EXPLOITING FEATURE CORRESPONDENCE CONSTRAINTS FOR IMAGE RECOGNITION

Linbo Wang[1] *Feng Tang*[2] *Yanwen Guo*[1] * *SukHwan Lim*[2] *Nelson L.Chang*[2]

[1]State Key Lab for Novel Software Technology
Nanjing University, China PR
wanglb.2005@gmail.com, ywguo@nju.edu.cn

[2]Hewlett-Packard Labs Palo Alto
CA 94304 USA
{feng.tang, suk-hwan.lim, nelson.chang}@hp.com

ABSTRACT

Image recognition is one of the fundamental problems in multimedia analysis. Typically in the training database, there will be more than one image for each object, however most existing bag-of-features based approaches treat them independently and completely ignore the feature correspondence relationship among them. As a result, features corresponding to the same physical point may be clustered into different clusters, which finally leads to inaccurate image representations for recognition. To tackle the problem, we present a supervised codebook construction algorithm exploiting the feature correspondence constraints in feature clustering. Features in different images of the same object are first matched, then homography between images are computed to remove outliers as well as recover the feature correspondences that are not correctly matched. Features belonging to the same physical point are enforced to be in the same cluster. We show via experiments that codebook constructed using this approach can improve the recognition performance.

Index Terms— Image recognition, bag-of-features

1. INTRODUCTION

Object recognition is a challenging task in computer vision and multimedia systems. Most existing recognition methods have two stages, training and testing. During training, a set of images of an object are first collected. These images are usually captured under different imaging conditions, e.g. under viewpoint change, illumination changes. Then a distinct representation is built for each photo in the database. When there is a new image to be recognized, features are extracted and matched against database representations to obtain the image identity. The most often used representation for recognition is the bag-of-features model based on invariant local features.

Sivic and Zisserman [1] was probably the first to use the "bag-of-features" model to build a visual codebook image representation for the application of image search. Visual codebook based approaches usually use robust appearance descriptors extracted from local image patches to describe the image appearance. It has been shown to give excellent performance in image retrieval, object recognition and categorization. Codebooks are usually constructed by using feature clustering algorithms such as k-means to cluster feature descriptors sampled either densely or sparsely from a set of training images. Each codebook entry which is also called a "visual word" corresponds to a cluster center. The representation of an image is obtained by building a histogram of visual word counts drawn from the codebook. The original bag-of-features representation has been extended in different ways to enhance representation power. In [2], the authors propose spatial weighted bag-of-features to exploit spatial relations between features during supervised training. In [3], a hierarchical tree is built to address the scalability issue of object recognition. In [4] randomized trees are used to quickly assign feature descriptors to visual words. Current methods assign a feature to a single cluster, this is not robust to clustering error. [5] explores techniques to map each visual region to a weighted set of words to recover the information lost in the quantization stage. Supervised discriminative codebook learning [6] has also been proposed to improve the discriminative power of the representation.

Most existing approaches treat the images in the training database independently and collect all the extracted features for clustering without considering the geometric relationship among them. In our application setting, each object has several images captured under different viewpoint. Features detected in these images are geometrically correlated with each other. The same physical point may have several feature descriptors describing the local image appearance under different viewpoints. In traditional bag-of-features model, these features are treated independently and may be clustered into different clusters. In this paper, we propose to exploit the correspondences among features of the same object and present a novel constraint based clustering approach for codebook construction. This method ensures that corresponding features are clustered into the same cluster. We show through experiments the codebook generated using this approach is more discriminative and can improve recognition performance.

*This work is supported by the National Science Foundation of China under Grants 61073098, 60723003, and 61021062, the National Fundamental Research Program of China (2010CB327903) and the Jiangsu Science Foundation (BK2009081). Yanwen Guo is also affiliated with the Jiangyin Information Technology Research Institute of Nanjing University.

The rest of this paper is organized as follows. In Section 2, we present a high-level overview of our method. The two key components of our method, exploiting the feature correspondence constraints and clustering the features using hierarchical constrained k-means are described in Section 3 and 4 separately. Experiments are given in Section 5 and the paper is concluded in Section 6.

2. OVERVIEW

Our system has a training stage and a testing stage. During training, we build a codebook using the constraint clustering method. In the training dataset, each object has several images captured under different conditions such as changing viewpoints and varied illuminations. We first extract SIFT features [7] from each image. These features are used to construct a codebook for the bag-of-features model. To exploit the geometric relationship among features, identify feature correspondences among training images via feature matching. A homography is fitted based on the matched features to remove outliers and recover correspondences that are missed. Note we assume the object is a planar surface which is true for our dataset containing CD covers and books. Thereafter, we construct the codebook with a hierarchical constraint k-means algorithm to ensure the matched feature points are clustered into the same cluster. This codebook is used to generate the bag-of-feature representation for database images.

3. GEOMETRIC FEATURE CORRESPONDENCE CONSTRAINTS

In order to construct the codebook, we first extract the SIFT features for each image in the training set. In order to compute the geometric feature correspondence constraints, a feature match and search process is carried out. Feature match tries to find the correspondences among features detected in different images. However, some features may not be able to find matches in other images due to the detector error. We use a feature search method to automatically find these missing features through the estimated homography. This can augment the features detected from the original training images.

Feature matching is usually obtained via naïve bipartite graph matching [2, 1]. This naïve approach does not consider geometric relationships among neighboring local feature points. To address this problem, we employ here a "local structure matching" algorithm [8]. The idea is that if two features match, their local structures, i.e. neighbor features, should match as well. The more matched neighboring features around the two features the more confidence on the match. The main advantage of this approach is that it effectively rejects outlier matches, while simultaneously keeping correctly matched ones intact.

Assume that the training images for a planar object are denoted by $G = \{I_|i = 1, ..., n\}$. The corresponding feature sets are $F = \{F_i|i = 1, ..., n\}$ in which F_i is the feature set extracted from image I_i. To find all the feature correspondences, a straightforward but expensive way is to match the features between each two feature sets in F. In our implementation, we instead select a feature set F_k with the largest number of features as the pivot and match it to all the other feature sets in F using the local structure matching. An example of the feature match is shown in Figure 1. Due to detector error, some features detected in other images may not be detected in the pivot image I_k, and vice versa. However, the features do have physical correspondences in I_k. We propose an approach that can fix the detector error and use a fitted homography to automatically find missing features to augment the original feature set.

Fig. 1. Example of feature matching.

Without loss of generality, assume we need to find in F_k, the corresponding features for the non-matched features $\{f_{il}^m|l = 1, ..., n_i^m\}$ in F_i. For each non-matched feature f_{il}^m, its corresponding location in F_k can be obtained by applying the homography between image I_i and I_k on f_{il}^m. We then compute the corresponding feature f_{kl}^m by extracting the SIFT descriptors[7] for all the points in a local neighborhood around the estimated location and taking the one with the minimum descriptor distance to f_{il}^m as f_{kl}^m. In order to handle occlusion or image boundary features, if the distance exceeds a given threshold, the corresponding feature is rejected. In this case, f_{il}^m does not have correspondence in F_k. Otherwise, the new feature is added to the feature set F_k. Once the feature search process is completed between F_k and each F_i, we obtain the augmented feature set F_k for I_k. This process is illustrated in Figure 2(a).

The above process enriches the features in F_k. With the new features, F_k is then backprojected to other images to locate and extract features that are not detected, as shown in Figure 2(b). The regenerated features in image F_i are shown as blue ellipses in the left image of Figure 2(b). Note the scale and orientation of these features are determined by the transformed scale and orientation of the original features. Finally, we obtain augmented feature sets for all the images.

4. FEATURE CLUSTERING WITH GEOMETRIC CONSTRAINTS

As we have established the correspondence among features corresponding to the same physical point, the corresponding features are grouped as point-sets. The correspondence relationship are taken as constraints when constructing the codebook. That means all features in the same point-set should be clustered into the same cluster. This constraint is never exploited in any previous codebook construction methods.

The feature clustering with set constraints can be formulated as follows: suppose we are given a family of point sets $S_1, S_2, ..., S_N$ in R^d space, the objective is to find k

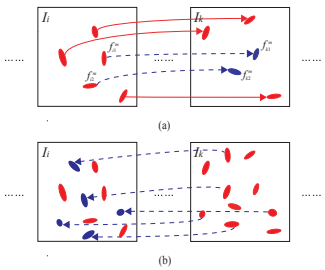

Fig. 2. Feature match and search. (a) The red(solid) arcs connect features direct matched and blue features in F_k are regenerated through feature match. (b) Once F_k is done, features in F_i is augmented with respect to the augmented F_k.

points(centers) $c_1, c_2, ..., c_k$ and an assignment of the N point sets to the k centers so that the total connection(or assignment cost) of the N point sets to their corresponding centers is minimized. The connection cost of a point-set S_i to a center c_j is the sum of the distance between each point in S_i and c_j. The point-sets clustering problem is a novel NP-hard problem [9]. In [9] the authors develop a unified approach to solve the problem. They first use a simple shape to capture the "shape" and "position" of each point-set, and then select representative points from each simple shape. The representative points are expected to "well preserve" certain crucial information, such as the position and size, of the point-sets. With the set of representative points, their approach then solves the problem using the induced metric distance function by the representative points. The authors demonstrate that the proposed approach can well approximate the exact solution. Inspired by it, we take the centroid and feature number of each feature point-set as its representative information and propose a hierarchical clustering algorithm for the formulated problem described at the beginning of the paragraph. More specifically, the basic algorithm works as follows:

1. Compute the centroid vector w_i and point number n_i of each feature point-set.
2. Initiate cluster centers using the farthest-nearest rule.
3. For each feature point-set, find its nearest cluster to its centroid and add its value $(wi * ni)$ to the cluster.
4. Calculate the centroid of each cluster. If all the computed centroids are equal to their previous centers, the best solution is obtained. Otherwise, update the cluster centers as the computed centroids and reiterate from step2 until the loop threshold is reached.

We firstly cluster the initial point-set into k groups using above algorithm, and then cluster each of the k groups to smaller feature groups in the same way. This process is repeated until enough clusters(generally 1 percent of total feature point-sets) are obtained. After clustering, each feature point is assigned to the cluster its point-set belongs to and the cluster centers are used as the visual words to generate the bag-of-features representation for each image.

5. EXPERIMENTS

To show the effectiveness of our method, we conduct quantitative experiments on two datasets. The first one is the University of Kentucky Recognition benchmark database [3]. In this dataset each object has 4 images from different viewpoints. We selected 1228 images which are mainly CD covers for experiments. Better performance is achieved on this dataset than conventional bag-of-features model. We observe that both training images and testing images in this dataset are captured in controlled environment with a very clean background. This is very different from the reality where testing images are captured at very cluttered background. To address this problem, we constructed a more challenging dataset with testing images captured with cluttered background. The new dataset consists of 1000 images, organized in groups of 4 images per object including one photo for testing captured with clutter. A large variety of objects including books, packing boxes and containers of living goods, etc are contained. Figure 3 shows some examples in the Kentucky dataset and ours.

Fig. 3. Sample images derived from Recognition benchmark database [3] (top 2 rows) and in our dataset(bottom 2 rows).

In Table 1, we show the performances of the conventional bag-of-features model and our approach on the selected Kentucky Recognition benchmark dataset(KRB1228) and our test dataset(NJUIR). These methods are evaluated using two criteria: 1) accuracy - percentage of top1 correct recognition and 2) a scoring method used in [3], which computes the average number of hits occurred in the top 3 retrieval images for each test image. Note that for both criteria on the datasets, our approach outperforms the conventional bag-of-features model, especially in the case of average top 3 recognition score. The results show that incorporating the correspondence relationships as constraints into our clustering algorithm to produce more accurate representations for images improves the performance. Our approach reinforces consistence of correctly matched features and strengthens the inter-correlation among images with the same object so that it leads to better recognition rate. For time consumption, although our approach consumes more time (about 5 minutes per group of 3 images) to establish feature correspondences during off-line training, it

takes about 0.1 second for each query on both datasets with our C++ implementation on a 2GHz pentium.

Dataset	Method	Top1 %	Average Top3 score
NJUIR	Conventional	81.2	2.264
NJUIR	Proposed	84.8	2.428
KRB1228	Conventional	91.86	2.384
KRB1228	Proposed	94.14	2.707

Table 1. Comparison results of the percentage of top1 correct retrieval and average top3 recognition score(full score: 3) for both the conventional bag-of-features model and our approach on the KRB1228 dataset and our NJUIR dataset.

In addition, we also do some statistics about how many group constraints are broken by using different clustering approaches. After feature matching and regeneration, 403977 groups are generated for our dataset. Our approach assigns no matched features to different clusters, while there are 39157 (9.7%) correspondence groups are broken after assignment using original hierarchical k-means.

To further illustrate the effectiveness of feature regeneration and constrained feature clustering, we designed four types of experiments. Method A is the original bag-of-features model using traditional hierarchical k-means. Method B applies our proposed clustering algorithm to bag-of-features model, with feature correspondence constraints identified through only feature matching. Method C generates codebook using hierarchical k-means on the updated feature sets after feature regeneration. Method D combines hierarchical constrained feature clustering with feature matching and regeneration. Method A is used as the baseline for performance comparison. Method B is designed to demonstrate the effectiveness of keeping the correspondence constraints during training by using our proposed clustering algorithm. Method C evaluates the usefulness of features regenerated based on our feature correspondence exploiting process. Method D assesses the combination power of the proposed two approaches. The parameters for each method include clustering tree level and tree branch. The best parameters are obtained by grid search. Results for all methods on our challenging dataset are plotted in Figure 4. As can be observed, applying hierarchical k-means on the feature sets updated by feature regeneration outperforms applying it on the initially extracted features. It indicates that the generated features, which are missed initially due to the feature detector error, are helpful in representing their corresponding images. Furthermore, constrained feature clustering based on feature matching also enhances the performance of conventional bag-of-features model. This is illustrated by the fact that our clustering algorithm keeps the correspondence relationships among features and results in more accurate representations of images. Finally, the combination of our constrained feature clustering algorithm with feature regeneration is more powerful and achieves even more improvements.

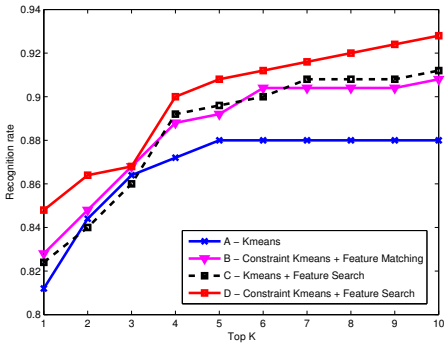

Fig. 4. Recognition performance, showing percentage (y-axis) of test images that at least one of the top x (x-axis) frames are correctly returned for different methods on our NJUIR dataset.

6. CONCLUSIONS

We presented a supervised codebook construction method that takes into account the feature correspondence constraints in clustering. Feature correspondences are obtained by matching images of the same object. Geometric relationships among these images are exploited by fitting homography, which are used to regenerate those features that are not detected due to the feature detector error. This feature search process significantly augments the feature set extracted from the original image. Different from traditional feature clustering approaches which treat features independently, we impose correspondence constraints on the clustering by making features that belong to the same physical point be clustered into the same cluster. Experimental results show the codebook generated using this method can improve the performance.

7. REFERENCES

[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003, pp. 1470–1477.

[2] M. Marszalek and C. Schmid, "Spatial weighting for bag-of-features," in *CVPR*, 2006, pp. 2118–2125.

[3] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, pp. 2161–2168.

[4] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.

[6] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," in *CVPR*, 2008, pp. 1–8.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–100, 2004.

[8] F. Tang and Y. Gao, "Fast near duplicate detection for personal image collections," in *ACM Multimedia*, 2009, pp. 701–704.

[9] G. Xu and J. Xu, "Efficient approximation algorithms for clustering point-sets," *Computational Geometry: Theory and Application*, vol. 43, no. 1, pp. 59–66, 2010.