{\bar{u}_j}{\hat{s}} + (1-\lambda)\frac{\bar{u}}{\hat{s}},$$

where λ is a shrinkage factor that shrinks $\hat{b}_j$ toward a common value. We set $\lambda = \frac{|\mathcal{C}_j|}{|\mathcal{C}_j|+1}$, which approaches 1 when the cluster size is large. The shrinkage estimator is intended to increase robustness against small sample size for small clusters. It also ensures positive $\hat{b}_j$ even for clusters containing a single image. By this estimation method, we obtain $s = 5.5$ for the training image set. Figure 5(c) shows the histogram of the scale parameters, b_j's, estimated for all the mixture components.

In summary, the modeling process comprises the following steps:

1) For each image category, optimize a set of prototypes by D2-clustering, partition images into these prototypes, and compute the distance between every image and the prototype it belongs to.
2) Collect the distances in all the image categories and record the prototype each distance is associated with. Estimate the common shape parameter s for all the Gamma distributions and then the scale parameter b_j for each prototype j.
3) Construct a mixture model for every image category using Equation (10). Specifically, suppose among all the $\bar{M}$ prototypes, prototypes $\{1, 2, ..., M_1\}$ belong to category 1, and prototypes in $\mathcal{F}_k = \{\bar{M}_{k-1}+1, \bar{M}_{k-1}+2, ..., \bar{M}_{k-1}+M_k\}$, $\bar{M}_{k-1} = M_1 + M_2 + \cdots M_{k-1}$, belong to category k, $k > 1$. Then the profiling model $\mathcal{M}_k$ for the kth image category has distribution:

$$\phi(\beta \mid \mathcal{M}_k) = \sum_{\eta\in\mathcal{F}_k} \omega_\eta \Big(\frac{1}{\sqrt{\pi b_\eta}}\Big)^{2s} e^{-\frac{\tilde{D}(\beta,\alpha_\eta)}{b_\eta}}\ ,$$

where the prior ω_η is the empirical frequency of component η, $\omega_\eta = |\mathcal{C}_\eta|/\sum_{\eta'\in\mathcal{F}_k}|\mathcal{C}_{\eta'}|$, $\eta \in \mathcal{F}_k$.

V. 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 $\mathcal{E}(w_i)$. For instance, the word 'castle' is among the description of concept 160, 404, and 405. Then $\mathcal{E}(castle) = \{160, 404, 405\}$.

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

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

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

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

We then sort $\{q(\beta, w_1), q(\beta, w_2), ..., q(\beta, w_K)\}$ in descending order and select top ranked words. Figure 5(d) 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.

The main computational cost in annotation comes from calculating the Mallows distances between the query and every

prototype of all the categories. The linear programming involved in Mallows distance is more computationally costly than some other matching based distances. For instance, the IRM region-based image distance employed by the SIMPLIcity [32] system is obtained by assigning matching weights according to the "most similar highest priority (MSHP)" principle. By the MSHP principle, pairwise distances between two vectors across two discrete distributions are sorted. The minimum pairwise distance is assigned with the maximum possible weight, constrained only by conditions in (4). Then among the rest pairwise distances that can possibly be assigned with a positive weight, the minimum distance is chosen and assigned with the maximum allowed weight. So on so forth. From the mere perspective of visual similarity, there is no clear preference to either the optimization used in the Mallows distance or the MSHP principle. However, for the purpose of semantics classification, as the D2-clustering relies on the Mallows distance and it is mathematically difficult to optimize a clustering criterion similar to that in (5) based on MSHP, the Mallows distance is preferred. Leveraging advantages of both distances, we develop a screening strategy to reduce computation.

Because weights used in MSHP also satisfy conditions (4), the MSHP distance is always greater or equal to the Mallows distance. Since MSHP favors the matching of small pairwise distances in a greedy manner [32], it can be regarded as a fast approximation to the Mallows distance. Let the query image be β. We fist compute the MSHP distance between β and every prototype α_η, $D_s(\beta, \alpha_\eta)$, $\eta = 1, ..., \bar{M}$, as a substitute for the Mallows. These surrogate distances are sorted in ascending order. For the M' prototypes with the smallest distances, their Mallows distances from the query are then computed and used to replace the approximated distance by MSHP. The number of prototypes for which the Mallows distance is computed can be a fraction of the total number of prototypes, hence leading to significant reduction of computation. In our experiment, we set $M' = 1000$ while $\bar{M} = 9265$.

VI. Experimental Results

We present in this section annotation results and performance evaluation of the ALIPR system. Three cases are studied: (a) annotating images not included in the training set but within the Corel database; (b) annotating images outside the Corel database and checking the correctness of annotation words manually by a dedicated examiner; (c) annotating images uploaded by arbitrary online users of the system with annotation words checked by the users.

Because the first case evaluation avoids the arduous task of manual examination of words, a large set of images is evaluated. Performance achieved in this case, however, is optimistic because the Corel images are known to be highly clustered, that is, images in the same category are sometimes extraordinarily alike. In the real-world, annotating images with the same semantics can be harder due to the lack of such high visual similarity. This optimism is addressed by a "self-masking" evaluation scheme, which we will explain shortly. Another limitation of this case is that annotation words are assigned on a category basis for the Corel database. The words for a whole category are taken as ground truth for the annotation of every image in this category, and these annotations may not be complete for a particular image.

To address these issues, we experiment in the second case with general-purpose photographs acquired completely independent from Corel. Annotation words are manually checked for correctness on the basis of individual images. This evaluation process is labor intensive, taking several months to accomplish.

The third case evaluation best reflects users' impression of the annotation system. It is inevitably biased by whoever uses the online system. As will be discussed, the evaluation tends to be stringent.

We omitted comparing ALIPR with annotation systems developed by other research teams, for instance, those of Barnard et al. [2] and Carneiro et al. [5], for several reasons. We consider the ultimate test of an annotation system to be its performance assessed by users on images outside the training database. In the current literature, when the Corel database is used for training, annotation results have been reported using images inside the database. It is, however, a daunting task to implement systems of other researchers and subject all the experiments to the same constraints because these systems are highly sophisticated in mathematics and computation. An additional difficulty comes from the intensive human labor needed to examine multiple sets of test results manually. Moreover, well-known existing annotation systems are not aimed at real-time tagging and understandably are not provided online for arbitrary testing.

Recall from earlier discussion that the Corel database comprises 599 image categories, each containing 100 images, 80 of which are used for training. The training process takes an average of 109 seconds CPU time, with a standard deviation of 145 seconds on a 2.4 GHz AMD processor.

A. Performance on Corel Images

For each of the 599 image categories in Corel, we test on the 20 images not included in training. As mentioned previously, the "true" annotation of every image is taken as the words assigned to its category. An annotation word provided by ALIPR is labeled correct if it appears in the given annotation, wrong otherwise. There are a total of 417 distinct words used to annotate Corel images. Fewer words are used in the online ALIPR system because location names are removed.

We compare the results of ALIPR with a nonparametric approach to modeling. For brevity, we refer to the nonparametric approach as NP. In addition, we create two baseline annotation schemes that rely only on the prior frequencies of words. The frequencies of words in the given annotation vary vastly. The most frequent word is assigned to 148 concepts, while 249 words only once. Because a highly skewed prior of words favors the numerical assessment of annotation results, we compare ALIPR with the two baseline schemes to demonstrate the gain from concept learning beyond what can be achieved by the prior alone.

Suppose each word w_j, $j = 1, ..., 417$, appears J_j times in the annotation of all the concepts. Since a word cannot repeat for one concept, J_j equals the number of concepts that are annotated by w_j. The prior probability of w_j is set to $\kappa_j = J_j / \sum_{j'} J_{j'}$. We rank the words in descending order according to κ_j's. In the *most-frequent-word* scheme, we annotate every image by the same set of top ranked words arranged in the fixed order. For instance, if a single word is used to describe an image, we will always choose the word with the highest prior. In the second scheme, namely, random tagging, we randomly select words one by one according to the prior probabilities conditioned on no

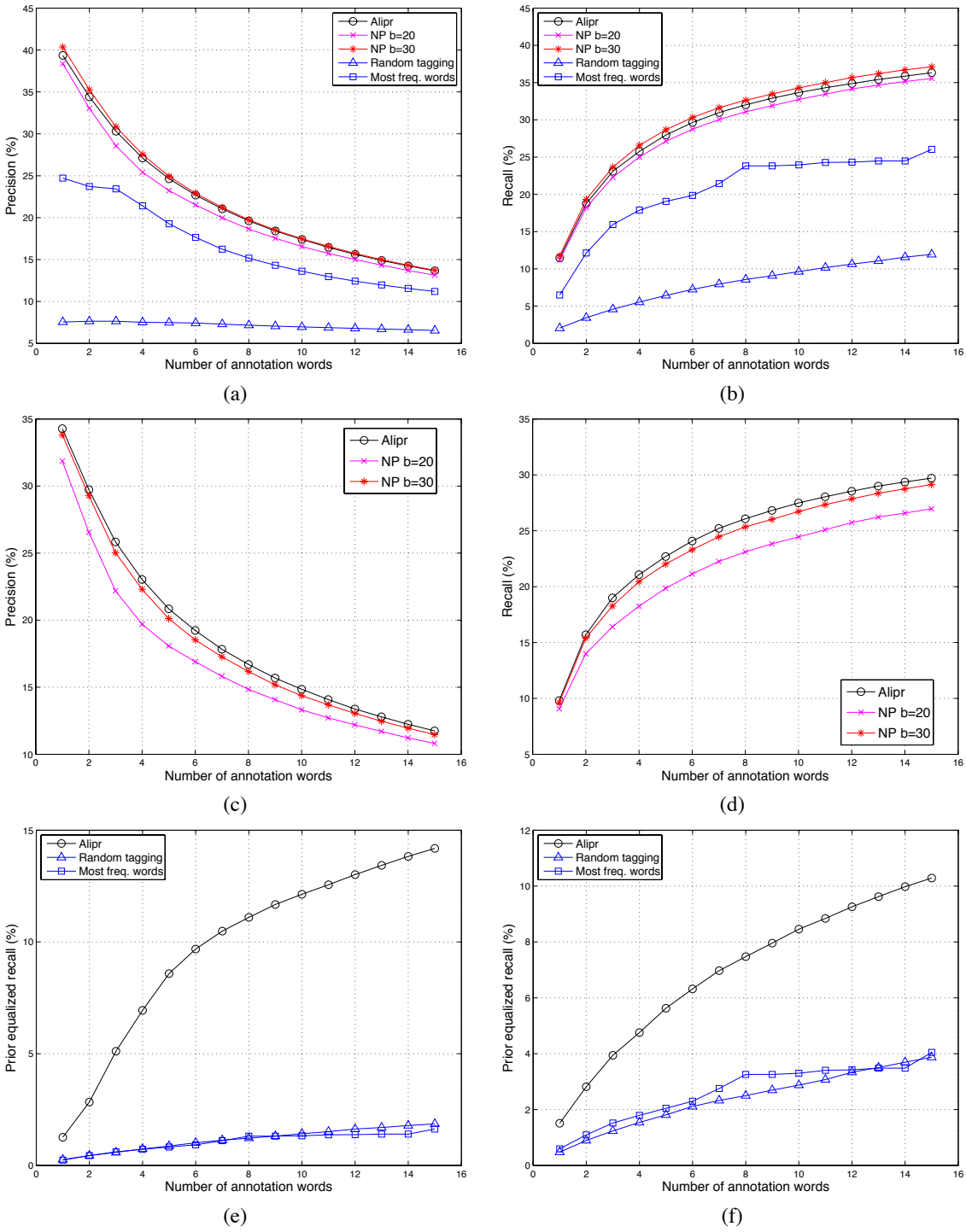


Fig. 7. Comparing annotation results of ALIPR, a nonparametric method NP, random tagging, and the most-frequent-word scheme, using test images in the Corel database. (a) Precision. (b) Recall. (c) Precision obtained by the self-masking evaluation scheme. (d) Recall obtained by the self-masking evaluation scheme. (e) The prior equalized recall. (f) The prior equalized recall over the words that are used in concept annotation more than once under the self-masking evaluation scheme.

duplicate. When a duplicate word is drawn, it is discarded and another random selection is made until a new word comes.

Under the NP approach, D2-clustering and the estimation of the Gamma distribution are not conducted. We form a kernel density estimate for each category, treating every image signature as a prototype. Suppose the training image signatures are $\{\beta_1, \beta_2, ..., \beta_N\}$. The number of images in class k is n_k, $\sum_{k=1}^{599} n_k = N$. Without loss of generality, assume $\{\beta_{\bar{n}_k+1}, ..., \beta_{\bar{n}_k+n_k}\}$ belong to class k, where $\bar{n}_1 = 0$, $\bar{n}_k = \sum_{k'=1}^{k-1} n_{k'}$, for $k > 1$. Under the nonparametric approach, the profiling model for each image category is

$$\phi(\beta \mid \mathcal{M}_k) = \sum_{i=\bar{n}_k+1}^{\bar{n}_k+n_k} \frac{1}{n_k} \left(\frac{1}{\sqrt{\pi b}}\right)^{2s} e^{-\frac{\tilde{D}(\beta,\beta_i)}{b}}.$$

In the kernel function, we adopt the shape parameter $s = 5.5$

since this is the value estimated using D2-clustering. When D2-clustering is applied, some clusters contain a single image. For these clusters, the scale parameter $b = 25.1$. In the nonparametric setting, since every image is treated as a prototype that contains only itself, we experiment with $b = 20$ and $b = 30$, two values representing a range around 25.1.

The NP approach is computationally more intensive during annotation than ALIPR because in ALIPR, we only need distances between a test image and each prototype, while the NP approach requires distances to every training image. We also expect ALIPR to be more robust for images outside Corel because of the smoothing across images introduced by D2-clustering, which will demonstrated by our results.

We assess performance using both precision and recall. Suppose the number of words provided by the annotation system is n_s, the number of words in the ground truth annotation is n_t, and the number of overlapped words between the two sets is n_c (i.e., number of correct words). Precision is defined as $\frac{n_c}{n_s}$, and recall is $\frac{n_c}{n_t}$. There is usually a tradeoff between precision and recall. When n_s increases, recall is ensured to increase, while precision usually decreases because words provided earlier by the system have higher estimated probabilities of occurring.

Figure 7(a) and (b) compare the results of ALIPR, NP, random tagging, and the most-frequent-word scheme in terms of precision and recall respectively. Precision and recall are shown with n_s increasing from 1 to 15. Both ALIPR and NP outperform random tagging and the most-frequent-word scheme significantly and consistently across n_s. The precision and recall percentages achieved by ALIPR are respectively 5.2 and 5.6 times as high as those by random tagging when one annotation word is assigned. Although the most-frequent-word scheme outperforms random tagging substantially, the words provided by the former tend to be general and less interesting to users. As will be discussed soon, real-world users often do not regard these generic words as correct annotation. We thus offset the skewed prior probabilities by defining a new measure of performance called *prior equalized recall*. This measure is the average of recall rates for individual words. Because the average is over words instead of images as in the computation of conventional recall, the words are weighted uniformly, avoiding dominance of a few high frequency words. Specifically, for a set of test images, suppose word j appears m times in the true annotation and the system selects this word correctly n times, the recall rate for word j is then $\frac{n}{m}$. The performance gap between ALIPR and random tagging, or the most-frequent-word scheme, is more pronounced under the prior equalized recall, as shown by Figure 7(e). Moreover, the most-frequent-word scheme performs similarly as random tagging.

The precision of ALIPR and NP with $b = 30$ is nearly the same, and the recall of NP with $b = 30$ is slightly better. As discussed previously, without cautious measures, using Corel images in test tends to generate optimistic results. Although the NP approach is favorable within Corel, it may have overfit this image set. Because it is extremely labor intensive to manually check the annotation results of both ALIPR and NP on a large number of test images outside Corel, we design the *self-masking* scheme of evaluation to counter the highly clustered nature of Corel images.

In self-masking evaluation, when annotating an image in category k with signature β, we temporarily assume class k is not trained and compute the probabilities of the image belonging to every other class m, $m \neq k$:

$$p_m(\beta) = \frac{\rho_m \phi(\beta \mid \mathcal{M}_m)}{\sum_{m' \neq k} \rho_{m'} \phi(\beta \mid \mathcal{M}_{m'})}\ ,$$
$$m = 1, 2, ..., k-1, k+1, ..., M\ .$$

For class k, we set $p_k(\beta) = 0$. With these modified class probabilities, words are selected using the same procedure described in Section V. Because image classes share annotation words, a test image may still be annotated with some correct words although it cannot be assigned to its own class. This evaluation scheme forces Corel images not to benefit from highly similar training images in their own classes, and better reflects the generalization capability of an annotation system. On the other hand, the evaluation may be negatively biased for some images. For instance, if an annotation word is used only for a unique class, the word becomes effectively "inaccessible" in this evaluation scheme. Precision and recall for ALIPR and NP under the self-masking scheme are provided in Figure 7(c) and (d). ALIPR outperforms NP for both precision and recall consistently over all n_s ranging from 1 to 15. This demonstrates that ALIPR can potentially perform better on images outside Corel. In Figure 7(f), the prior equalized recall rates under self-masking evaluation for the "accessible" words are compared between ALIPR and the two baseline annotation schemes.

An important feature of ALIPR is that it estimates probabilities for the annotation words in addition to ranking them. In the previous experiments, a fixed number of words is provided for all the images. We can also select words by thresholding their probabilities. In this case, images may be annotated with different numbers of words depending on the levels of confidence estimated for the words. Certain images not alike to any trained category may be assigned with no word due to low word probabilities all through. A potential usage of the thresholding method is to filter out such images and to achieve higher accuracy for the rest. Discarding a portion of images from a collection may not be a concern in some applications, especially in the current era of powerful digital imaging technologies, when we are often overwhelmed with the amount of images.

Figure 8(a) and (b) show the performance achieved by thresholding without and with self-masking respectively. For brevity of presentation, instead of showing precision and recall separately, the mean value of precision and recall is shown. When the threshold for probability decreases, the percentage of images assigned with at least one annotation word, denoted by p_a, increases. The average of precision and recall is plotted against p_a. When p_a is small, that is, when more stringent filtering is applied, annotation performance is in general better. In Figure 8(a), without self-masking, ALIPR and NP with $b = 30$ perform closely, with ALIPR slightly better at the low end of p_a. Results for NP with $b = 20$, worse than with $b = 30$, are omitted for clarity of the plots. In Figure 8(b), with self-masking, ALIPR performs substantially better. The gap between performance is more prominent at the low end of p_a.

B. Performance on Images Outside the Corel Database

To assess the annotation results for images outside the Corel database, we applied ALIPR to more than 54,700 images created by users of flickr.com and provide the results at the Website: `alipr.com`. This site also hosts the ALIPR demonstration system that performs real-time annotation for any image either

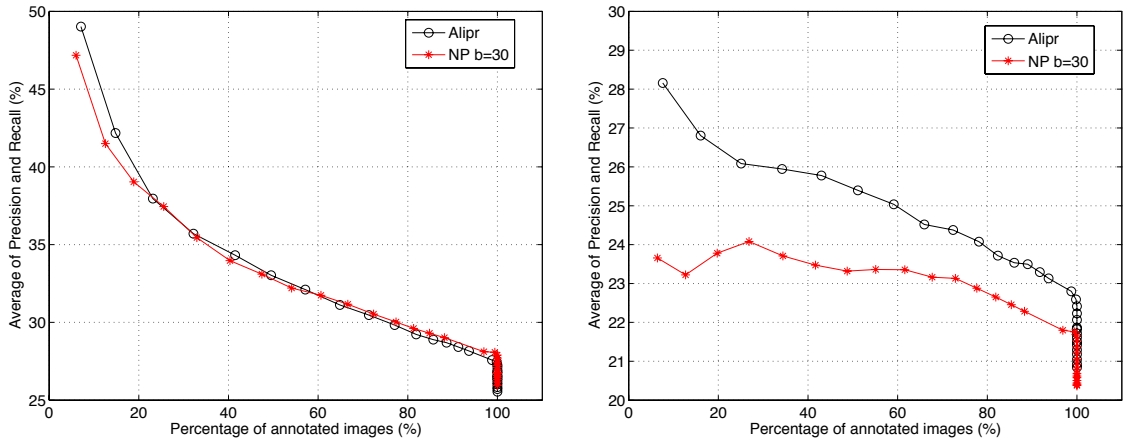


Fig. 8. Comparing annotation results of ALIPR and a nonparametric method NP achieved by thresholding word probabilities for test images in the Corel database. (a) The average of precision and recall without self-masking. (b) The average of precision and recall with self-masking.

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

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

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

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

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

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

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

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

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

Fig. 9. Automatic annotation for photographs and paintings. The words are ordered according to estimated likelihoods. The six photographic images were obtained from flickr.com. The six paintings were obtained from online Websites.

uploaded directly by the user or downloaded from a user-specified URL. Annotation words for 12 images downloaded from the Internet are obtained by the online system and are displayed in Figure 9. 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.

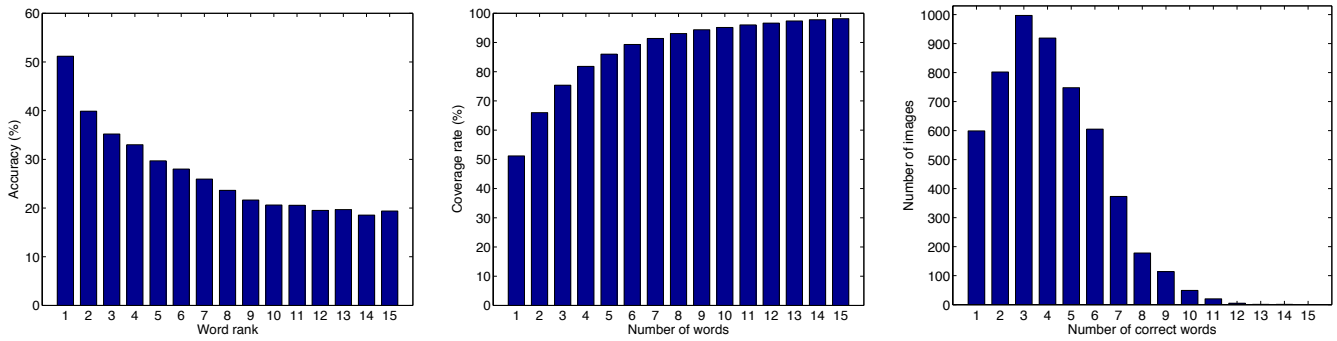
There are not many completely failed examples. However, we picked some unsuccessful examples, as shown in Figure 10. In

(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

Fig. 10. 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.



(a) (b) (c)

Fig. 11. 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.

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. 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 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 [16] and various heuristics [2] 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. 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.

Annotation performance is reported from several aspects in Figure 11. Each image is assigned with 15 words listed in the descending order of the likelihood of being relevant. Figure 11(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 11(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 11(c). On average, 4.1 words are correct for each image.

C. Performance Based on ALIPR Users

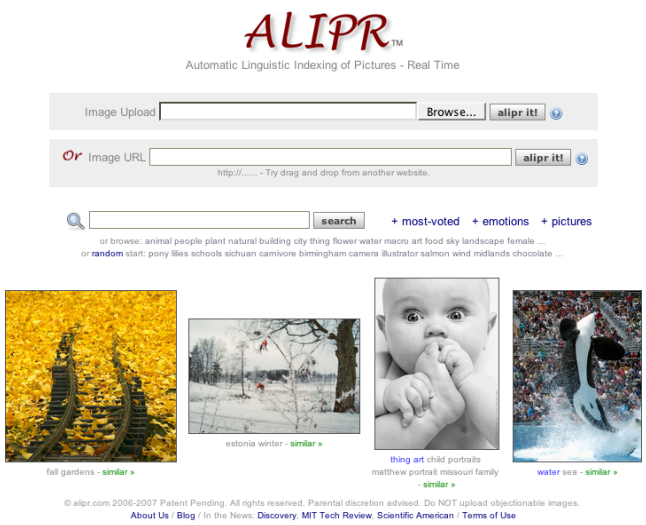


Fig. 12. The ALIPR Web-based demonstration site allows the users to upload or enter the URL of a test image.

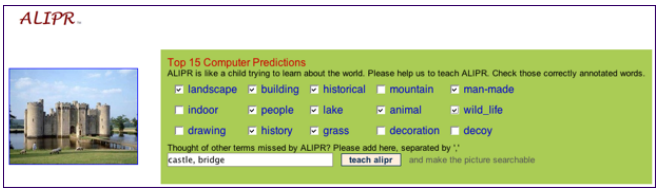


Fig. 13. The ALIPR system suggests tags in real time.

The alipr.com Website allows users to upload their own pictures, or specify an image URL, and acquire annotation in real time. The Web interface is shown in Figure 12. Every word has a check box preceding it. Upon uploading an image, the user can click the check box if he or she regards the ALIPR-predicted word as correct and can also enter new annotation words in the provided text box (Figure 13).

The site was made public on November 1, 2006. It was subsequently reported on the news. In a given day, as many as 2,400 people used the site. Many of them used the site as an image search engine as we provide keyword-based search, related image search, and visual similarity search of images (Figures 14 and 15). Many of these functions rely on the accurate tags stored for the images. In order to make the site more useful for image search, we added more than one million images from terragalleria.com and flickr.com. We used ALIPR to verify the keywords or tags provided by these sites. For reporting the accuracy of ALIPR, however, we use only those images uploaded directly by online users.

Over time, we observed the following behaviors of the users who uploaded images:

- Many users are more stringent on considering an ALIPR-predicted word as correct. They often only check words that accurately reflect the main theme of the picture but neglect other appropriate words. For example, for the picture in Figure 16(a), the user checked the words building and car as correct words. But the user did not check other reasonable predictions including people. Similarly, for the picture in Figure 16(b), the user checked only the words people and sky. Other reasonable predictions can include sport and male. In rare cases, as in Figure 16(c), the user checked all reasonable predictions.
- Users tend to upload difficult pictures just to challenge the system. Although we mentioned on the *About Us* page that the system was designed only for color photographic images, many people tested with gray-scale images, manipulated photos, cartoons, sketches, framed photos, computer graphics, aerial scenes, etc. Even for the photographic images, they often use images with rare scenes (Figure 17).

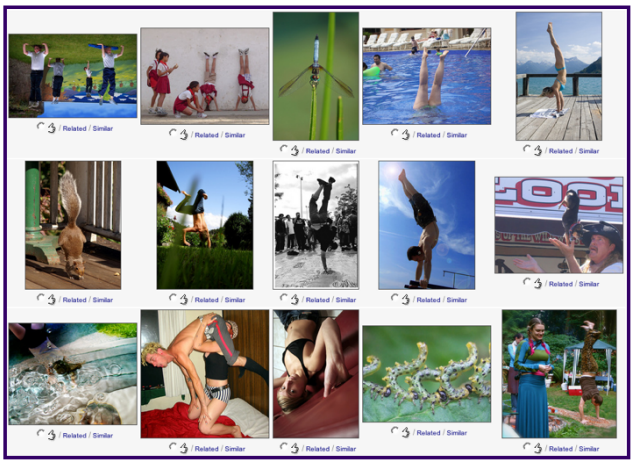


Fig. 14. Related images search is provided, leveraging the tags checked by the users.

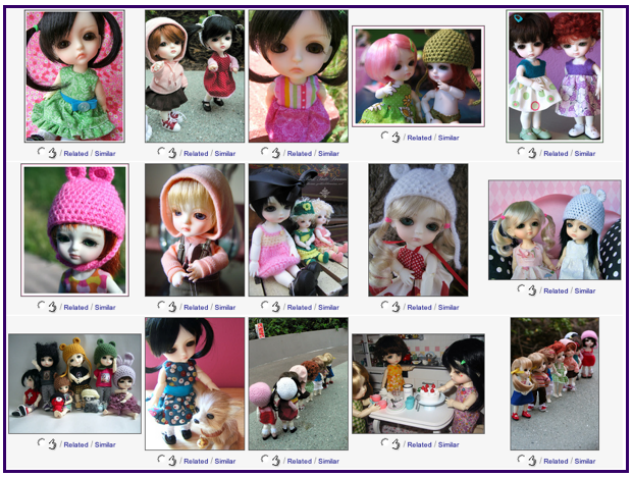


Fig. 15. Visual similarity search uses the SIMPLIcity system and the tags checked by the users.

Up to the point when the manuscript was written, a total of 10,809 images had been uploaded with some user-checked words. On average, 2.24 words among the top 15 words predicted by ALIPR are considered correct for a picture. The users added an average of 1.67 words for each picture. A total of 3,959 unique

(a) (b) (c)

Fig. 16. Sample results collected on the alipr.com site. Underlined words are those considered correct by the user who provided the image. (a) ALIPR words: people, man-made, cloth, guard, parade, holiday, yuletide, sport, landscape, building, historical, child, car, painting, mural. User-added words: female, bikini, model, pose, outdoor. (b) ALIPR words: people, cloth, sky, man-made, water, balloon, ocean, boat, sport, female, male, couple, landscape, house, animal. User-added word: (none). (c) ALIPR words: landscape, building, man-made, train, garden, sculpture, estate, rural, historical, people, ocean, tree, isle, grass, car. User-added word: water.

(a) (b) (c) (d)

Fig. 17. Pictures of rare scenes are often uploaded to the alipr.com site. (a) ALIPR words: man-made, texture, color, people, indoor, food, painting, royal guard, fruit, feast, holiday, mural, cloth, abstract, guard. User-added words: thirsty, kitty. (b) ALIPR words: building, historical, landscape, animal, landmark, ruin, grass, snow, wild life, sky, people, photo, rock, fox, castle. User-added words: forest, cloud, lake. (c) ALIPR words: flower, natural, pattern, landscape, texture, man-made, rural, pastoral, plant, tree, green, rock, color, animal, grass. User-added words: China. (d) ALIPR words: people, man-made, building, historical, landscape, life, face, indoor, food, occupation, cloth, child, youth, decoration, male. User-added words: furniture, Buddha.

TABLE I
THE DISTRIBUTION OF THE NUMBER OF CORRECTLY-PREDICTED WORDS.

# of checked tags	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of images	3277	2824	2072	1254	735	368	149	76	20	22	3	1	2	3	3
(%)	30.3	26.1	19.2	11.6	6.8	3.4	1.4	0.7	0.2	0.2	0.	0.	0.	0.	0.

TABLE II
THE DISTRIBUTION OF THE NUMBER OF USER-ADDED WORDS.

# of added tags	0	1	2	3	4	5	6	7	8	9	10	11
# of images	3110	3076	1847	1225	727	434	265	101	18	3	0	3
(%)	28.8	28.5	17.1	11.3	6.7	4.0	2.5	0.9	0.2	0.	0.	0.

IP addresses have been recorded for these uploaded images. The distribution of the number of correctly-predicted words and user-added words are shown in Tables I and II, respectively. A total of 295 words, among the vocabulary of 332 words in the ALIPR dictionary, have been checked by the users for some pictures.

VII. 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 as one substantial step toward meeting this established need. Our work has shown that the computer can learn, using a large collection of example images, to annotate general photographs with substantial accuracy. To achieve this, we have developed novel statistical modeling and optimization methods useful for establishing probabilistic relationships between images and words. The ALIPR system has been evaluated rigorously using real-world pictures.

Future work to improve the accuracy of the system can take many directions. First, the incorporation of 3-D information in the learning process may improve the models, perhaps through learning via stereo images or 3-D images. Additionally, shape information can be utilized to improve the modeling process. Second, better and larger amounts of training images per semantic concept may produce more robust models. Contextual information may also help in the modeling and annotation process. Third, this

method holds promise for various application domains, including biomedicine. Finally, the system can be integrated with other retrieval methods to improve usability.

Acknowledgments: The research is supported in part by the US National Science Foundation under Grant Nos. 0705210, 0219272 and 0202007. We thank Diane Flowers for providing manual evaluation on annotation results, Dhiraj Joshi for designing a Web-based manual evaluation system and for incorporating image data from collaborators and other Websites, Walter Weiss for developing the initial keyword-based search and for maintaining many of our systems, David M. Pennock of Yahoo! for providing test images, Hongyuan Zha for useful discussions on optimization, and Takeo Kanade for encouragements. We would also like to acknowledge the comments and constructive suggestions from reviewers.

J. Li developed the D2-clustering and the generalized mixture modeling algorithms. Both authors contributed to the design of the ALIPR system and conducted the experimental studies.

Appendix A

We now prove Equation (11) gives the ML estimation for the parameters of the Gamma distributions under a common shape parameter. Recall that the total number of prototypes across all the image classes is $\bar{M}$ and the index set of images assigned to prototype j is $\mathcal{C}_j$, $j = 1, ..., \bar{M}$. We need to estimate the scale parameter b_j for every prototype j and the common shape parameter s. The collection of distances is $\mathbf{u} = (u_1, u_2, ..., u_N)$, $N = \sum_{j=1}^{\bar{M}} |\mathcal{C}_j|$. The ML estimator maximizes the log likelihood:

$$L(\mathbf{u}|s, b_1, b_2, ..., b_{\bar{M}}) = \sum_{j=1}^{\bar{M}} \sum_{i \in \mathcal{C}_j} \log f(u_i)$$

$$= \sum_{j=1}^{\bar{M}} \sum_{i \in \mathcal{C}_j} \left[(s-1)\log u_i - s \log b_j - \frac{u_i}{b_j} - \log \Gamma(s) \right] . \quad (12)$$

With a fixed s, $L(\mathbf{u}|s, b_1, b_2, ..., b_{\bar{M}})$ can be maximized individually on every b_j:

$$\max L(\mathbf{u}|s, b_1, b_2, ..., b_{\bar{M}})$$

$$= \sum_{j=1}^{\bar{M}} \max \sum_{i \in \mathcal{C}_j} \left[(s-1)\log u_i - s \log b_j - \frac{u_i}{b_j} - \log \Gamma(s) \right] . \quad (13)$$

Since $\sum_{i \in \mathcal{C}_j} \left[(s-1)\log u_i - s \log b_j - \frac{u_i}{b_j} - \log \Gamma(s) \right]$ is a concave function of b_j, its maximum is determined by setting the first derivative to zero:

$$\sum_{i \in \mathcal{C}_j} -\frac{s}{b_j} + \frac{u_i}{b_j^2} = 0 ,$$

Let

$$\bar{u}_j = \frac{\sum_{i \in \mathcal{C}_j} u_i}{|\mathcal{C}_j|}$$

be the average distance for prototype j. Then, b_j is solved by

$$b_j = \frac{\bar{u}_j}{s} . \quad (14)$$

Now substitute Equation (14) into (13) and suppress the dependence of L on b_j:

$$\max L(\mathbf{u} \mid s)$$

$$= \sum_{j=1}^{\bar{M}} \max \sum_{i \in \mathcal{C}_j} \left[s \log s + s \cdot (\log \frac{u_i}{\bar{u}_j} - \frac{u_i}{\bar{u}_j}) - \log \Gamma(s) - \log u_i \right] .$$

Note that $\log \Gamma(s)$ is a convex function of s. It is easy to show that $L(\mathbf{u} \mid s)$ is a concave function of s, and hence is maximized by setting its first derivative to zero:

$$N \log s + \sum_{j=1}^{\bar{M}} \sum_{i \in \mathcal{C}_j} \log \frac{u_i}{\bar{u}_j} - N\psi(s) = 0 ,$$

which is equivalent to:

$$\log \hat{s} - \psi(\hat{s}) = \log \left[\prod_{j=1}^{\bar{M}} \bar{u}_j^{|C_j|/N} / (\prod_{i=1}^{N} u_i)^{1/N} \right] . \quad (15)$$

Combining (14) and (15), we have proved the ML estimator in Equation (11).

References

[1] A. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo, "Trademark matching and retrieval in sports video databases," *Proc. International Workshop on Multimedia Information Retrieval, ACM*, pp. 79–86, Augsburg, Bavaria, Germany, September 2007.

[2] 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.

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

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

[5] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 394–410, 2007.

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

[7] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *Journal of Machine Learning Research*, vol. 5, 913-939, August 2004.

[8] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, 65 pages, 2008, to appear.

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

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

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

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

[13] 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," *Proc. Multimedia Information Retrieval Workshop*, pp. 15–22, New York, NY, 2004.

[14] X. He, W.-Y. Ma, and H.-J. Zhang. "Learning an image manifold for retrieval," *Proc. ACM Multimedia Conference*, pp. 17–23, New York, NY, 2004.

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

[16] 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.

[17] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[18] G. J. McLachlan and D. Peel, *Finite Mixture Models*, New York: Wiley, 2000.
[19] C. L. Mallows, "A not on asymptotic joint normality," *Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 508–515,1972.
[20] F. Monay and D. Gatica-Perez, "On image auto-annotation with latent space models," *Proc. ACM Multimedia Conference*, pp. 275–278, Berkeley, CA, 2003.
[21] T. Quack, U. Monich, L. Thiele, and B. S. Manjunath, "Cortina: a system for large-scale, content-based web image retrieval," *Proc. ACM Multimedia Conference*, pp. 508-511, New York City, NY, 2004.
[22] S. T. Rachev, "The Monge-Kantorovich mass transference problem and its stochastic applications," *Theory of Probability and its Applications*, 29: 647-676, 1984.
[23] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distribution with applications to image databases," *Proc. International Conference on Computer Vision*, pp. 59-66, Bombay, India, 1998.
[24] 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.
[25] A. Singhal, J. Luo, and W. Zhu, "Probabilistic spatial context models for scene content understanding," *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 235-241, Madison, WI, June 2003.
[26] 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.
[27] J. R. Smith and S.-F. Chang, "VisualSEEk: A fully automated content-based image query system," *Proc. ACM Multimedia Conference*, pp. 87–98, Boston, MA, 1996.
[28] K. Tieu and P. Viola, "Boosting image retrieval," *International Journal of Computer Vision*, vol. 56, no. 1/2, pp. 17–36, 2004.
[29] C. Tomasi, "Past performance and future results," *Nature*, vol. 428, page 378, March 2004.
[30] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," *Proc. ACM Multimedia Conference*, pp. 107–118, 2001.
[31] 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.
[32] 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.
[33] J. Z. Wang and J. Li, "Learning-based linguistic indexing of pictures with 2-D MHMMs," *Proc. ACM Multimedia Conference*, pp. 436-445, Juan Les Pins, France, ACM, 2002.
[34] 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.

Jia Li received the BS degree in electrical engineering from Xi'an Jiao Tong University in 1993, the MSc degree in electrical engineering in 1995, the MSc degree in statistics in 1998, and the PhD degree in electrical engineering in 1999, all from Stanford University. She is an associate professor of statistics and by courtesy appointment in computer science and engineering at The Pennsylvania State University, University Park. She worked as a visiting scientist at Google Labs in Pittsburgh from 2007 to 2008, a research associate in the Computer Science Department at Stanford University in 1999, and a researcher at the Xerox Palo Alto Research Center from 1999 to 2000. Her research interests include statistical modeling and learning, data mining, computational biology, image processing, and image annotation and retrieval. She is a senior member of the IEEE.

James Z. Wang received the BS degree (summa cum laude) in mathematics and computer science from the University of Minnesota, Twin Cities, in 1994, the MSc degree in mathematics and the MSc degree in computer science from Stanford University, Stanford, California, in 1997, and the PhD degree in medical information sciences from the Stanford University Biomedical Informatics Program and Computer Science Database Group in 2000. He is an associate professor at the College of Information Sciences and Technology, and by courtesy in the Department of Computer Science and Engineering and the Integrative Biosciences Program at The Pennsylvania State University, University Park. He was the holder of the endowed PNC Technologies Career Development Professorship from 2000 to 2006, a recipient of a US National Science Foundation Career award in 2004, a Visiting Professor of the Robotics Institute at Carnegie Mellon University from 2007 to 2008, and the lead guest editor for IEEE Transactions on Pattern Analysis and Machine Intelligence Special Issue on Real-World Image Annotation and Retrieval in 2008. He is a senior member of the IEEE.