PLSA ON LARGE SCALE IMAGE DATABASES

Rainer Lienhart
Multimedia Computing Lab
University of Augsburg
Augsburg, Germany

Malcolm Slaney
Yahoo! Research
Santa Clara, CA 95054
USA

ABSTRACT

The web and image repositories such as Fickr™ are the largest image databases in the world. There are billions of images on the web, and hundreds of million high-quality images in image repositories. Currently, these images are indexed based on manually-entered tags and individual and group usage patterns. In this work we explore a third information dimension: image features. We explore probabilistic latent semantic analysis (pLSA) in order to infer which visual patterns describe each object. We build models that connect words and image features, and use content features and tags to find similar images. We demonstrate that image features using gray-scale salient points and an aspect model based on pLSA outperforms a conventional word-frequency model as well as refined color-histrogram approach on an image-similarity task.

Index Terms— large scale image retrieval, probabilistic semantic analysis, color coherence vectors.

1. INTRODUCTION

The usage of *Probabilistic Latent Semantic Analysis* (*pLSA*) [4] — a statistical technique to derive hidden concepts from data — has recently become very popular in the image domain. So far, pLSA has only been applied to relatively small, carefully selected image databases ranging from a few hundred to a few thousand images [2][8]. In this paper we study pLSA on a large-scale, real-world image database for improving image retrieval based on image similarity as perceived by humans. Our work centers around finding visual "words" that are typical for the various kinds of aspects an image can show.

One of the largest image repositories on the web is Flickr™. For this work we have download 253,460 images that were tagged with at least one out of the 23 tags listed in Table 1[1]. These words where grouped into 12 categories for our image-retrieval task. The resulting image database was not cleaned nor pre-processed in any way to increase consistency.

Since these tags are provided by the creators of the pictures with unknown intensions, the techniques we investigate must be able to tolerate — from a pure visual similarity standpoint — a significant fraction of incorrect labels. A good example, for instance, are the images tagged "Christmas" in Flickr. Only a very small fraction of the images depict a religious event (as one might expect). Instead the tag mostly denotes the time and date of creation. Thus thousands of vacation and party photos pop up with no real common theme. The ambiguity of tags makes image retrieval more difficult.

On this real-world database we explore two questions:

(1) Does it matters how visual words are created? We compare three different techniques: (a) random selection, (b) clustering random subsets, and (c) clustering tag-based subsets.

(2) Does pLSA outperform a simple word-occurrence statistic? How does pLSA on grayscale SIFT [5] features compare to well-known global color-retrieval techniques such as color-coherence vectors [9]?

Category #	OR list of tags	# of image
1	wildlife animal animals cat cats	30476
2	dog dogs	26119
3	bird birds	21279
4	flower flowers	28816
5	graffiti	22318
6	sign signs	14488
7	surf surfing	29998
8	night	33999
9	food	19582
10	building buildings	17303
11	goldengate goldengatebridge	24362
12	baseball	12390
	Total # of Images (Note images may have multiple tags)	253,460

Table 1: The image database and its 12 categories

[1] These images were selected from all public Flickr images uploaded prior to 8 Sep. 2006 and labeled with one of the following tags: sanfrancisco, beach, tokyo and geotagged.

We evaluate these different retrieval configurations purely based on image similarity as perceived by a number of users without any special context knowledge.

2. DERIVING VISUAL WORDS

pLSA was originally derived in the context of document retrieval, where words are the elementary parts of a document. For images — our visual documents — we need comparable elementary parts we call *visual words*.

In this work we use the popular SIFT features [5] to find salient visual parts in each image. SIFT features are calculated in a two-step process: First, a sparse set of salient areas in an image are determined and described by position, scale, and orientation. Then for each salient point we derive a 128-dimensional edge-based feature vector to describe the unique grayscale content of that salient area in a scale- and orientation-invariant manner.

Since SIFT feature vectors can take on almost every value in $\Re^{128}$, we wish to find a small set of representative feature vectors to become our visual words. Thus the problem of deriving visual words is as follows:

Given

- a set of images $\boldsymbol{I}=\{d_i\}$ with $|\boldsymbol{I}| = I_N =$ # of images
- a set of feature $\boldsymbol{F}=\{f_l\}$ with $|\boldsymbol{F}| = F_{\mathrm{N}} =$ # of features (here 128-dim. SIFT features) derived from I_N images
- a set $\boldsymbol{C} = \{c_r\}$ of image categories (see Table 1) with $|\boldsymbol{C}| = C_N$ categories in total (here C_N =12)

derive a vocabulary $\boldsymbol{V} = \{\mathrm{v_j}\}$ of $|\boldsymbol{V}| = V_N$ visual words.

Finding the structure in such a large set of data (millions of images, thousands of salient points per image) is computationally expensive. We investigate three ways to determine the V_N visual words and we will evaluate their utility later in this paper:

(v1) Random: Select all V_N sample features randomly from the set $\boldsymbol{F}$ of all features.

(v2) K-means clustering (with subselection): Randomly select S_N sample features from the set $\boldsymbol{F}$ of all features. Apply K-means clustering to each set of S_N samples to derive (V_N /C_N) visual words. Perform this subselection C_N times. In total this will result in $\mathrm{C_N}$ * (V_N /C_N) = V_N visual words.

(v3) Tag subselection: For each of the C_N categories derive (V_N/C_N) visual words by means of K-means clustering by randomly sampling S_N sample features from images in each category only. In total this will result in $\mathrm{C_N}$ * (V_N /C_N) = V_N visual words.

Method (v2) is the approach commonly used in image retrieval [2][8]. Since K-means clustering is computationally expensive (quadratic in the number of samples and the number of clusters), it is more efficient to break up (C_N * S_N) samples into C_N subsets of S_N samples and find V_N/C_N clusters from this subset instead of determining all V_N clusters on the entire set of (C_N * S_N) samples directly. For our 12 categories (C_N = 12) the speedup is C_N * C_N = 144 times.

The rational behind method (v3) is to explore whether the tags in the database provide useful information for deriving visual words. Within each category the images should be less diverse and thus make it easier for K-means clustering to find the dominant visual words. The better the visual words, the better pLSA should work and thus improve retrieval. Concepts that have no representative visual words cannot be learned.

The random method (v1) is added to answer the question whether K-means clustering is necessary at all. The answer to this question has a few important implications: Firstly, clustering is often the slowest part of the learning algorithm. If it can be skipped without harm, it would greatly reduce the computational complexity. Secondly, if clustering is not necessary, the set of visual vocabulary can easily be extended any time needed by additional random samples.

In each experiment we derived 12 * 200 = 2400 visual words that are used to describe each image in our database. In Section 4 we will compare these three methods based on their similarity retrieval results in user studies.

3. MEASURING IMAGE SIMILARITY

3.1 Term-Document Matrix

Using the visual vocabulary $\boldsymbol{V}$, each feature f_l of $\boldsymbol{F}$ can be quantized by its most similar feature vector in $\boldsymbol{V}$. Thus we represent each image d_i as an image document consisting of L instances of the visual words $\{w_1,\ldots,w_L\}$, w_p□$\boldsymbol{V}$.

Given the collection of I_N image documents $\boldsymbol{I}=\{d_i\}$ with F_{N} visual words $\boldsymbol{W} = \{w_j\}$ from the vocabulary $\boldsymbol{V}$ and given that we ignore the sequential ordering of the word occurrences in the images (the so-called *bag-of-word model*), the image data can be summarized by an $I_N \times V_N$ matrix of visual word occurrence counts N = $(n(d_i,v_j))_{\mathrm{ij}}$, where $n(d_i,v_j)$ specifies the number of times the word v_j occurred in document $d_{\ i}$. The resulting table is called the term-document matrix (see Figure 1).

Note by normalizing each document vector to 1 using the L1-norm, the document vector of d_i becomes the estimated mass probability distribution $P(v_j\ |d_i)$.

The similarity between two documents can be calculated using the cosine metric between two document vectors $\mathbf{a}$=d_i

and **b**=d_p. The *cosine metric* between to vector **a** and **b** is defined as

$$CSMetric(\mathbf{a},\mathbf{b}) = \frac{<\mathbf{a},\mathbf{b}>}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}$$

It is commonly used in text retrieval [1].

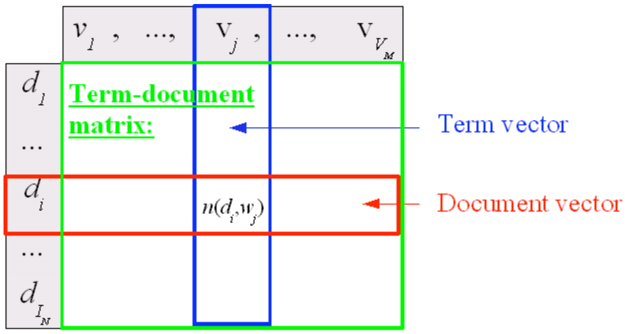

Figure 1: Term-document matrix

3.2 pLSA

Each L1-normalized row in the term-document matrix describes the distribution of the visual words in each document, i.e., $P(v_j|d_i)$.

The idea of pLSA is to introduce a mediator known as *aspects* or *concepts* between the document and the words. Thus, every word occurring in a document is generated by an unobservable aspect variable z_k leading to the following generative model for the document vector [4]:

(1) Pick a document d_i with prior probability $P(d_i)$
(2) Select a latent concept z_k with probability $P(z_k|d_i)$
(3) Generate a word v_j with probability $P(v_j|z_k)$

An important aspect of this model is that word occurrences are conditionally independent from the document given the unobservable aspects. Thus

$$P(v_j \mid d_i) = P(d_i) \sum_{k=1}^{K} P(v_j \mid z_k) P(z_k \mid d_i) .$$

In addition, every document is modeled as consisting of one or more aspects. This is very natural since images consist of multiple objects and thus multiple aspects in different image areas. pLSA can model this fact very efficiently. For instance, an image with a lion and jeep — each object described by a set of SIFT features — might be described by two hidden aspects 'lion' and 'jeep'. Dependent on the aspects the probabilities of each visual word v_j is different.

We learn the unobservable probability distributions $P(z_k|d_i)$ and $P(v_j|z_k)$ from the data using the Expectation-Maximization-Algorithm (EM-Algorithm) [3][4]:

E-Step:

$$P(z_k \mid d_i, v_j) = \frac{P(v_j \mid z_k) P(z_k \mid d_i)}{\sum_{l=1}^{K} P(v_j \mid z_l) P(z_l \mid d_i)}$$

M-Step:

$$P(v_j \mid z_k) = \frac{\sum_{i=1}^{N} n(d_i, v_j) P(z_k \mid d_i, v_j)}{\sum_{m=1}^{M} \sum_{i=1}^{N} n(d_i, v_j) P(z_k \mid d_i, v_j)}$$

$$P(z_k \mid d_i) = \frac{\sum_{j=1}^{M} n(d_i, v_j) P(z_k \mid d_i, v_j)}{n(d_i)}$$

Given a new test image d_{test}, we estimate the aspect probabilities, similar to above, from the observed words. The only difference is that the learned conditional word distributions $P(v_j|z_k)$ are never updated.

The similarity between two documents is calculated using the cosine metric between two the two aspect vectors **a**= $(P(z_k|d_i))_k$ and **b**= $(P(z_k|d_p))_k$. We model the collection of visual words with 48 aspects in total — analogous to a 48-mixture Gaussian mixture model.

3.3 Color Coherence Vectors

As a baseline for comparison we use one of the best traditional global color features: Color Coherence Vectors (CCVs) [9]. It is computed by first quantizing each pixel's color by using the 2 most significant bits per color channel, resulting in only 64 possible different color values. Then, for each pixel we measure the area connected (with an 8-neighborhood) of the same quantized color. If the area is above a threshold (usually 1% of the pixel count in the image), then the pixel is added to the "coherent" histogram, otherwise to the "incoherent" color histogram. Combining both histograms results in a 128-dimensional vector. Dissimilarity between two CCV vectors **a** and **b** is computed based on the L1-norm.

4. EXPERIMENTAL RESULTS

Performance Metric: For evaluation we selected *randomly* 5 query images from each of our 12 categories, i.e., 60 query images in total. Then, for each query image each retrieval technique was used to return the top 20 most similar images. In each of the three experiments below, three rival techniques were compared based on the judgments of a number of users: For each query, the retrieval results (top 20 images, tiled 5 by 4 on a sheet of paper) for the three techniques under comparison were shown to the user. The user had to put the results from each query image into an order from best to worst retrieval result. The technique with the best retrieval result received 2 points, the second best 1 point, and the worst 0 points. We computed the average score for each technique over the 60 samples queries to assign a single performance number. The technique with the highest score obviously performs best.

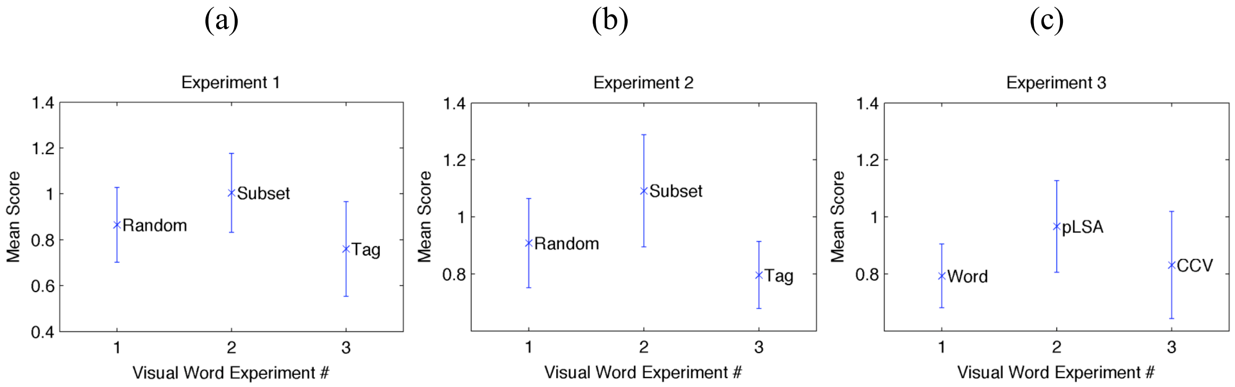

Figure 2: Results from the three experiments: a) cosine similarity on word-histogram feature, b) cosine similarity on pLSA, and c) comparing best cosine methods with CCV baseline algorithm.

Exp. 1: In this experiment we compared the three visual word extraction techniques (v1), (v2), and (v3) against each other by using the document vectors from the term-document matrix with the cosine metric for similarity retrieval. Figure 2a shows the average scores for 8 different subjects.

Exp. 2: In this experiment we compared the three visual word extraction techniques (v1), (v2), and (v3) against each other by using the aspect vectors of the image documents with the cosine metric for similarity retrieval. Figure 2b shows the average scores for 8 different subjects.

In both of these experiments deriving visual words using plain clustering produced the best results. Selecting visual words completely at random is computationally cheap, and should work well asymptotically, but not evidently at this level. We are surprised that deriving specific visual words based on category subsets did not produce an overall benefit, but an informal analysis suggests that these category-specific words helped for a category like dogs.

Exp. 3: In our final experiment we compared the random-subset visual word selection approaches that won from Exp.1 (cosine metric of word histograms) and Exp. 2 (cosine metric of pLSA histograms) to a baseline using CCV features. This test is difficult for subjects because in such a large database the matches in a color space are at first glance identical to the query. It is only when the picture is studied does one realizes that the objects shown are so different. This is especially true when we look color similarity with our full 2.5M image database.

The results of this test are shown in Figure 2c. Seven subjects judged that images found by using a cosine metric in pLSA space are more similar to the query image than a direct comparison in word space, or the baseline CCV approach. Much like it does in text-based retrieval, calculating similarity in subspace formed by the aspect model gives better results.

5. CONCLUSION

In this paper we have shown that the aspect model, using an approach like pLSA, is as important for image-retrieval as it is for text-retrieval [1]. The aspect model learns the probability of each visual word given an unobserved aspect. We have extended Bosch's work [2] by showing that pLSA improves performance on a similarity task. The dimensionality reduction due to an aspect model is important as we go to larger databases.

In future work we want to verify our results with a larger number of subjects, and we want to test the similarity on the full 2.5M image database.

References

[1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.

[2] A. Bosch, A. Zisserman and X. Munoz. Scene Classification via pLSA. Proceedings of the European Conference on Computer Vision (2006).

[3] Dempster, A. P., N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Loyal Statistical Society*,B.39, 1977.

[4] Thomas Hoffmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, Vol. 42, Issue 1–2, pp. 177–196, 2001.

[5] D. Lowe. Distinctive image features from scale invariant keypoints. In *IJCV* 60(2):91–110, 2004.

[6] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool. A comparison of affine region detectors. In *IJCV* 65(1/2):43–72, 2005.

[7] K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. In *PAMI* 27(10):1615–1630.

[8] P. Quelhas, F. Monay, J.-M Odobez, D. Gatica-Perez, T. Tuytelaars, L. Van Gool. Modeling scenes with local descriptors and latent aspects. ICCV 2005, Vol. 1, pp. 883–890, Oct. 2005.

[9] G. Pass, R. Zabih, and J. Miller. Comparing Images Using Color Coherence Vectors. In *Proc. of the 4th ACM Int. Conf. on Multimedia,* Boston, MA , pages 65–73, 1996.