

Toward Bridging the Annotation-Retrieval Gap in Image Search

Ritendra Datta, Weina Ge, Jia Li, and James Z. Wang
The Pennsylvania State University

By combining novel statistical modeling techniques and the WordNet ontology, we offer a promising new approach to image search that uses automatic image tagging directly to perform retrieval.

Quick ways to capture pictures, cheap devices to store them, and convenient mechanisms for sharing them are all part and parcel of our daily lives. With multitudes of pictures to deal with, everyone would benefit from smart programs to manage photo collections, tag them automatically, and make them searchable through keywords. To satisfy such needs, the multimedia, information retrieval, and computer vision communities have, time and again, attempted automated image annotation, as we have witnessed in the recent past.¹⁻⁴ While many interesting ideas have emerged, we haven't seen much attention paid to the direct use of automatic annotation for image search. Usually, it is assumed that good annotation implies quality image search. Moreover, most past approaches are too slow to be of practical use for today's massive picture collections.

The problem would not be interesting if all pictures came with tags, which in turn were reliable. Unfortunately, for today's picture collections such as Yahoo! Flickr, this is seldom the case. These collections are characterized by their mammoth volumes, lack of reliable tags, and the diverse spectrum of topics they cover. In Web image search systems such as those of Yahoo! and Google, surrounding text forms the basis of keyword searches, which come with their own problems.

In this article, we discuss our attempt to build an image search system based on automatic tagging. Our goal is to treat automatic annotation as

a means of satisfactory image search. We look at realistic scenarios that arise in image search, and propose a framework that can handle them through a unified approach. To achieve this, we look at how pictures can be accurately and rapidly placed into a large number of categories, how the categorization can be used effectively for automatic annotation, and how these annotations can be harnessed for image search. For this, we use novel statistical models and the WordNet ontology,⁵ as well as state-of-the-art content-based image retrieval (CBIR) methods⁶⁻⁸ for comparison. Our method significantly outperforms competing strategies for this problem, and also suggests nonintuitive results.

Bridging the gap

Our motivation to bridge the annotation-retrieval gap is driven by a desire to effectively handle common, challenging cases of image search in a unified manner. Four real-world scenarios, schematically presented in Figure 1, are as follows:

- *Scenario 1.* Either a tagged picture or a set of keywords is used as a query. The problem arises when all or part of the image database (such as Web images) is not tagged, making this portion inaccessible through text queries. We study how our annotation-driven image search approach performs in first annotating the untagged pictures, then performing multiple keyword queries on the partially tagged picture collection.
- *Scenario 2.* An untagged image is used as a query, aiming to find semantically related pictures or documents from a tagged database or the Web. We look at how our approach performs in first tagging the query picture, then in retrieval.
- *Scenario 3.* The query image as well as all or part of the image database is untagged. This is the case that best motivates CBIR, since the only available information is visual content. We study the effectiveness of our approach in tagging the query image and the database, then in retrieval.
- *Scenario 4.* A tagged query image is used to search a tagged image database. The problem is that these tags might be noisy and unreliable, as is common in user-driven picture tagging portals. We study how our approach

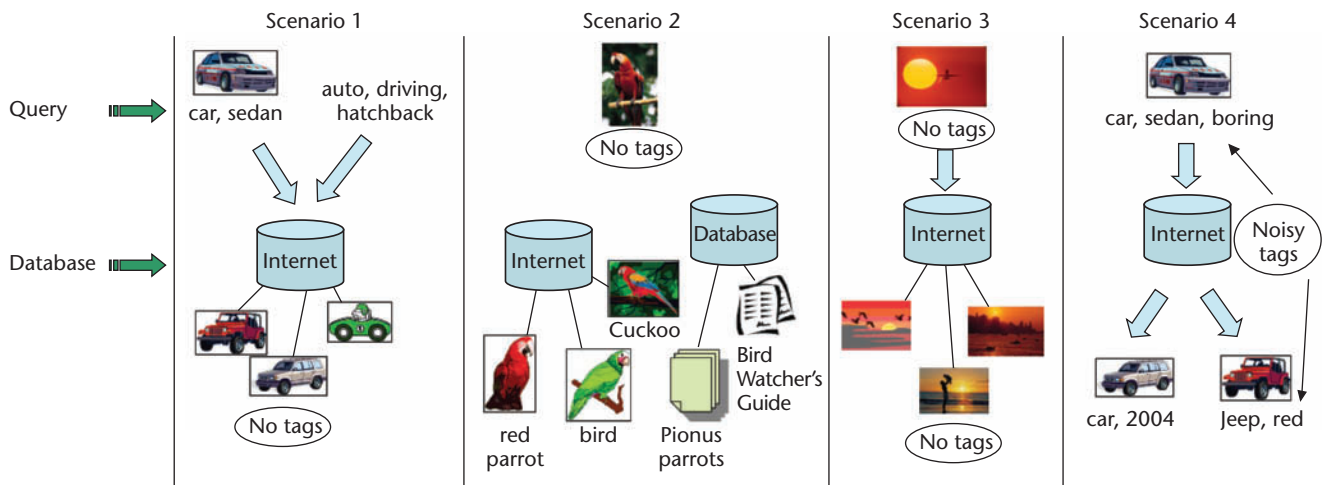


Figure 1. Four common scenarios for real-world image retrieval.

helps improve tagging by reannotation, and subsequently performs retrieval.

In each case, we look at reasonable and practical alternative strategies for search, with the help of a state-of-the-art CBIR system. For scenario 4, we are also interested in analyzing the extent to which our approach helps improve annotation under varying noise levels.

Additional goals include the ability to generate precise annotations of pictures in near-real time. While most previous annotation systems assess performance based on annotation quality alone, this measure is only part of our goal. For us, the main challenge is to have the annotations help generate meaningful retrieval. To this end, we developed our approach by first building a near-real-time categorization algorithm (about 11 seconds per image) capable of producing accurate results and then generating categorization-based annotation, ensuring high precision and recall. With this annotation system in place, we assess its performance as a means of image search under the preceding scenarios.

Model-based categorization

We employ generative statistical models for accurate, near-real-time categorization of generic images. This implies training independent statistical models for each image category using a small set of training images. We can then assign category labels to new pictures via smart use of the likelihood overall models. In our system, we use two generative models (per image category) to provide evidence for categorization from two different aspects of the images. We generate final categorization by combining these evidences.

In the case of many generic image categories,

it is challenging to build a robust classifier. Feature extraction becomes extremely critical, since it must have the discriminative power to differentiate between a broad range of image categories. We base our models on the following intuitions: For certain categories such as sky, marketplace, ocean, forests, and Hawaii, or those with dominant background colors such as paintings, color and texture features might be sufficient to characterize them. In fact, a structure or composition for these categories may be too hard to generalize. On the other hand, categories such as fruits, waterfalls, mountains, lions, and birds might not have dominating color or texture but often have an overall structure or composition that helps us identify them despite heavily varying color distributions.

Motivated by these facts, we built two models to capture different visual aspects: a structure-composition model that uses Beta distributions to capture color interactions in a very flexible manner and a Gaussian mixture model in the joint color-texture feature space. In essence, we want to examine each picture from two separate viewpoints and place it in a category after consulting both. While our approach models color and texture explicitly, it models the segments and edges implicitly through the structure-composition (S-C) model.

Structure-composition models

We wanted a way to represent how the colors interact with each other in certain picture categories. The description of an average beach picture could comprise a set of relationships between different colored regions—for example, orange (sun) completely inside light blue (sky), and light blue sharing a long border with dark blue (ocean). For tiger pictures, we have yellow and black regions

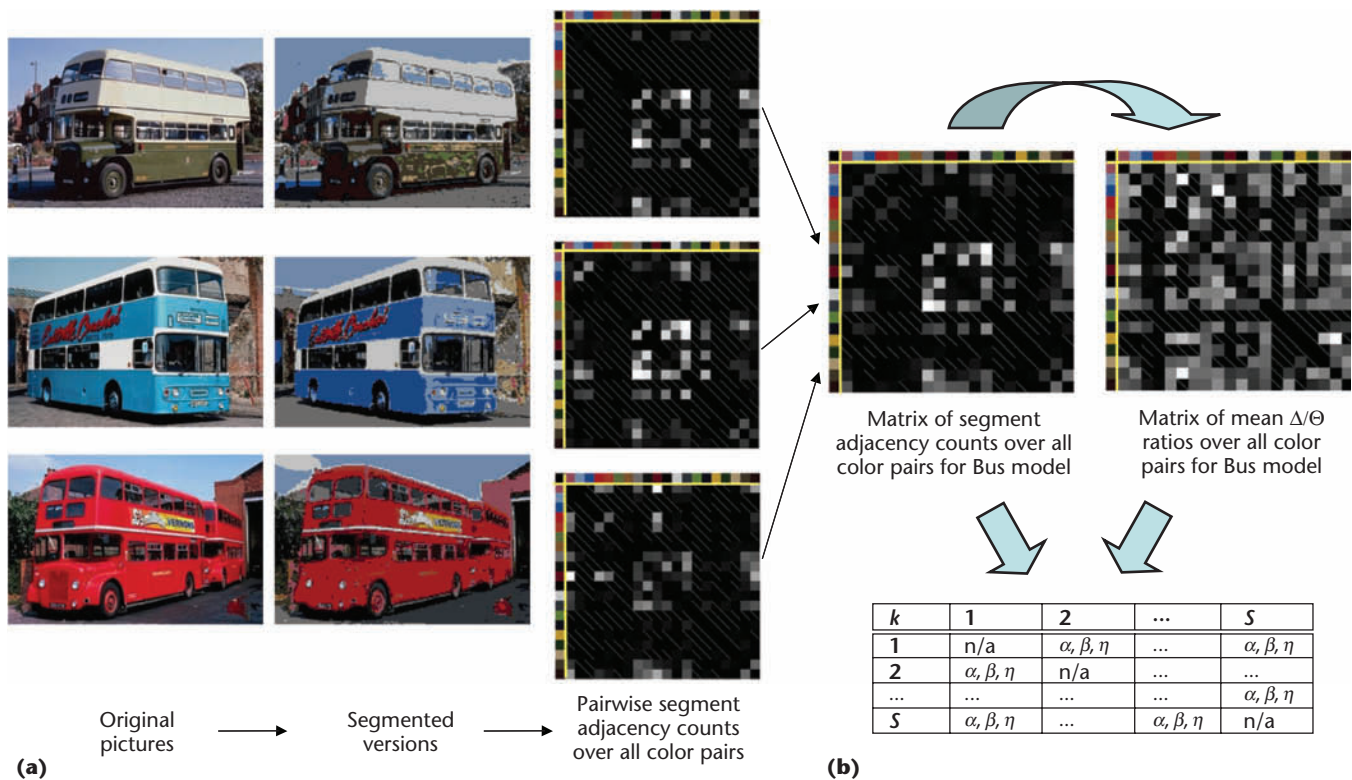


Figure 2. Steps toward generating the structure-composition (S-C) model. (a) Three training pictures from the bus category, their segmented forms, and a matrix representation of their segment adjacency counts. (b) The corresponding matrix representations over all three training pictures. These matrices combine to produce the S-C model, shown here schematically as a matrix of Beta parameters and counts.

sharing similar borders with each other (stripes) and remaining colors interacting without much pattern. The description for coarse texture patterns such as pictures of beads of different colors could comprise any color (bead) surrounding any other color (bead), some color (background) completely containing most colors (beads), and so on. This idea led to a principled formulation of our rotation and scale invariant S-C models.

Given the set of all training images across categories, we take every pixel from each image and map it to the *LUV color space*, which is a perceptually uniform space consisting of the luminance component *L* and the chrominance components *U* and *V*, and perform *K*-means clustering on a manageable random subsample of it. This process yields a set of 20 cluster centroids, such as shades of red or yellow, giving us a color palette representing the entire training set.

We then perform a nearest-neighbor-based segmentation on each training image *I* by assigning every pixel a cluster label closest to it in the color space to obtain a new image *J*, which is essentially a color-quantized representation. This helps build a uniform model representation for all image categories. To get disjoint segments from the image, we perform an *8-connected component labeling* (that groups pixels into islands, treating diagonally located pixels as having an

adjoint, or neighboring, relationship) on the color-segmented image *J*.

Let χ_i be the set of neighboring segments to segment s_i . Here, a neighborhood implies that for two segments s_i and s_j , at least one pixel is in each of s_i and s_j that is 8-connected. We wish to characterize the interaction of colors by modeling how each color shares (if at all) boundaries with every other color. Let's denote by Δ the length of the shared border between a segment and one of its neighboring segments, and by Θ the total length of the segment's perimeter.

We want to model the Δ/Θ ratios for each color pair by the flexible Beta distribution, which is appropriate for modeling ratios in the $[0,1]$ range. The distribution is characterized by shape parameters α and β . We build models consisting of a set of Beta distributions for every color pair.

For each category, and for every color pair, we find instances in the training set (for that category) in which segments of that color pair share a common border. Let the number of such instances be η . We then compute the corresponding set of Δ/Θ ratios and estimate a Beta distribution—that is, parameters α and β —using these values for that color pair. Figure 2 shows the overall process of estimating S-C models, along with their representation.

In the model representation, diagonal entries

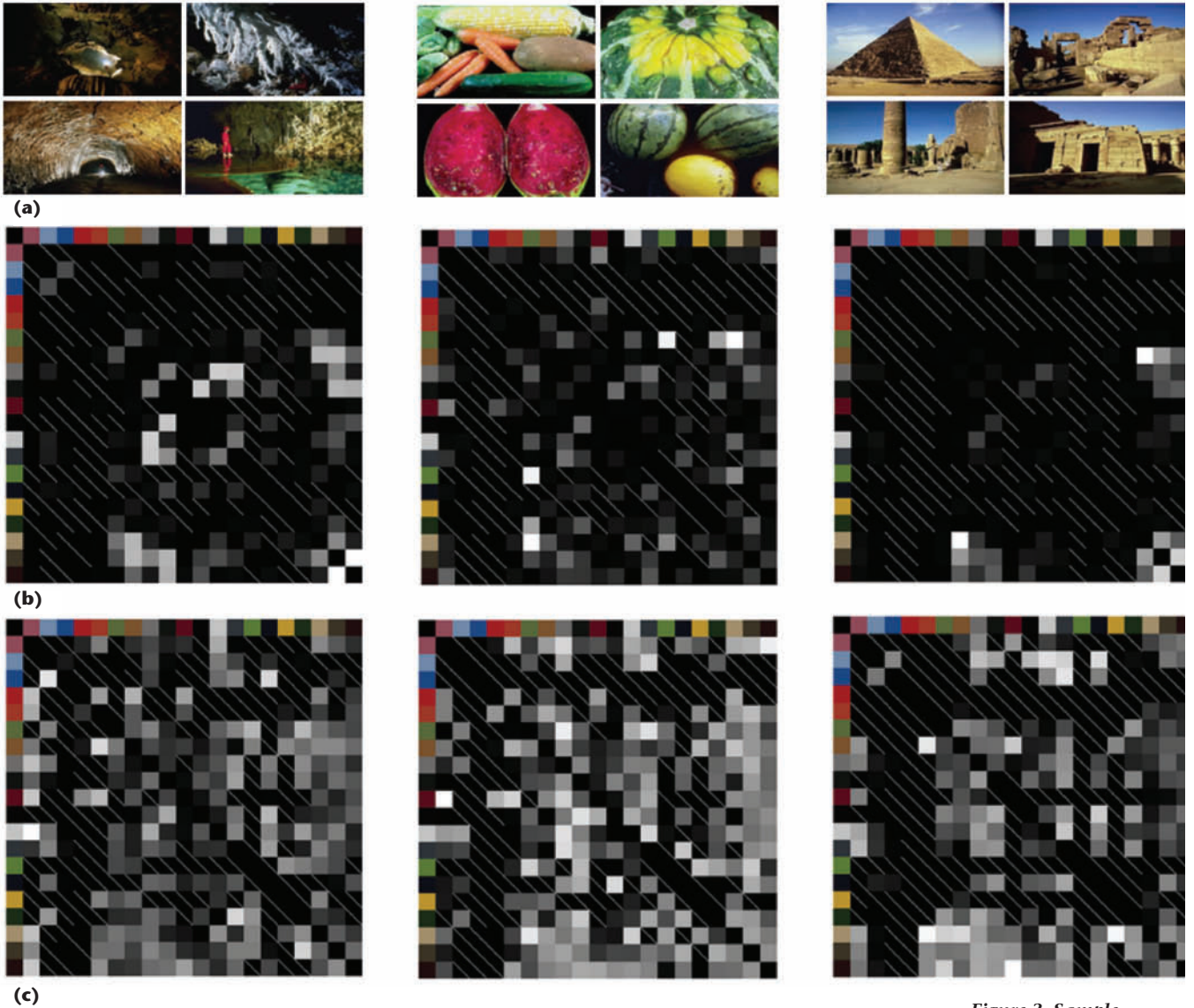


Figure 3. Sample categories and corresponding S-C model representations. (a) Sample training pictures. (b) Matrices of segment adjacency counts. (c) Matrices of mean Δ/Θ ratios. Brightness levels represent the relative magnitude of values.

are not defined, and the matrix entries are asymmetric, since we treat colors as ordered pairs. The number of samples η used to estimate the α and β for each entry are stored alongside as well. We build and store such models for every category. In Figure 3, we show simple representations of the learned models for three such picture categories. (See Figure 2 to better understand these representations.) From these representations, computing the actual model parameters is straightforward.

To make the modeling process fast, we use a simple *moment matching method* (statistical parameter estimation by equating sample moments to corresponding population moments) for estimating the Beta distributions. Depending on the number of samples η we have for a color pair, we face a parameter estimation issue. For low values of η , the estimates are undefined or poor. Yet, we

might often have training pictures with few or no borders of a given color pair. But, this may not necessarily mean that such borders won't occur in test pictures.

We play it safe here by treating those color pairs as “unknown.” For this reason, we first estimate parameters α'_k and β'_k for the distribution of all Δ/Θ ratios across all color pairs in a given category k of training images, and then store them as part of that model. Now, for a test picture, we segment it the same way we did in training and find all the segment boundaries. For each such boundary x , we compute the Δ/Θ ratio coming from color pair (i, j) . We then compute the probability of x given a model k as

$$P_{SC}(x|k) = \frac{\eta}{\eta+1} f(x|\alpha, \beta) + \frac{1}{\eta+1} f(x|\alpha'_k, \beta'_k)$$

which has parameters defined in the usual way, for color pair (i, j) in model k . Here, P_{sc} is the conditional density for the S-C model, and $f(\cdot|\alpha, \beta)$ is the Beta density function. What we have here essentially is a regularization measure often used in statistics when the amount of confidence in some estimate is low: a weighted probability is computed instead of the original one, with the weights varying with the number of samples used for estimation. Naturally, when η gets smaller, more importance is given to the prior estimates (α'_k, β'_k) over all color pairs, because we don't have high confidence in the estimates for that specific color pair.

Finally, we use a standard conditional independence assumption for computing the overall likelihood of a test image I given an S-C model k . This is simply the product of the individual P_{sc} values corresponding to each border x in the image. We take the logarithm of this value and denote this log-likelihood as $l_{sc}(IM_k)$. Here, the conditional independence assumption might not hold true in reality, but it helps reduce computation significantly, which is essential for producing rapid categorization.

Color-texture models

Many image categories, especially those not containing specific objects, can probably be described best by their color/texture distributions. In fact, a well-defined structure might not even exist per se for high-level categories such as China and Europe, but the overall ambience that certain colors and textures form often characterizes them.

We use a mixture of multivariate normal distributions to model the joint color-texture feature space for a given category. The motivation is simple: In many cases, two or more representative regions in the color-texture feature space can represent the image category best. For example, beach pictures typically have one or more yellow areas (sand), a blue nontextured area (sky), and a blue textured region (sea). Mixture models for the normal distribution are well-studied, with many tractable properties in statistics. Yet, these simple models have not been widely exploited in image categorization.

We extract the same color and texture features used in the Automated Linguistic Indexing of Pictures (ALIP),³ taking nonoverlapping 4×4 blocks of the image. For each category, we get mixture-model parameter estimates out of the features extracted from the training pictures. We

use Bouman's cluster package for this estimation.⁹

As usual, this package implements the well-known expectation-maximization (EM) algorithm for mixture models. Using this package, we compute and store color-texture models for each picture category. Thus, for a 4×4 block x and learned model θ_k , we denote the probability of x given that model as $P_{ct}(x|\theta_k)$.

Ignoring spatial dependence among blocks, we finally compute the log-likelihood of a test image I given a category k by taking the log of the product of $P_{ct}(x|\theta_k)$ values over each 4×4 block of the image I . We denote by $l_{ct}(IM_k)$ this log-likelihood. We argue that the conditional independence assumption here is reasonable, for three reasons: we intend rapid categorization, which is aided by this simplifying assumption; our S-C model already captures spatial relationships at a more meaningful granularity than fixed size blocks; and dependencies among blocks have been explicitly modeled by Li and Wang's method,³ which our approach empirically outperforms.

Annotation and retrieval

We use the categorization models for annotation, which in turn helps us with image search.

Automatic tagging basics

Three important considerations we make in automatic tagging are

- how strongly the categorization results favor a tag,
- how frequently we see that tag in the training set—that is, the likelihood of its chance appearance, and
- whether the tag is meaningful in the context of the picture's other tags.

Suppose we have a 600-category training-image data set (the setting for all our experiments), each category annotated by three to five tags—for example, [sail, boat, ocean] and [sea, fish, ocean]—with many tags shared among categories. Initially, all the tags from each category are pooled together. Tag saliency is measured in a way similar to computing inverse document frequency in the document retrieval domain. The total number of categories in the database is C . We count the number of categories that contain each unique tag t , and denote it by $F(t)$. For a given test image I , the S-C models and the C-T

models independently generate ranked lists of predicted categories. We generate these lists by placing all the categories in descending order of log-likelihoods, for each model type. We choose the top 10 categories that each model predicts and pool them together for annotation. We denote the union of all unique words from both models by $U(I)$, which forms the set of candidate tags. Let the frequency of occurrence of each unique tag t among the top 10 model predictions be $f_{sc}(t|I)$ and $f_{ct}(t|I)$, respectively.

WordNet is a semantic lexicon that groups English words into sets of synonyms and records the semantic relations among the synonym sets.⁵ Based on this ontology, researchers have proposed numerous measures of word relatedness. A measure that we observed to produce reasonable relatedness scores was the Leacock and Chodorow (LCH),¹⁰ which we use in our experiments. We convert this relatedness measure to a distance measure by taking the exponent and normalizing it to get a $[0, 24]$ range of values. Inspired by Jin et al.,¹¹ we measure a congruity score for a candidate tag t and denote it by $G(t|I)$ for tag t and image I . (See Datta et al. for further technical details.)¹²

In essence, a tag that is semantically distinct from the rest of the words predicted will likely have a low congruity score, while a closely related one will have a high score. The measure can potentially remove noisy and unrelated tags from consideration. Having computed the three measures, for each of which higher scores indicate better fitness for inclusion, the overall score for a candidate tag t is given by a linear combination as follows:

$$R(t|I) = a_1 f(t|I) + \frac{a_2}{\log C} \log \frac{C}{1+F(t)} + a_3 G(t|I)$$

Here, $a_1 + a_2 + a_3 = 1$ and $f(t|I) = b f_{sc}(t|I) + (1 - b) f_{ct}(t|I)$ is the main model combination step for annotation, linearly combining the evidences each model generated for the tag. Experiments show us that combining the models helps significantly over either model independently. The value of b is a measure of relative confidence in the S-C model. A tag t is chosen for annotation only when its score is within the top ϵ percentile among the candidate tags, where ϵ basically controls the number of annotations generated per image.

After performing experiments on a validation set of 1,000 pictures, we arrive at a satisfactory set of values for the aforementioned constants,

**Our aim is to make
the entire picture collection
searchable by keywords
and to allow all types of
searches under a common
framework.**

namely $a_1 = 0.4\%$, $a_2 = 0.2\%$, $b = 0.3\%$, and $\epsilon = 60\%$. This choice of weights depends on the image database used and can be determined by an appropriate grid search. Also, after the system chooses candidate tags, the final set of tags are selected fairly independently of each other. Our hope is that candidate tag selection reflects the tags' cooccurrence in the training set and that the congruity measure implicitly introduces dependence, but a joint modeling of tags might produce better results.

Performing annotation-driven image search

We are now equipped to search pictures, using automatic annotation and a bag-of-words distance. Whenever tags are missing in the query image and/or the database, the system performs automatic annotation. Next, a bag-of-words distance measure (and hence, a support for multiple keyword queries) between query picture tags and the database tags helps rank the pictures, which also supports multiple keyword queries. When tags are present but are known to be noisy, the system intuitively combines these tags and the learned models to improve the tagging prior to performing search.

Our aim is to make the entire picture collection searchable by keywords and to allow all types of searches under a common framework. The bag-of-words distance measure we use is the *average aggregated minimum distance*.¹³ Put plainly, the approach attempts to match each word in bag 1 to the semantically closest word in bag 2 (again, using the WordNet-based LCH distance), then match each word in bag 2 to the closest in bag 1, and finally compute the weighted average of the matched distances based on the bag sizes to ensure that the measure is symmetric.

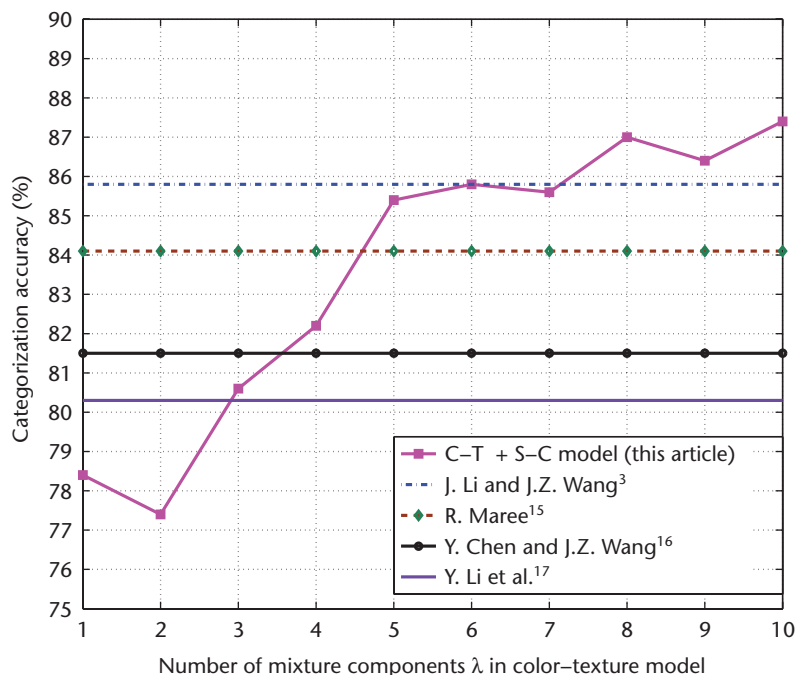


Figure 4. Categorization accuracies for the 10-class experiment. Performance of our combined S-C + C-T model with varying numbers of mixture components in the C-T model. We also show previously reported best results for comparison.

Experimental validation

We investigate our system’s performance on four grounds: how accurately it identifies picture categories, how well it tags pictures, how well it performs reannotation of noisy tags, and how much it improves image search for the four scenarios we described earlier. However, the improvement in image search quality is the main focus of this work. The data sets we look at consist of

- 54,000 Corel stock photos encompassing 600 picture categories and
- a 1,000 picture collection from Yahoo! Flickr.

Of the Corel collection, we use 24,000 to train the two statistical models, and use the rest for assessing performance.

Identifying picture categories

To fuse the two models for categorization, we use a simple combination strategy¹⁴ that results in impressive performance. Given a picture, we rank each category k based on likelihoods from both models, to get ranks $\pi_{sc}(k)$ and $\pi_{ct}(k)$. We then linearly combine these two ranks for each category, $\pi(k) = \sigma\pi_{sc}(k) + (1 - \sigma)\pi_{ct}(k)$, with $\sigma = 0.2$ working best in practice. We then assign the category that yields the highest linearly combined score to this picture.

We decide how well our system is doing in

predicting categories by involving two picture data sets. The first one is a standard 10-class image data set that has been commonly used for the same research question. Using 40 training pictures per category, we assess the categorization results on another 50 per category. We compute accuracies while varying the number of mixture components in the C-T model.

We present our results—along with those previously reported¹⁵⁻¹⁷ on the same data—in Figure 4. We see that our combined model does a better job at identifying categories than previous attempts. Not surprisingly, as we increase the number of mixture components, the C-T models become more refined. We thus continue to get improved categorization with greater components, although more components mean more computation as well.

Our second data set consists of the same 600 category Corel images that were used in the ALIP system.³ With an identical training process for the two models (the number of mixture components is chosen as 10), we observe the categorization performance on a separate set of 27,000 pictures. What we find is that the actual picture categories coincide with our system’s top choice 14.4 percent of the time, are within our system’s top two choices 19.3 percent of the time, and are within our system’s top three choices 22.7 percent of the time. The corresponding accuracy values for the ALIP system are 11.9, 17.1, and 20.8 percent, respectively.

Our system takes about 26 seconds to build a structure-composition category model and about 106 seconds to build a color-texture model, both on a 40-picture training set. As with generative models, we can independently and in parallel build the models for each category and type. To predict the top five ranked categories for a given test picture, our system takes about 11 seconds. Naturally, we have a system that is orders of magnitude faster than the ALIP system, which takes about 30 minutes to build a model and about 20 minutes to test on a picture. Most other automatic tagging systems in the literature do not explicitly report speed.

However, many of them depend on sophisticated image segmentation algorithms, which can easily bottleneck performance. The improved performance in model building means that even larger numbers of models can be built (for example, one model per unique tag), and the modeling process can be made dynamic (retraining at intervals) to accommodate changing picture col-









Image				
Our labels	sky, city, modern, building, Boston	door pattern, Europe, historical building, city	train, car, people, life, city	man, office, indoor, fashion, people
Flickr labels	Amsterdam, building, Maheler4, Zuitas	Tuschinski, Amsterdam	honeymoon, Amsterdam	hat, Chris, cards, funny
Image				
Our labels	lake, Europe, landscape, boat, architecture	lion, animal, wild life, Africa, super-model	speed, race, people, Holland, motorcycle	dog, glass, animal, rural, plant
Flickr labels	Amsterdam, canal, water	leopard, cat, ragged photo, animal	Preakness, horse, jockey, motion, unfound photo	Nanaimo Torgersons, animal, Quinn, dog, camera-phone

Figure 5. Sample automatic tagging results on Yahoo! Flickr pictures taken in Amsterdam, along with manual tags.

lections, such as Web sites that let users upload pictures.

Tagging the pictures

We now look at how our system performs when it comes to automatic picture tagging. Tagging is fast, since it depends primarily on the categorization speed. Over a random test set of 10,000 Corel pictures, our system generates an average of seven tags per picture. We use standard metrics for evaluating annotation performance: *precision*, the fraction of tags predicted that are correct, and *recall*, the fraction of tags for the picture that are correctly guessed. Average precision over this test set is 22.4 percent, while average recall is 40.7 percent.

Thus, on an average, roughly one in four of our system's predicted tags are correct, while our system guessed two in five correct tags. In general, results of this nature are useful for filtering and classification. A potential domain of thousands of tags can be reduced to a handful, making human tagging much easier, as used in the Automatic Linguistic Indexing of Pictures—Real Time system (ALIPR; see <http://alipr.com>).¹⁸ Increased homogeneity and reduced ambiguity in the tagging process are additional benefits.

We make a more qualitative assessment of tagging performance on the 1,000 Flickr pictures. We point out that the training models are still those built with Corel pictures, but because they represent the spectrum of photographic images well, these models serve as fair knowledge bases.

In this case, most automatically generated tags are meaningful and generally very encouraging. In Figure 5, we present a sampling of these results. Getting quantitative performance is harder here because Flickr tags are often proper nouns (such as names of buildings and people) that aren't contained in our training base.

Reannotating noisy tags

We assess our annotation system's performance in improving tagging at high noise levels. The scenario of noisy tags, at a level denoted by e , is simulated in the following manner: For each of 10,000 test pictures with the original (reliable) tags, a new set of tags is generated by replacing a tag by a random tag e fraction of the times, at random, and is unchanged at other times. The resulting noisy tags for these test images, when assessed for performance, give precision and recall values that directly correlate with e . In the absence of learned models, this is our baseline case.

When such models are available, we can use the noisy tags and the categorization models to reannotate the pictures, because the noisy tags still contain exploitable information. We perform reannotation by simply treating each noisy tag t of a picture I as an additional instance of the word in the pool of candidate tags.

In effect, we increment the values of $f_{sc}(tI)$ and $f_{ca}(tI)$ by a constant Z , thus increasing the chance of t to appear as a tag. The value of Z controls how much we want to promote these tags and is natu-

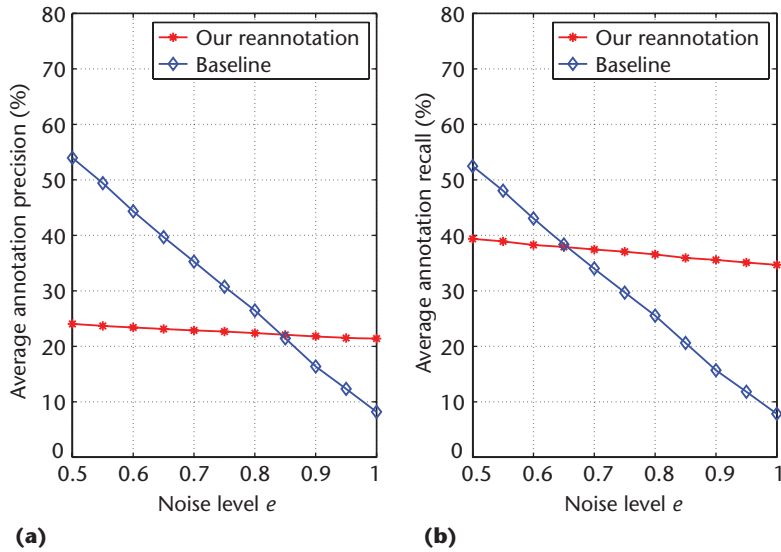


Figure 6. (a) Precision and (b) recall achieved by reannotation, with varying noise levels in original tags. Note that the linear correlation of the baseline case to e is intrinsic to the noise simulation.

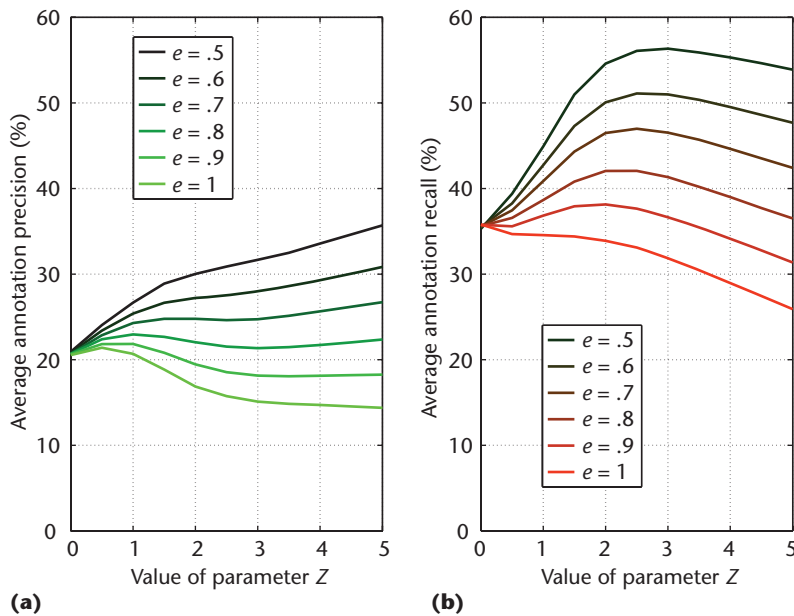


Figure 7. (a) Precision and (b) recall achieved by reannotation, varying parameter Z , shown for five noise levels.

rally related to the noise in the tags. Figure 6 shows the annotation precision and recall this approach achieves, with e varying from 0.5 (moderately noisy) to 1 (completely noisy, no useful information) for the case of $Z = 0.5$. We notice that, at high levels of noise, our reannotation produces better performance than the baseline.

Figure 7 shows a summary of a more general analysis of the trends, with larger values of Z —

that is, greater confidence in the noisy tags. The graph shows precision and recall for our reannotation, varying Z for each of five error levels. We observe that for the same value of Z , less noisy tags lead to better reannotation. Moreover, after reaching a peak near $Z = 2.5$, the recall starts to drop, while precision continues to improve. This graph can be useful in selecting parameters for a desired precision or recall level after reannotation, given an estimated level of noise in the tags.

Searching for pictures

Compared to traditional methods, our approach improves image search performance. We assume that either the database is partially tagged, or the search is performed on a picture collection visually coherent with some standard knowledge base. In all our cases, the statistical models are learned from the Corel data set. For scenario 4, we assume that everything is tagged, but some tags are incorrect or inconsistent. Once again, we train a knowledge base of 600 picture categories, and then use it to do categorization and automatic tagging on the test set. This set consists of 10,000 randomly sampled pictures from among the remaining Corel pictures (those not used for training).

We now consider the four image search scenarios we discussed in the “Bridging the gap” section. For each scenario, we compare results of our annotation-driven image search strategy with alternative strategies. For those alternative strategies involving CBIR, we use the IRM distance used in the SIMPLiCity system⁸ to get around the missing tag problem in the databases and queries.

We choose the alternative strategies and their parameters by considering a wide range of possible methods. We assess the methods based on the standard information retrieval concepts of precision (percentage of retrieved pictures that are relevant) and recall (percentage of relevant pictures that are retrieved). We consider two pictures/queries to be relevant whenever there is overlap between their set of tags. In this article, we report performance in terms of precision, which is usually considered a more useful metric in information retrieval. Recall performance can be found in our earlier work.¹²

Scenario 1. Here, the database doesn’t have any tags. Queries may be in the form of either one or more keywords or tagged pictures.

Keyword queries on an untagged picture database is a key problem in real-world image search.

We look at 40 percent (out of the 417 unique ones in the Corel set) of randomly chosen pairs of query words (each word is chosen from the 417 unique words in our training set). In our strategy, we perform a search by first automatically tagging the database, and then retrieving images based on bag-of-words distances between query tags and our predicted tags.

The alternative CBIR-based strategy used for comparison is as follows: without any image as a query, CBIR can't be performed directly on query keywords. Instead, suppose the system is provided access to a knowledge base of tagged Corel pictures. A random set of three pictures for each query word is chosen from the knowledge base, and we compute IRM distances between these images and the database. We then use the average IRM distance over the six pictures to rank the database pictures. We report these two results, along with the random results, in Figure 8a. Clearly, our method performs impressively and significantly better than the alternative approach.

Scenario 2. In this case, the query is an untagged picture, and the database is tagged. First, we tag the query picture automatically, and then rank the database pictures using bag-of-words distance. We randomly choose 100 query pictures from Corel and test it on the database of 10,000 pictures. The alternative CBIR-based strategy we use is as follows: the IRM distance is used to retrieve five (empirically observed to be the best count) pictures most visually similar to the query, and the union of all their tags is filtered using the expression for $R(t|I)$ to get automatic tags for the query (the same way that we filter our annotation, as we described in the "Annotation and retrieval" section). Now, the search proceeds identically to ours. We present these results, along with the random scheme, in Figure 8b. As the figure shows, our strategy has a significant performance advantage over the alternate strategy. The CBIR-based strategy performs almost as poorly as the random scheme, which is probably because of the direct use of CBIR for tagging.

Scenario 3. In this case, neither the query picture nor the database is tagged. We test 100 random picture queries on the 10,000-image database. Our strategy is simply to tag both the query picture as well as the database automatically, and then perform bag-of-words-based retrieval. Without any tags present, the alternative CBIR-based strategy we used here is essen-

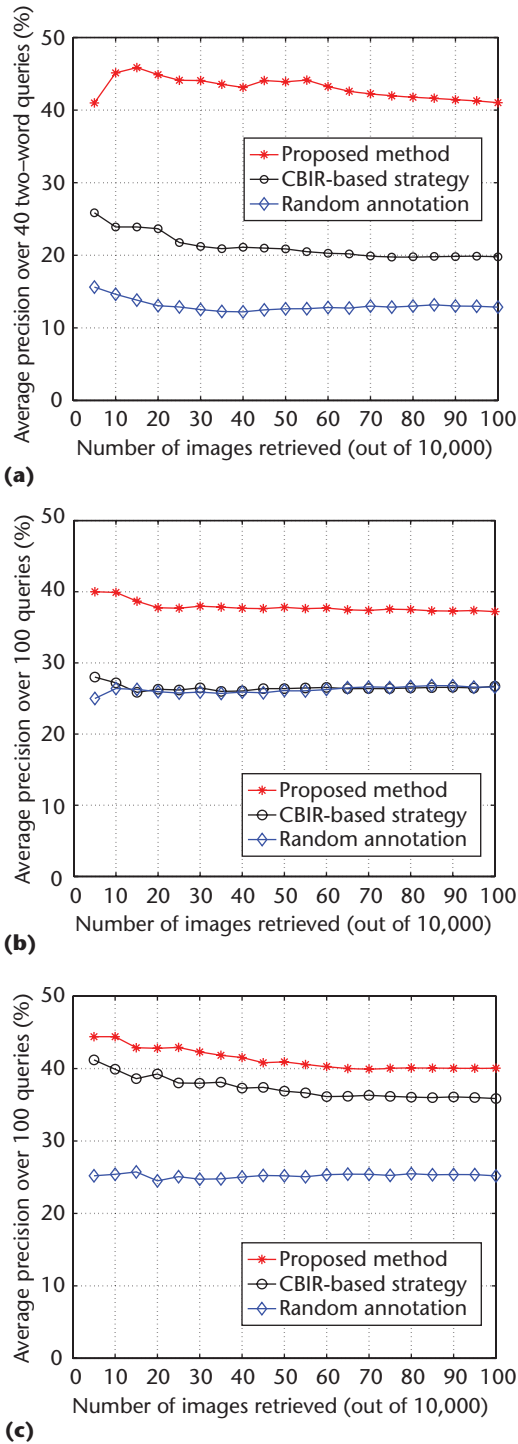


Figure 8. Retrieval precision under (a) scenario 1, (b) scenario 2, and (c) scenario 3, compared to baseline/random strategies.

tially a standard use of the IRM distance to rank pictures based on visual similarity to the query. We present these results, along with the random case, in Figure 8c. Once again, we see the advantage of our common image search framework over straightforward visual similarity-based retrieval. What we witness is how, in an indirect way, the learned knowledge base helps to

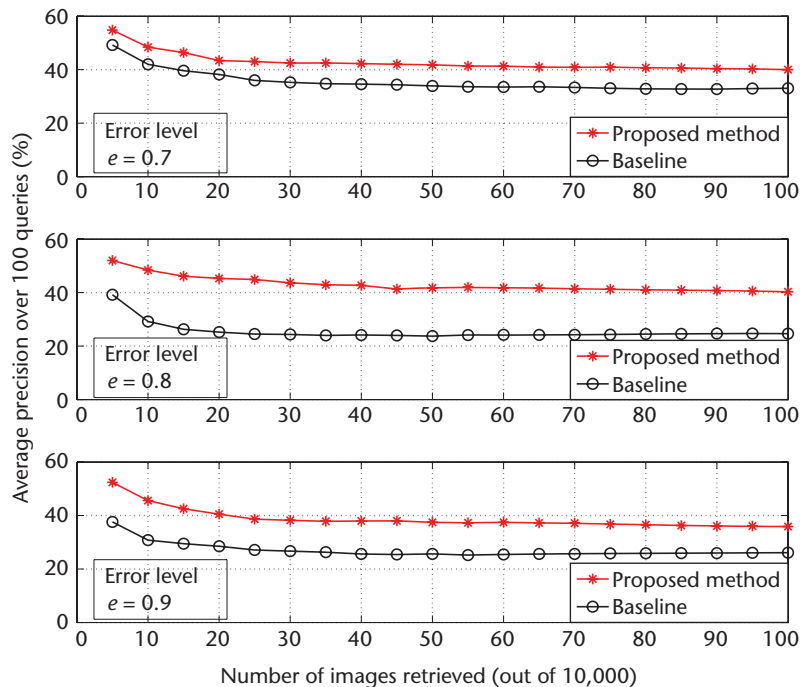


Figure 9. Retrieval precision for scenario 4 at three noise levels.

improve search performance over a strategy that doesn't involve statistical learning.

Scenario 4. Here, the query picture and the database are both fully tagged, but many tags are incorrect. This is a situation that often arises under user-driven tagging because of reasons such as subjectivity. We use our reannotation approach to refine these noisy tags prior to performing retrieval. Introducing noise levels of $e = 0.7, 0.8,$ and 0.9 and using parameter $Z = 1.5$, we test 100 random picture queries on the 10,000 images. For this, queries and the database are first reannotated.

The alternate strategy includes the baseline case, as we described in the "Reannotating noisy tags" section. Figure 9 shows the precision results over the top 100 retrieved pictures for the three noise levels. Interestingly, even at $e = 0.7$, where our reannotation approach doesn't surpass the baseline in annotation precision, it does so in retrieval precision, making it a useful approach at this noise level. Moreover, the difference with the baseline is maximum at noise level 0.8. Note that for $e \leq 0.5$, our approach didn't yield a better performance than the baseline, since the tags were sufficiently clean. These results suggest that at relatively high noise levels, our reannotation approach can lead to significantly improved image retrieval performance compared to the baseline.

Conclusion

The framework for our novel annotation-driven image search is standard for different scenarios and different types of queries, which should make implementation fairly straightforward. In each scenario we discussed, our approach yields more promising results than traditional methods. In fact, the categorization performance in itself improves on previous attempts. Moreover, we are able to categorize and tag the pictures in a very short time. All of these factors make our approach attractive for real-world implementation. **MM**

Acknowledgments

The research is supported in part by the US National Science Foundation, grants 0347148 and 0202007. We thank David M. Pennock at Yahoo! for providing some test images. We would also like to acknowledge the comments and constructive suggestions from reviewers.

References

1. K. Barnard et al., "Matching Words and Pictures," *J. Machine Learning Research*, vol. 3, 2003, pp. 1107-1135.
2. E. Chang et al., "CBSA: Content-Based Soft Annotation for Multimodal Image Retrieval Using Bayes Point Machines," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 1, 2003, pp. 26-38.
3. J. Li and J.Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 19, 2003, pp. 1075-1088.
4. F. Monay and D. Gatica-Perez, "On Image Auto-Annotation with Latent Space Models," *Proc. ACM Int'l Conf. Multimedia*, ACM Press, 2003, pp. 275-278.
5. G. Miller, "WordNet: A Lexical Database for English," *Comm. ACM*, vol. 38, no. 11, 1995, pp. 39-41.
6. R. Datta, J. Li, and J.Z. Wang, "Content-Based Image Retrieval: Approaches and Trends of the New Age," *Proc. ACM SIGMM Int'l Workshop Multimedia Information Retrieval*, ACM Press, 2005, pp. 253-262.
7. A.W. Smeulders et al., "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, 2000, pp. 1349-1380.
8. J.Z. Wang, J. Li, and G. Wiederhold, "SIMPLcity: Semantics-Sensitive Integrated Matching for Picture Libraries," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, 2001, pp. 947-963.
9. C.A. Bouman, "Cluster: An Unsupervised Algorithm for Modeling Gaussian Mixtures"; <http://dynamo.ecn.purdue.edu/~bouman/software/cluster>.
10. C. Leacock and M. Chodorow, "Combining Local

Context and WordNet Similarity for Word Sense Identification," *WordNet: An Electronic Lexical Database*, C. Fellbaum, ed., MIT Press, 1998, pp. 265-283.

11. Y. Jin et al., "Image Annotations By Combining Multiple Evidence and WordNet," *Proc. ACM Int'l Conf. Multimedia*, ACM Press, 2005, pp. 706-715.
12. R. Datta et al., "Toward Bridging the Annotation-Retrieval Gap in Image Search by a Generative Modeling Approach," *Proc. ACM Int'l Conf. Multimedia*, ACM Press, 2006, pp. 977-986.
13. J. Li, "A Mutual Semantic Endorsement Approach to Image Retrieval and Context Provision," *Proc. ACM SIGMM Int'l Workshop Multimedia Information Retrieval*, ACM Press, 2005, pp. 173-182.
14. T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, 1994, pp. 66-75.
15. R. Marée, "Random Subwindows for Robust Image Classification," *Proc. IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, IEEE CS Press, vol. 1, 2005, pp. 34-40.
16. Y. Chen and J.Z. Wang, "Image Categorization by Learning and Reasoning with Regions," *J. Machine Learning Research*, vol. 5, 2004, pp. 913-939.
17. Y. Li, Y. Du, and X. Lin, "Kernel-Based Multifactor Analysis for Image Synthesis and Recognition," *Proc. Int'l Conf. Computer Vision (ICCV)*, IEEE CS Press, pp. 114-119.
18. J. Li and J.Z. Wang, "Real-Time Computerized Annotation of Pictures," *Proc. ACM Int'l Conf. Multimedia*, ACM Press, 2006, pp. 911-920.



Ritendra Datta is a PhD candidate in computer science and engineering at The Pennsylvania State University. His research interests include statistical learning, computer vision, computational aesthetics, multimedia, and information retrieval. He has a BE in information technology from the Bengal Engineering and Science University, Shibpur, India. He is a recipient of the Glenn Singley Memorial Graduate Fellowship in Engineering.



Weina Ge is a PhD candidate and research assistant in the Computer Science and Engineering Department at The Pennsylvania State University. Her research interests include computer vision, image retrieval, and data mining. She has a BS in computer science from Zhejiang University, China.



Jia Li is an associate professor of statistics at The Pennsylvania State University. Her research interests include statistical learning, data mining, and image processing, retrieval, and annotation. She has a PhD in electrical engineering from Stanford University.



James Z. Wang is an associate professor at The Pennsylvania State University. His research interests include semantics-sensitive image retrieval, linguistic indexing, biomedical informatics, story picturing, and computational aesthetics. He has a master's degree in both mathematics and computer science, as well as a PhD in medical information sciences, all from Stanford University.

Readers may contact Ritendra Datta at datta@cse.psu.edu.

For further information on this or any other computing topic, please visit our Digital Library at <http://www.computer.org/publications/dlib>.

Submit your ideas
and videos to the
IEEE MultiMedia video blog!

Visit
<http://computer.org/multimedia>
for more details