LAPLACIAN SIFT IN VISUAL SEARCH

Xin Xin[1], Zhu Li[2] and Aggelos K. Katsaggelos[1]

[1]Dept of EECS, Northwestern University, Evanston, IL, USA
[2]Media Networking Lab, FutureWei (Huawei) Technology, Bridgewater, NJ, USA

ABSTRACT

With the explosive growth of video capture capable mobile handsets and online visual data repositories, the popular query-by-capture applications call for a compact visual descriptor with minimum descriptor length. How to preserve the visual descriptor information while minimizing the bit rate for representing the descriptors, is a focus of the on-going MPEG7 CDVS (Compact Descriptor for Visual Search) standardization effort. In this work, we present a SIFT descriptor dimension reduction scheme based on Laplacian Graph Embedding, which computes a linear embedding that preserves the topological relationship among visual descriptors. Simulation results demonstrate the effectiveness of the proposed solution at low bit rates.

Index Terms—SIFT, Visual Search, Scalability, Graph Laplacian

1. INTRODUCTION

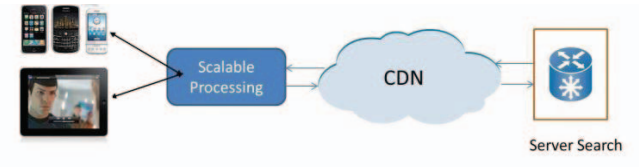

Fig. 1 Scalable image and video service architecture.

Mobile handsets have become very popular during the recent years. With the explosive growth of mobile handsets, like android, iPhone and PDA, mobile based multimedia services are enjoying intense innovation and development. Application scenarios of mobile services can be location based services [1], logo search, and so on. Examples of deployments of mobile services are Google Goggles (www.google.com/mobile/goggles/), Nokia Point & Find (pointandfind.nokia.com/) and Snaptell (snaptell.com/). The various products of visual search design and use their own visual features. These visual search products are made possible with robust image features like SIFT [2] and SURF [3] and search technologies like bag of words (BoW) methods [4, 5] and subspace methods. With the BoW method, an image is represented as an index list of quantized image features; comparing then two images is performed though the comparison of their index lists. With subspace methods, the dimension of each image feature is first reduced and then indexed with a kd-tree; searching is done sequentially by searching the kd-tree leaf nodes. Subspace methods are easier to be implemented in scalable mode. We therefore choose to use subspace methods in this work.

Visual search has unique challenges. We can either send the entire image or sets of feature representations to the server. Sending a JPEG image requires a relatively large bandwidth, while sending sets of features only consumes very limited bandwidth. A lot of work has been focusing on how to minimize the descriptor length while acquiring higher retrieval performance. Still, we do not have a standard to cope with all these tasks. So, currently, there is an on-going effort with MPEG7 named Compact Descriptors for Visual Search (CDVS). MPEG is planning standardizing technologies that will enable efficient and interoperable design of visual search applications. In particular they are seeking technologies for visual content matching in images or video. Visual content matching includes matching of views of objects, landmarks, and printed documents that is robust to partial occlusions as well as changes in vantage point, camera parameters, and lighting conditions. Several works have been submitted to MPEG7 CDVS including ours.

Some works have been focusing on compact features. PCA-SIFT is an early effort for compact feature representation. With PCA-SIFT, a 3042-element normalized image gradient vector is created and projected into our feature space using the stored Eigenspace. As shown in [6], retrieval accuracy becomes low with 12 dimensions, which still represents a lot of data for each dimension. Besides that, in [7], a 60-bit representation was presented with ChoG, which is a feature descriptor with a tree structure that can be compressed. Their results show very low error matching rate at 60 bit representation.

In this paper, we consider a more compact feature representation. In visual search applications, the matching ability of the descriptors of the query image to the descriptors of the database image is critical to successful matching. However, reducing the descriptor length with global methods like PCA and quantization is not

satisfactory, because it preserves information for the re-construction of the descriptor, rather than preserving the nearest neighbor (NN) topological relationship among the descriptors, which is more crucial for visual search applications. Therefore, we propose to use graph embedding [8] to learn a more compact representation of SIFT features.

This solution tries to preserve descriptor neighborhood information by minimizing a penalty function on descriptor matching. The resulting compact feature representation can achieve high matching accuracy with only 32-bit per descriptor. Besides, we propose to select only sub-sets of features to save even more bandwidth.

An added advantage of Laplacian SIFT is to offer scalability in bit-rate vs matching accuracy, i.e., offer scalable query by capture services by selecting sub-sets of image features and reducing feature dimensions by learning feature spaces. As shown in Fig. 1, mobile users are requesting certain image and video services. Client end would extract image features (e.g., SURF) and the scalable processing unit would reduce the dimensionality and number of features. Then, with wireless networks and CDN, client requests are transmitted to server end which will process the search and return the result back to the client end.

The paper is organized into the following sections. Section 2 develops the problem formulation of Laplacian-SIFT embedding for descriptor compression. Section 3 develops the ideas on providing scalability in visual search. Simulations are presented in section 4. In section 5, we draw conclusion and outline our future work.

2. PROBLEM FORMULATION

Dimensionality reduction techniques like Principal Component Analysis (PCA) have been extensively applied in image search applications. PCA is used to reduce dimensions while preserving most of the variance of the dataset. In visual search applications, the matching ability of the descriptors from the query image to the database image is critical to successful matching. However the objective of PCA is to preserve the information for minimum error reconstruction, while for visual search, we need to preserve the nearest neighbor relationship among visual descriptors. Our formulations try to preserve this nearest neighbor relationship. So from this point of view, Laplacian embedding should be more suitable to visual search. The problem is therefore formulated as a Laplacian embedding problem [8].

The topological relationship among a given set of visual descriptors can be represented by its affinity matrix. Let the SIFT features for a dataset be represented as $X = [x_1, x_2, \ldots, x_N]$, $x_i \in \mathbb{R}^m$, where N is the sample number and m the feature dimension, which is equal to 128 for SIFT features. Let $G = \{X, W\}$ be a weighted graph with vertex set X and similarity matrix W, which is defined as:

$$W_{ij} = \begin{cases} 0, & |x_i - x_j| > T \\ e^{\frac{-(x_i - x_j)^2}{2\sigma^2}}, & |x_i - x_j| \le T \end{cases} \quad (1)$$

where σ and T are the kernel parameter and the threshold for determining neighbor relations, respectively. Threshold T controls the sparsity of the similarity matrix. As T increases, we actually try to preserve more of the neighborhood information. These two parameters have influence over the subspace model. The linear compression model is $y = X^T A$, where A can be obtained from graph embedding [8]. It is given by

$$A^* = arg\ min_{A^T X B X^T A = d} \sum \| A^T x_i - A^T x_j \|^2 W_{ij} \quad (2)$$

This expression can be simplified with the Laplacian function:

$$A^* = arg\ min_{A^T X B X^T A = d} A^T X L X^T A \quad (3)$$

This representation can be solved as a generalized eigenvalue decomposition problem.

$$\tilde{L} v = \lambda \tilde{B} v$$

where $\tilde{L} = XLX^T$ and $\tilde{B} = XBX^T$.

We performed this analysis with the 128-dimensional SIFT features. Intuitively, by keeping more dimensions, higher retrieval performance can be achieved at the expense of increased communication overhead. In scenarios of mobile image search, if we transmit only 10-dimensional features instead of 128-dimensional features, we can roughly save 92% of the bandwidth. In this work, we found that a large reduction in feature dimensionality can still guarantee a robust search performance and we can avoid most of the communication overhead. In our work, different dimensions of Laplacian SIFT are evaluated on our datasets.

3. SCALABILITY VIA LAPALCIAN EMBEDDING

Different visual search tasks may require different quality of service. For example, if one wants to match a simple intel logo, 3-dimensional features and 30 features may be adequate. While if a complex building is to be searched, we may need more dimensions and a larger number of features. We therefore provide scalability towards mobile visual search. In this section, we discuss two types of scalability, i.e., feature dimension scalability and feature number scalability.

3.1 Feature Dimension Scalability

Features can be transformed to a lower dimension with $y = X^T w$, where X is an $n \times 1$ original feature, w is an $n \times d$ transform matrix, and y is the $d \times 1$ transformed

feature. Controlling the transform dimension d would result in a different dimension scalability.

3.2 Feature Number Scalability

Typically, images of size 1024 * 768 pixels can have up to 1500 features. In practice, however, only a subset of the features are needed for successful image retrieval. So, we propose the following two different ways to select a subset of features.

1. Quad Tree Selection: First, we obtain the centroid of all keypoints; then, we partition the whole image region into four regions the same way as in quad-tree partition. In each sub-region, key points are randomly selected.
2. Random Selection: Image key points are randomly selected from all key point sets.

Quad-tree selection guarantees that we can get almost the same numbers of features in each sub-region, while random selection could be unbalanced in different regions. More complex schemes, like growing a minimum spanning tree, can be performed to select feature points. Experiments are performed on our datasets to compare these two methods.

4. SIMULATION

4.3. Dataset Description

We perform our experiments on both the Northwestern/Huawei (NU/HU) and HK datasets. The NU/HU dataset has 1000 images with 100 categories, each of 10 images. The HK dataset has 2000 images with 200 categories, each of 10 images. Some sample images are shown in Fig. 2. Summaries about the NU/HU and HK datasets are provided in table 1

	Categories	Total Nums	Original Resolution	Description
NU/HUdataset	100	1000	1024 * 768	Buildings at Northwestern University
HK dataset	200	2000	640 * 480	Buildings in Hong Kong

Table. 1 Summaries about NU/HU and HK datasets

Fig. 2 Sample images of the KH dataset

In this proposed architecture, a subspace model is first applied on the whole dataset. Then, we build two kd-trees in case some of the feature points are located at the boundary of two leaf nodes. With multiple kd-trees we may be able to get rid of feature points in leaf node boundaries at the expense of increased computations. At each leaf node, we perform nearest neighbor matching to infer whether there is a matching. The details of the algorithm are given below

Data Preprocessing:

1. For feature points $x_i\ i = 1 \dots n$, do graph Laplacian embedding [3], find transform matrix A, that transforms x according to the equation: $y_i = A\ x_i$, where $A\ is\ m \times 128$, and m represents the desired feature dimension.
2. Build two kd-trees (kd_1, kd_2) on transformed feature points $\{y_i | i = 1 \dots n\}$ from different starting dimensions.

Query Processing:

1. For query feature points $y_{qi}, i = 1 \dots n_q$, where n_q is the number of features for query image q, select features according to one of the two presented feature selection algorithms. Search through the kd-trees and locate leaf nodes $leaf_1, leaf_2$ in kd_1, kd_2, respectively.
2. Merge the two leaf nodes $leaf_1, leaf_2$ to $leaf$
3. For all feature points in leaf node, calculate distance $DIS(.)$ with feature y_{qi} and sort the distances.
4. For all features with distances smaller than T_{dis}, add one point to the image containing the corresponding feature.

After step 2, sort image score and find relevant images by thresholding scores over T_{score}.

The result is evaluated with MAP score, defined as:

Rank: position of an image in the list of retrieved images

Precision at a given cut-off rank r for a single query: P(r) = (number of relevant images of rank r or less) / r

Average precision: defined as follows

$$AveP = \frac{1}{R}\sum_{r=1}^{N} P(r)\,\mathrm{rel}(r)$$

where N is the number of retrieved images, R is the number of relevant images, and rel(r) = 1 if image at rank r is relevant, 0 otherwise.

Mean average precision: average precision for a set of queries is defined as follows:

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AveP(q)$$

where Q is the number of queries.

4.4. Retrieval Performance on different dimensions at 8 bit per dimension representation

One experiment is performed on the NU/HU and HK datasets, with 45,994,130 and 67,694,200 features, respectively. We build kd-trees with height 12. So each leaf

node has approximately 500 feature points. Euclidean similarity is chosen to match feature points in leaf nodes because it is shown to be better than Cosine distance in this application scenario. It is defined as

$$dis(a,b) = \text{norm}(\text{a} - \text{b})$$

When T_{dis} is too large, many unwanted candidates will generate noise. When T_{dis} is too small, many wanted candidates are discarded.

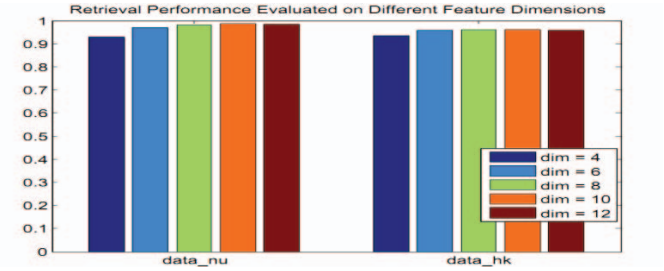

Fig. 3 Retrieval Performance evaluated on different feature dimensions

Quantization is performed at 8 bit per dimension. One result is shown in Fig. 3. With only a 4-dim feature that is 93.75% compression of data, we can still obtain 92.8% retrieval performance on the NU/HU data set and 93.5% retrieval performance on the HK data set. This result shows that features are very compressible. With only a very small number of features, we can still obtain very good retrieval performance. For this dataset, the proposed Laplacian SIFT feature performed well with both a 32 bit and a 48 bit representations, which is more compact than the 60 bit representation in [6].

4.5. Retrieval Performance Evaluated on Feature Selections.

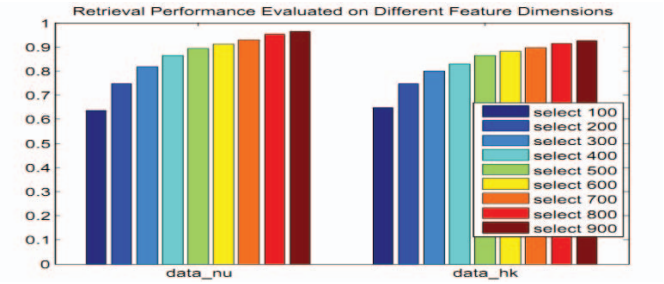

Fig. 4 Retrieval Performance evaluated on feature number selection

One sample result is shown in Fig. 4. With only 400 features, we can still maintain 88.6% retrieval performance on the NU/HU data set and 87.8% retrieval performance on the HK data set. This result shows that with only a very small number of features, we can still achieve very good retrieval performance.

5. CONCLUSION AND FUTURE WORK

In this work, we propose a compact SIFT feature representation to perform image retrieval. This framework is not limited to SIFT features and can also be applied with other features like SURF features. This method learns the geometric relationship in feature space and reduces feature dimensions by maintaining the geometric relationship of features. With Laplacian SIFT, we propose a 32 bit representation by keeping only 4 dimensions and an 8 bit quantization for each dimension. This representation performs well on the NU/HU and HK datasets with a MAP score of 92.8% and 93.5%, respectively. Furthermore, we propose to select a subset of image features to perform retrieval. With half of the features, we can still obtain high retrieval performance. In the future, we will test these compact features on additional datasets and consider additional applications, such as video copy detection and image near duplicate detection. Besides, we will combine this work with geometric verification to achieve better retrieval performance.

6. REFERENCES

[1] Z. Ye, X. Chen, and Z. Li. "Video based mobile location search with large set of SIFT points in cloud", *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing* (MCMC '10), ACM, New York, USA.

[2] D.G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision,* 60, pp. 91-110, 2004.

[3] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, pp. 346--359, 2008

[4] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos", *Proceedings of the International Conference on Computer Vision* , 2003.

[5] X. Xin and A.K. Katsaggelos, "A Novel Image Retrieval Framework Exploring Inter Cluster Distance", *Int. Conf. Image Processing*, September 2010.

[6] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", *Computer Vision and Pattern Recognition*, 2004

[7] B. Girod, V. Chandrasekhar, R. Grzeszczuk, Y. Reznik, "Mobile Visual Search: Architectures, Technologies, and Emerging MPEG Standard", *IEEE Multimedia*.

[8] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction", *IEEE Trans. Pattern Anal. Mach.* Intell. 2007.