Optimizing Visual Vocabularies Using Soft Assignment Entropies

Yubin Kuang, Kalle Åström, Lars Kopp,
Magnus Oskarsson and Martin Byröd

Centre for Mathematical Sciences
Lund University, Sweden

Abstract. The state of the art for large database object retrieval in images is based on quantizing descriptors of interest points into visual words. High similarity between matching image representations (as bags of words) is based upon the assumption that matched points in the two images end up in similar words in hard assignment or in similar representations in soft assignment techniques. In this paper we study how ground truth correspondences can be used to generate better visual vocabularies. Matching of image patches can be done e.g. using deformable models or from estimating 3D geometry. For optimization of the vocabulary, we propose minimizing the entropies of soft assignment of points. We base our clustering on hierarchical k-splits. The results from our entropy based clustering are compared with hierarchical k-means. The vocabularies have been tested on real data with decreased entropy and increased true positive rate, as well as better retrieval performance.

1 Introduction

One of the general problems in computer vision is to automate the recognition process using computer algorithms. For problems such as object recognition and image retrieval from large databases, the state of the art is based on the bags of words (BOW) framework [18, 20, 21, 23]. Firstly a set of interest points are extracted in each of the images using interest point detectors[6, 11, 14, 13] or dense sampling. Then feature descriptors e.g. SIFT or SURF [11, 2, 15] are computed at each interest point. To enable fast matching, feature descriptors are quantized into visual words as a vocabulary, where the descriptors assigned with the same word are regarded as matched. Finally, the co-occurrence of visual words between a query image and those in the database is then used to generate hypotheses of matched images. The matching is often based on the histograms of visual words and the L_1 norm or L_2 norm of differences between two histograms (or the intersection of two histograms) after normalization.

A good vocabulary in the quantization step of the BOW pipeline is crucial for the recognition and retrieval system. Traditional approaches [23, 18, 21, 9] construct vocabulary by clustering descriptor vectors derived from training images in an unsupervised way, i.e. without ground truth information on which correspondence class a specific feature belongs to. These approaches either suffer from

quantization errors or have difficulties in matching wide variety of appearances of objects in images, due to large differences in view points, lighting conditions and background clutter as well as the large intra-class variations of the objects themselves. One way to resolve this is through learning, with the presence of large amount of correspondence ground truth data. While obtaining ground truth data from raw images can be expensive, incorporating such information with proper schemes can enable efficient and accurate recognition performance.

Efforts have been made on learning vocabulary with ground truth information. Winn et al. [26] quantized features with k-means after which the resulting words were merged to obtain intra-class compactness and inter-class discrimination. On the other hand, Moosman et al. used random forests as the quantizer such that at each split an entropy measure based on the class labels is maximized [17]. In [19] Perronnin et al. used class-level labels and proposed to train class-specific vocabularies modeled by GMMs and combine them with a universal vocabulary. The most related work to ours in technical aspects, is the work by Lazebnik et al. in [10], where they simultaneously optimize the quantizer in Euclidean feature space and the posterior class distribution. All these previous works imposed the supervision such that each word in the vocabulary has a discriminative representation of the different object classes. However, they have mainly focused on object categorization and the number of class labels is relatively small (≈ 20) except for the the work in [7] introduced hidden Markov random fields for semantic embedding of local patch features with relatively large number of class labels (≈ 3600). Our approach is designed for image retrieval and uses very large scale ($\approx 80K - 250K$) partially labeled patch correspondences to quantize feature space in a hierarchical manner.

For object recognition, the learned vocabulary has to be more specific regarding matching features. Each word in the vocabulary should contain only small number of features such that each word might encode the appearance variations of a single physical point. In [16], Mikulik et al. start with an unsupervised vocabulary and apply a supervised soft-assignment afterwards, where words are connected based on the statistics of matched feature points from a huge dataset with ground truth correspondences. Another line of work [22, 24], is to incorporate the supervision into the feature metric learning before quantization such that the matched pairs of features have small distances than non-matched pairs in the new mapping. Both methods achieves substantial improvement in the retrieval tasks. Our approach works on the original feature space and encodes the ground truth correspondences in the process of vocabulary generation.

In this paper, we focus on vocabulary for recognition and would like to address systematically the following questions: (*i*) How large should the vocabulary be? In the current literature the sizes range from less than a thousand to millions of words in the vocabulary. This could of course be highly application dependent. (*ii*) How can we evaluate the quality of the vocabulary? (*iii*) What is the optimal division of the feature space into words and How do we avoid splitting matching features into different words? To address the first two questions, we first studied statistically how true positive rates and false positive rates in matching features

of a vocabulary affect the retrieval performance. We then present a framework for supervised vocabulary training using partial or full ground truth information on correspondences. In such a way, we obtain a vocabulary that encodes the intra-class variation of each correspondence class leading to improved retrieval performance.

The rest of the paper is organized as follows. Section 2 contains brief discussion on methods for obtaining the ground truth correspondences. In Section 3 we present the modelling of mAP from vocabulary matching statistics. In Section 4 we describe our optimization method for training the vocabulary using ground truth data. The method is then tested on real image data in Section 5.

2 Ground Truth Correspondence Data

In order to obtain a good visual vocabulary for object recognition in images, we propose to learn the vocabulary using ground truth information on corresponding points. The motivation is that we believe that this strengthens the vocabulary as opposed to just doing unsupervised clustering and we expect the gain to be worthwhile since the the more expensive training with ground truth is a off-line process in the retrieval pipeline.

In order for the learned visual vocabulary to be robust a wide variety of appearances of objects in images, the ground truth datasets should preferably present for the same physical point or same object (*i*) Large intra-class variability of the objects themselves. (*ii*) Large differences in lighting conditions. (*iii*) Large differences in view points. We will discuss in the following some of the methods for obtaining such data sets.

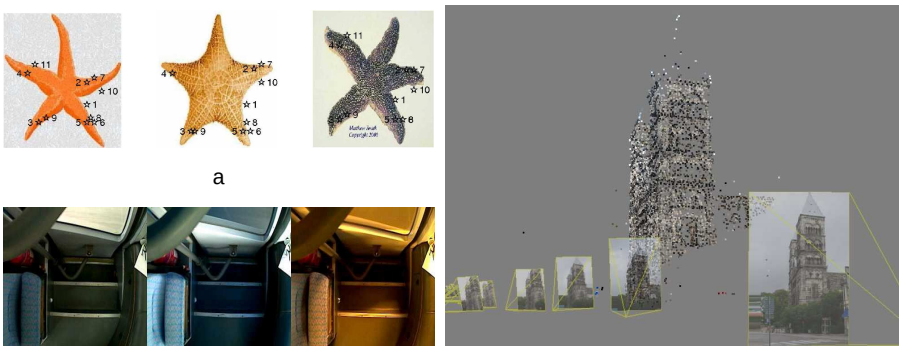

Fig. 1. Three methods of obtaining ground truth correspondences for vocabulary training using (a) deformable shape model (b) static scene with lighting changes and (c) structure from motion algorithms.

.

For object categorization, intra-class variability that is present for most object categories that are interesting for recognition. Deformable models can be used here to generate correspondences. Training these models can be cumbersome, but we believe that this will benefit the training process of the visual vocabulary enabling a very fast but accurate bottom up search process, that is based on learned high level features. In Figure 1(a) a deformable shape model estimated from image data is shown, where the correspondences are based on optimizing the minimum description length according to [8]. The images are from the starfish category of the Caltech-256 object category dataset [4].

The lighting variabilities could be achieved by having images of static scenes taken under substantially changing lighting conditions. Images taken by a camera mounted at the entrance of a moving bus are shown in Figure 1(b). View points variabilities could be obtained by estimating the geometry of objects from images taken at different view points using a RANSAC framework in combination with epipolar geometry estimation such as in e.g. [5, 1]. In Figure 1(c) the typical result from the geometry estimation is shown. From the corresponding points in the images, feature descriptors can then be extracted from the images. In [16], Mikulik et al. present an efficient way of generating large scale ground truth dataset from collections of images by image matching graph. An alternative in [24], Strecha et al. also utilize geo-tags in their 3D-reconstruction pipeline to obtain geometrically consistent patches. In the experimental section of this paper the visual vocabularies are trained on partial ground truth data obtained from the UBC Patch Data [25].

3 Modelling Mean Average Precision from Vocabulary Statistics

One key argument made in this paper is that good retrieval systems, e.g. as measured by mean Average Precision (mAP) can be obtained by studying the properties of the vocabulary on the statistics of descriptor distribution both for random (not necessarily matching) descriptor pairs and for matching descriptor pairs. By matching descriptor pairs we do not mean descriptors that end up in the same word in the vocabulary, but rather descriptors of matching interest regions, i.e. regions which are matching in a ground truth sense.

We can evaluate a vocabulary with two simple characteristics, (i) the false positive rate p_{fp} or FPR, which is the probability that two random descriptors end up in the same word and (ii) the true positive rate p_{tp} or TPR, which is the probability that two matching descriptors end up in the same word.

We argue that the mapping from true positive and false positive rates to mean average precision can be modelled and analyzed. High mean average precision is obtained using vocabularies with low false positive rates and high true positive rates.

The mapping depends on many characteristics of the test, such as the number of features in each image, the number of images in the database, the proportion of positive vs negative answers to a image retrieval query etc. In this model we

have for simplicity assumed that histograms are measured with the normed L_1 distance, but other distance metrics could be used. In fact the modelling could come to good use in determining which metrics to use.

Modelling the L_1-distance Distribution for Two Random Images

Assuming that the distribution of features in different visual words is known, and assuming that features in two random images are independent, it is possible to simulate and model the distribution of L_1 distances. In Figure 2a three such distributions are shown for small, medium and large vocabularies.

For large vocabularies the histograms are sparse. A reasonable approximation here is that the distance is $d = (2n - 2o)/n$, where n is the number of features in the images and o is the number of common features. The number of overlapping features o can be approximated reasonably using binomial distributions using n samples with probability $p = n/N_w$, where N_w is the size of the vocabulary. For increasing vocabulary size this distribution is pushed towards the right end of the spectrum.

Modelling the L_1-distance Distribution for Two Matching Images

For two matching images we assume that there are a number of matching features. For each matching feature pair there is a certain probability p_t that they end up in the same word. For the remaining features we assume that they end up in random words according to the distribution above. The resulting distribution of L_1-distance is similar to that of two random features, but pushed slightly to the left. In Figure 2a three such distributions are shown, again for small, medium and large vocabularies.

Modelling Precision, Recall and Mean Average Precision

For each vocabulary as characterized by its true and false positive rates (p_{tp}, p_{fp}), we can estimate the probability distribution of matched image L_1-distance, p_m, and the probability distribution of two random image L_1-distance, p_r.

Assuming that in a random query there are N_{inlier} matching images and $N_{outlier}$ non-matching images. For each threshold D of L_1-distances we obtain a query result with recall

$$R = \frac{N_{inlier} \int_0^D p_m(x)dx}{N_{inlier} \int_0^2 p_m(x)dx} = \int_0^D p_m(x)dx$$

and precision

$$P = \frac{N_{inlier} \int_0^D p_m(x)dx}{N_{inlier} \int_0^D p_m(x)dx + N_{outlier} \int_0^D p_r(x)dx} = \frac{\int_0^D p_m(x)dx}{\int_0^D p_m(x)dx + K \int_0^D p_r(x)dx}$$

, where $K = \frac{N_{outlier}}{N_{inlier}}$ is the ratio of outliers to inliers in a typical query.

Note that the domain of the normalized L_1 distance is between [0,2]. Therefore, in the equation for recall, we have used 2 as the integration limit in denominator. It follows that the integral in the denominator is 1. From these two curves it is straightforward to estimate the mean average precision.

Figure 2b shows how the mean average precision depends on the 10-log of the true and false positive rates (p_{tp}, p_{fp}). Notice that this confirms the theory that quite large vocabularies are needed for good performance.

A key argument made here is that e.g. for hierarchical vocabulary building, increased levels of splitting of the vocabulary gives lower true and false positive rates. But already for small vocabularies, by demonstrating that one obtains higher true positive rates, while retaining a low false positive rate will be beneficial for the end performance as measured by the mean average precision.

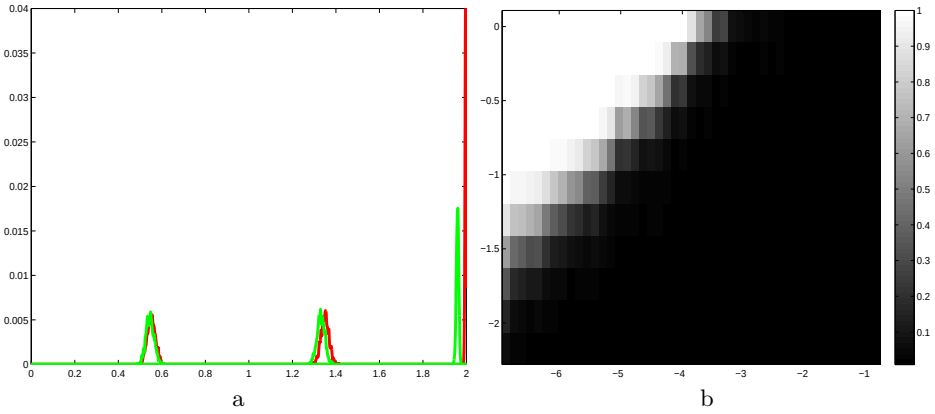

Fig. 2. (a) L_1-distance distributions for random image pairs (red) and matching image pairs (green) for three vocabularies of different size. (b) Mean Average Precision as a function of 10-log of false positive rate (x-axis) and 10-log of true positive rate (y-axis).

4 Optimizing the Vocabulary with respect to Entropy

We will concentrate on hierarchical divisions of the descriptor space. The resulting vocabularies have the advantage that visual word generation is extremely efficient. Another advantage during training is that the learning and corresponding optimizations only have to be done at each hierarchical split in the tree.

We assume that a number of descriptors are given, $x_i \in \mathbb{R}^d, i = 1 \ldots N$, and that correspondences among such descriptors are known. Here we have represented such correspondences as the correspondence class C_i for each point i. The number of correspondence classes is denoted by N_c. In the typical datasets that we have worked on, the numbers of descriptors are in the order of 500K and the numbers of correspondence classes are in the order of 150K. The correspondences have been generated by sampling from 3D reconstructions of scenes. See Section 5.1 for details. Other ways of generating correspondences could be through geometric matching in image pairs, tracking of points in image sequences or by hand annotated data.

We will concentrate on the problem of recognizing specific scenes and the data that we have used is chosen so that points are in correspondence if they

denote the same physical point in the scene. That descriptors are in different correspondence classes does not necessarily mean that they are *not* in correspondence. On the contrary, we expect there to be many descriptors in different correspondence classes that actually correspond quite well. However for points that are in correspondence we would like the corresponding descriptors to end up in the same word in the final vocabulary.

Our hierarchical division will be based on k splits in the descriptor space D at each step. Each such split is represented by k center points $c_1, \ldots, c_k$ and a scalar m that can be interpreted as a margin. Low values of m represent sharper cuts and high values represent softer classification.

We study both hard assignment and soft assignment in the following sense. For hard assignment a descriptor is put in the bin i corresponding to the closest center c_i. For soft assignment we put each point x in the k bins in proportion to the weight w_i according to

$$w_i = \frac{exp(\frac{||x-c_i||_2}{m})}{\sum_{j=1}^{k} exp(\frac{||x-c_j||_2}{m})}. \quad (1)$$

where $||.||_2$ is the L_2 norm. Contrary to [10] we use exponential distributions, which give smoothing that only depends on the differance of distances to the cluster center. Descriptors for which the distance to the closest centers is similar fall into several bins to a fair degree, whereas descriptors for distance difference between the two closest centers is much larger than m fall essentially only into one part of the tree.

4.1 Entropy Model

To optimize the division parameters $z = (c_1, \ldots, c_k)$ for hard assignment and $z = (c_1, \ldots, c_k, m)$ for soft assignment, we use entropy as a criterion. Entropy takes into account both that the split is balanced, i.e. that approximately equal number of descriptors fall into each bin, and that the correspondence classes are split as cleanly as possible. The entropy for a random variable X with N possible states is defined as $E = -\sum_{i=1}^{N} p(i) \log_2(p(i))$, where p is the probability density function of X. Here we use the 2-log as it is more intuitive and easier to interpret.

Entropy is fairly easy to use in the sense that it is straightforward to define for both hard and soft assignment. The probability density function is calculated in the following manner. In each split we calculate the (weighted) histogram of descriptors in each correspondence class $h_{tot} = (h(1), \ldots, h(N_c))$ before the split. Each descriptor falls partly in the k different parts of the tree, thus contributing in part to both the k-weighted histogram $h_1, \ldots h_k$.

By normalizing the histograms with the sum, we obtain correspondence class probabilities, i.e. $p_{tot}(i) = \frac{h_{tot}(i)}{\sum_{i=1}^{N_c} h_{tot}(i)}$, for the distribution of descriptors among the correspondence classes before the split and similarly for $p_1, \ldots, p_k$. The entropy before the split is defined as $E_{tot} = \sum_{i=1}^{N_c} -p_{tot}(i) \log_2(p_{tot}(i))$, and similarly for the k branches, $E_j = \sum_{i=1}^{N_c} -p_j(i) \log_2(p_j(i))$. For the split as a whole

we define the entropy as $E_{split} = \sum_{j=1}^{k} \frac{n_j}{n_{tot}} E_j$. Here $n_j = \sum_{i=1}^{N_c} h_j(i)$. Ideally each split, which uses $log_2(k)$ extra bits of information, should lower the entropy with $log_2(k)$ bits, i.e. we expect E_{split} to be approximately $log_2(k)$ less than E_{tot}. In practice it is difficult to split all examples in the descriptor space as cleanly as this.

4.2 Optimizing Entropy

For training data $(x_1, \ldots, x_N), (C_1, \ldots, C_N)$ with possible weights $(y_1, \ldots, y_N)$, it is thus possible to define the split entropy E_{split} as a function of the division parameters z. For hard assignment, using $z = (c_1, \ldots, c_k)$, this function is not smooth. The entropy is typically constant as the decision boundaries are perturbed as long as they do not pass through any of the points x_i. For soft assignment, however, entropy is a smooth function of the division parameters $z = (c_1, \ldots, c_k, m)$.

In our experiments we have tried a few different approaches for optimizing E with respect to z. We did not optimize E with respect to the margin m in this paper.

In the main approach we initialize using k-means iterations with a couple of different starting points. The best initial estimate is then used as an initial estimate to a non-linear optimization of z. Here we have calculated the analytical derivatives $\frac{dE}{dz}$, which are then used in a non linear optimization.

The entropy for the split can be written as $E_{split} = \sum_{j=1}^{k} \frac{n_j}{n_{tot}} E_j$. which since $n_j p_j(i) = h_j(i)$ gives $E_{split} = \sum_{j=1}^{k} \frac{1}{n_{tot}} \sum_{i=1}^{N_c} (-h_j(i) \log_2(p_j(i)))$. The derivative of E_{split} is thus

$$\frac{dE_{split}}{dz} = \frac{-1}{n_{tot}} \sum_{j=1}^{k} \sum_{i=1}^{N_c} \left(\frac{dh_j(i)}{dz} \log_2(p_j(i)) + \frac{n_j}{\ln(2)} \frac{dp_j(i)}{dz}\right) \tag{2}$$

Here the sum of the second term over all i is zero, since the sum of the probabilities is constant. Thus

$$\frac{dE_{split}}{dz} = \frac{1}{n_{tot}} \sum_{j=1}^{k} \sum_{i=1}^{N_c} \left(-\frac{dh_j(i)}{dz} \log_2(p_j(i))\right). \tag{3}$$

The derivatives of the histogram bins are $\frac{dh_j(i)}{dz} = \sum_{l, C_l = i} \frac{d\omega_j(l)}{dz}$. Finally the derivatives of the weights are

$$\frac{d\omega_j(l)}{dz} = \frac{\frac{de_j(l)}{dz}}{\sum_{j=1}^{k} e_j(l)} - e_j(l) \frac{\sum_{j=1}^{k} \frac{de_j(l)}{dz}}{(\sum_{j=1}^{k} e_j(l))^2}, \tag{4}$$

where $e_j(l) = \frac{exp(||x_l - c_j||_2)}{m}$ and

$$\frac{de_j(l)}{dz} = e_j(l) \left(\frac{(x_l - c_j)}{m||x_l - c_j||_2} \frac{dc_j}{dz} - \frac{||x_l - c_j||_2}{m^2} \frac{dm}{dz}\right). \tag{5}$$

The value E and the gradient $\frac{dE}{dz}$ are utilized in a non-linear optimization update with the limited-memory Broyden-Fletcher-Goldfarb-Shanno method, [3, 12]. In the implementation we have limited the maximum number of iterations of the optimization to 20 iterations for the first levels, but increased to 30 iterations for the subsequent levels to avoid over-fitting. This scheme is general for different values of k.

5 Experimental Validation

We have tested our method on vocabulary construction with real image data. The dataset is described in details in Section 5.1. The resulting vocabularies are evaluated in Section 5.2.

5.1 Dataset and Evaluation

We use three sets of data with partial ground truth on correspondences, from the UBC Patch Data [25]. These datasets contain scale and orientation normalized patches (from either difference of Gaussians (DOG) or Harris corners detectors) sampled from 3D reconstructions of three landmarks (Statue of Liberty, Notre Dame and Yosemite). In Figure 3 we show two sets of patches in the same correspondence class from the Statue of Liberty and Notredame dataset respectively. Each dataset (Notre Dame, Liberty and Yosemite) contain approximately 500K descriptors in 150K correspondence classes.

For our experiments, we extracted SIFT descriptors on DOG patches. To provide correspondence ground truth for training and evaluation, we generated the whole set of matched pairs for each correspondence class, and a random non-matched for each patch to form non-matched pairs (with the same reference image as suggested by [25]).

We then used the methods in Section 4 to construct vocabularies based on the SIFT descriptors and partial ground truth for these datasets. We have here used a subset of the data for the training and another non-overlapping subset for the testing.

5.2 Vocabularies with Hard Assignment

In the first experiment we trained vocabularies with hierarchical $k = 3$ splits with 9 levels by optimizing entropy based on soft assignment. When testing, we used hard assignment with respect to the optimized k cluster centers. We compare the results with those of hierarchical k-means with 3 splits in each node. The vocabularies are trained both for hierarchical k-means and for entropy optimization on a subset (50 percent) of the Statue of Liberty dataset.

The resulting vocabularies were then tested on a subset of the Statue of Liberty dataset (20 percent) which does not contain the same correspondence classes as were used in the training. We measured how the entropy decreases with increasing vocabulary size. Also a subset of matching points were used to test

how often two matching points end up in the same word (True Positive Rate, TPR) . Finally a subset of pair of random unmatched points in the dataset were used to see how often two unmatched points end up in the same word (False Positive Rate, FPR). This result is shown in figure 4. Notice also that the probability of two matching features ending up in the same word is higher for the entropy minimized vocabulary for unseen data points, which suggests the generality of the learned vocabulary. Moreover, we obtained slightly lower FPR across different levels of the tree. We also observed that the entropy is lowered by approximately 1.5 bits ($\log_2(3) \approx 1.585$) with each level in the hierarchical split, but slightly more so when using an entropy minimized vocabulary, suggesting that entropy is a fair measure on the quality of the resulting clusters. In this experiment we used a fixed setting for the margin $m = 1$.

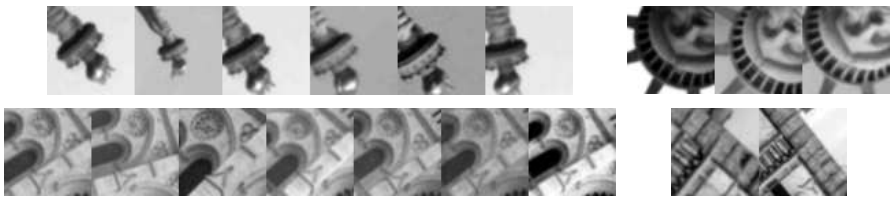

Fig. 3. Correspondence patches from the Statue of Liberty (Top) and Notredame (Bottom) dataset..

To further investigate the generality of the method, we have trained vocabularies on 50% of the features from the Statue of Liberty, the Notre Dame and the Yosemite datasets and tested it on the remaining 50% features. The optimized vocabulary compared to hierarchical k-means results in lower entropy and higher TPR. The resulting plot is very similar to Figure 4 suggesting the optimized vocabulary generalized well to new data.

5.3 Vocabularies with Soft Assignment

In the next experiment, we used the same vocabulary as in Section 5.2, but switched to soft assignment when passing unseen feature points down the hierarchical tree. Features can then fall in several children nodes where their weights to the corresponding centers are larger than a preset threshold $\epsilon = 10^{-6}$. This results in multiple word ID's for a single feature. If we regard two features as matched if they share the same words as before, we will expect higher TPR as matched features will have greater possibility of overlapping. On the other hand, two random non-matched features will also tend to have one of the word ID's in common. Consequently, the FPR will also increase. Here, we also fixed the margin to $m = 1$ during training.

We expect our optimization framework to improve the TPR while controlling the FPR by training on ground truth data. In Figure 4, we can see that, the

proposed method is marginally better than the hierarchical k-means with respect to the TPR and FPR curve. Only achieving marginal optimality might be due to the fact that we have not used enough data for training. On the other hand, we noted that both soft assignment vocabularies have better matching property than hard assignment vocabularies. For instance if we aim for 5% false positive rate, soft assignment achieves approximately 60% true positive rate while hard assignment obtains only 45%.

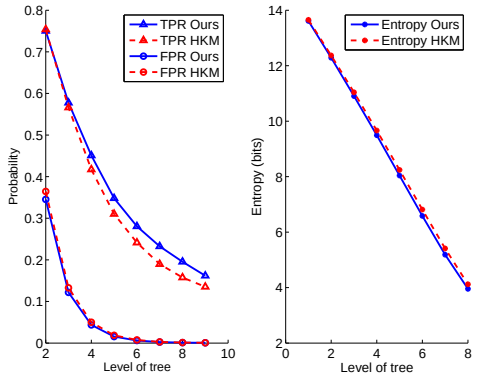

Fig. 4. Evaluation on Liberty data. (50% for training and 20% for testing with k = 3. Left: Estimated probability of two corresponding (TPR) and two random descriptors (FPR) ending up in the same word as a function of tree depth. Middle : Entropy as a function of tree depth. Notice that with each depth entropy is lowered close to 1.5 bit.

5.4 Effects of Margin

In this section, we studied the effect of different margins on soft assignment tests. Here we fixed the value of m during the training stage and evaluate how margins affect the match performance for test data. Note that as m becomes smaller, the soft assignment behaves in a similar way as hard assignment. On the other hand, larger m implies more ambiguities for each features ending up in different words; therefore, possibly higher false positive rate for matching.

We have experimented with $m = 0.25, 0.5, 1$ (Figure 5). As expected, when increasing the margin we can achieve better TPR with the trade-off of worse FPR at the same level of the tree. The optimized vocabularies are better than hierarchical k-means across different margins indicating the usefulness of utilizing ground truth. More importantly, the overall statistics shed lights on how we should choose the size of the vocabularies (level of hierarchical trees). The converging trend of all curves with different m's suggests that at certain number of words, we can always obtain better TPR with soft assignment but approximately the same FPR. However, such better performance comes at the price

of heavier computation when assigning features to multiple leaf nodes. If m is too large, features will end up in many words at the leaf node. Therefore, the efficiency of vocabulary representation of features is overwhelmed by computing the intersections in the space of word ID's.

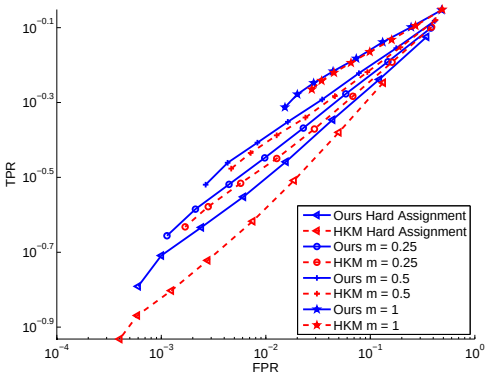

Fig. 5. The effects of different margins on soft assignment with respect to TPR and FPR. $m = 0.25, 0.5, 1$ and hard assignment, where k = 3

5.5 Image Retrieval

In this section, we verify the usefulness of optimized vocabulary in the recognition pipeline on the Oxford 5K dataset [20, 21]. The task is to retrieve similar images to the 55 query images (5 for each of the 11 landmarks in Oxford) in the dataset of 5062 images. The performance is then evaluate with mean Average Precision (mAP) score. Higher mAP indicates that the underlying system on average retrieves the similar corresponding images at the top of the ranked list.

We follow the BOW baseline system, and use a hierarchical k-means vocabulary and our optimized vocabulary respectively for vocabulary training. We trained the vocabulary with 50% of a mixture of Liberty, Notredame and Yosemite patch data which contains approximately 800K features and 250K correspondence classes in total. After that, we use hard assignment to quantize the SIFT features from the Oxford 5K images. We observe that our optimized vocabulary is always superior to the unsupervised hierarchical k-means by capturing the local characteristics of the feature space. When increasing the number of levels to 11 we can see that the performance drops both for hierarchical k-means and our method. This can be an indication that the vocabulary is over-trained on the patch data. Note that these results are not directly comparable with [20] in which vocabularies are trained on features in the images where the actual retrieval is performed.

Level	HIK k = 3	Our Method k = 3
9	0.1744	0.1955
10	0.1849	0.1979
11	0.1805	0.1837

Table 1. mAPs with different levels of hierarchical k-means and our method with k = 3 on the Oxford 5K dataset

6 Conclusions

In this paper, we have developed a general method for optimizing hierarchical visual vocabularies using correspondence ground truth between features. The ground truth prior knowledge on the feature space is utilized to refine the local structures of the trained vocabulary such that matched features will tend to fall in the same word. We propose the use of a soft margin hierarchical k-splits tree where the optimization of the tree is based on minimizing an entropy criterion defined on ground truth data. Unlike the traditional clustering methods such as hierarchical k-means, optimization with respect to entropy enables the cluster centers to adjust locally to capture the implicit connections between features. We demonstrate the method on real dataset with promising results. Compared to the unsupervised hierarchical k-means with hard assignment, the optimized vocabulary obtained higher true positive rate and lower false positive rates. We also show that soft assignment boosts the overall performance regarding matching features.

We have in this paper focused on the optimization aspects of vocabulary training using existing ground truth data. Due to the high dimensionality of the parameter space, the learning requires huge amounts of data in order to avoid over-fitting. Therefore, as future work, we aim to generate and utilize large scale ground truth data to facilitate robust training with geometry or deformable models. We need also to cope with the inherent quantization errors introduced by hierarchical quantization. We would like to investigate how the soft-assignment process might mitigate the such quantization errors. To enable large scale training, we are also pursuing efficient optimization techniques for our approach.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: Proc. 12th Int. Conf. on Computer Vision, Kyoto, Japan. (2009)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: 9th European Conference on Computer Vision, Graz Austria (2006)
3. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms. Journal of the Institute of Mathematics and its Applications **6** (1970) 76–90
4. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)

5. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004) Second Edition.
6. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of the 4th Alvey Vision Conference. (1988) 147–151
7. Ji, R., Yao, H., Sun, X., Zhong, B., Gao, W.: Towards semantic embedding in visual vocabulary. In: Proc. Conf. Computer Vision and Pattern Recognition, San Francisco, California, USA. (2010)
8. Karlsson, J., Åström, K.: MDL patch correspondences on unlabeled data. In: Proc. International Conference on Pattern Recognition, Tampa, USA. (2008)
9. Lamrous, S., Taileb, M.: Divisive hierarchical k-means. CIMCA-IAWTIC'06,IEEE Computer Society (2006)
10. Lazebnik, S., Raginsky, M.: Supervised learning of quantizer codebooks by information loss minimization. IEEE Trans. Pattern Analysis and Machine Intelligence **31** (2009) 1294 – 1309
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. Journal of Computer Vision **60** (2004) 91–110
12. Luenberger, D.G.: Linear and Nonlinear Programming. Addison-Wesley (1984)
13. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vision Computing **22** (2004) 761–767
14. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: European Conference on Computer Vision, Springer (2002) 128–142 Copenhagen.
15. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: Proc. Conf. Computer Vision and Pattern Recognition. (2003)
16. Mikulik, A., Perdoch, M., Chum, O., Matas, J.: Learning a fine vocabulary. In: Proc. 11th European Conf. on Computer Vision, Crete, Greece. (2010)
17. Moosmann, F., Triggs, B., Jurie, F.: Randomized clustering forests for building fast and discriminative visual vocabularies. In: Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, Canada. (2006)
18. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: IEEE Conference on Computer Vision and Pattern Recognition. (2006) 2161–2168
19. Perronnin, F.: Universal and adapted vocabularies for generic visual categorization. IEEE Trans. Pattern Analysis and Machine Intelligence **30** (2008) 1243–1256
20. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007)
21. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2008)
22. Philbin, J., Isard, M., Sivic, J., Zisserman, A.: Descriptor learning for efficient retrieval. In: Proc. 11th European Conf. on Computer Vision, Crete, Greece. (2010)
23. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proceedings of the International Conference on Computer Vision. (2003)
24. Strecha, C., Bronstein, A., Bronstein, M., Fua, P.: LDAhash: Improved matching with smaller descriptors. Technical report, CVlab, EPFL Switzerland,Tel-Aviv University and Israel Institute of Technology, Israel (2010)
25. Winder, S., Hua, G., Brown, M.: Picking the best daisy. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR09), Miami (2009)

26. Winn, J., Criminisi, T., Minka, T.: Object categorization by learned visual dictionary. In: Proc. 10th Int. Conf. on Computer Vision, Bejing, China. (2005)