Leveraging Structure from Motion to Learn Discriminative Codebooks for Scalable Landmark Classification

Alessandro Bergamo
Dartmouth College
Hanover, NH, USA
aleb@cs.dartmouth.edu

Sudipta N. Sinha
Microsoft Research
Redmond, WA, USA
sudipsin@microsoft.com

Lorenzo Torresani
Dartmouth College
Hanover, NH, USA
lorenzo@cs.dartmouth.edu

Abstract

In this paper we propose a new technique for learning a discriminative codebook for local feature descriptors, specifically designed for scalable landmark classification. The key contribution lies in exploiting the knowledge of correspondences within sets of feature descriptors during codebook learning. Feature correspondences are obtained using structure from motion (SfM) computation on Internet photo collections which serve as the training data. Our codebook is defined by a random forest that is trained to map corresponding feature descriptors into identical codes. Unlike prior forest-based codebook learning methods, we utilize fine-grained descriptor labels and address the challenge of training a forest with an extremely large number of labels. Our codebook is used with various existing feature encoding schemes and also a variant we propose for importance-weighted aggregation of local features. We evaluate our approach on a public dataset of 25 landmarks and our new dataset of 620 landmarks (614K images). Our approach significantly outperforms the state of the art in landmark classification. Furthermore, our method is memory efficient and scalable.

1. Introduction

Given a pre-defined set of locations and landmarks, we address the problem of recognizing the landmark (or location) from a single image. We are specifically interested in designing an approach that can scale to classify a very large number of different landmarks and that can perform recognition efficiently. With billions of tagged Internet photo publicly available on the Internet, addressing this problem at a large scale is of fundamental importance, as evidenced by the increasing interest in this topic by the computer vision community [12, 17, 18].

The main challenge in landmark recognition arises from the great diversity of camera viewpoints as well as variations in scale and appearance due to illumination changes from time-of-day, weather, seasons etc. Although the intrinsic scene appearance does not change a lot, the visibility of scenes covering a wide area can change dramatically with viewpoints, producing a diverse set of images (see the *Piazza Navona* reconstruction in Figure 1). In addition, scene appearance at locations such as *Times Square* change with time, making them very challenging to recognize.

Broadly speaking, most existing methods for landmark classification build upon scalable image retrieval techniques [28, 22, 13, 3, 26, 12]. Although this enables fast querying against a database of many millions of images, constant access to this huge database is necessary with consequent high storage and memory costs. For high precision, an expensive re-ranking step is often used for geometric post-verification of feature matches over image pairs. Recently proposed alternatives include direct image matching to sparse SfM point clouds [27, 17, 18]. Despite bringing some compression, these methods still have high storage and computational requirements.

Image categorization methods, on the other hand, use labeled data to train classifiers that are typically efficient to evaluate at query time and that have much lower feature storage costs [6, 24, 25]. However, most of these techniques have traditionally focused on categorization tasks involving a small number of classes [2]. Large-scale and fine-grained classification has recently gained interest amongst researchers, however, with the exception of [16, 8], we are not aware of much work focusing on landmark or location classification within this framework.

In this paper, we pose landmark recognition as a categorization task, by treating each landmark as a class. We focus on discovering a compact yet exhaustive set of discriminative features and using them within a standard classification pipeline. Within this setting, our objective is to learn a compact, discriminative codebook from training data and use it with existing feature encoding schemes and some variants that we propose. Building upon recent advances in viewpoint invariant feature matching and structure from

Figure 1: [TOP]: Examples of SfM reconstructions from LANDMARK-620 our Internet photo collection dataset (the largest component is shown and the number of cameras and 3D points are listed). [BOTTOM]: Six out of approx. 8 million tracks *i.e.* sets of matching keypoints recovered from running SfM on LANDMARK-620 (patches are visualized at their original scales).

motion [29], we construct a database of millions of feature descriptors with automatically-established scene correspondences, i.e., we know which descriptors (from different images) correspond to the same 3D scene point, with geometric consistency guaranteed.

Figure 1 shows a few such reconstructions and examples of discovered feature correspondences which we refer to as tracks. We propose to exploit such feature correspondences in the training data to learn a compact codebook that maps matching descriptors into the same code. Large codebooks learned in an unsupervised setting can also be discriminative [22, 26], but they are only suitable for large-scale image retrieval [22, 26, 20]. In classification tasks, compact codebooks provide computational efficiency, as larger codebooks typically increase the dimensionality of feature vectors used by the classifiers. Recently, feature correspondences were exploited to learn spatial co-occurrence statistics [34, 36]. However, these methods tackle the retrieval problem, and cannot be directly applied to large-scale multiclass classification, which is the problem we address.

Our codebook is expressed by a random forest that we learn in a supervised fashion, using the training descriptors and their fine-grained *track* labels. We use the codebook as a tree quantizer for bag of words histogram encoding and also with other advanced schemes [13]. The supervised training of the codebook using track labels reduces quantization error on these repeatable and discriminative features. Unlike prior discriminative methods to codebook learning [21, 15] which use descriptors labeled by *landmark categories*, our descriptors have fine-grained labels. This is beneficial as it forces the codebook learning method to predict the same label only for descriptors that truly correspond to the same physical scene point. However, this also implies that we must train a codebook with 3–4 orders of magnitude more labels ($\sim 10^5$). To address this issue, we propose to learn the codebook using a random forest [5] which is efficient to train and it enables fast quantization. In order to help dealing with the large number of class labels we exploit two ideas: (1) we adopt the splitting rule used in *random projection trees* [7] originally proposed for unsupervised tasks, and (2) we use an improved information gain estimator proposed in [23] to select the best split.

We evaluate our approach on a public landmark dataset from [11], and our own dataset of 620 landmarks (614K images). Our proposed technique significantly outperforms [16], which is the approach closest to our work. It is also worth noting that just using SfM to remove outliers from a collection of more than 2 million web images, significantly boosted the accuracy of the baseline [16].

2. Related work

Most existing location classification methods are based on an *instance recognition* framework [3], that builds upon earlier work in efficient and scalable image retrieval techniques [28, 22, 26]. However, recently Li *et al*. [16] treated landmarks as visual categories and proposed a generic image classification pipeline [6, 24] based on Kmeans codebooks and bag of words encoding used with multi-class SVMs. While their goal is similar to our objective in this paper, they did not exploit 3D scene structure in their work.

Meanwhile, better SfM algorithms have led to direct 2d-3d matching approaches [17, 18, 27], where known 3D scene structure is utilized to create a structured database [18] – these methods can in addition compute a precise camera pose for the query image. SfM techniques have also been used indirectly by Mikulik *et al*. [20], who exploit feature correspondences to reduce quantization errors in nearest neighbor quantizer codebooks. An improved similarity measures to compare bag of words histograms is proposed for image retrieval with very large codebooks in [20].

Better aggregation techniques such as VLAD [13] was shown to improve upon bag of words methods for retrieval as well as classification tasks [2]. Other aggregation strategies such as learning feature weights have also been proposed to handle background clutter [1, 14]

Although unsupervised codebooks continue to be popular, discriminative codebooks are known to be superior [32] as they aim to encode the differences between categories [24]. Our work is related to supervised learning of nearest neighbor quantizers [15] that optimize an objective based on minimizing information loss, and to tree-based quantizers such as Extremely Randomized Clustering (ERC) forests proposed by Moosmann *et al*. [21, 21]. However, these discriminative codebooks were learnt using image category labels in a setting where the number of categories *i.e*. the number of labels were quite small. Our codebooks are based on random forests [21] also, but are trained on many thousands of fine-grained descriptor labels. Like in [21], each tree in our forest serves as an independent randomized codebook, assigning each descriptor efficiently to one codevector per tree. This differs from soft quantization approaches [26, 10], where a descriptor is assigned multiple codevectors using a single codebook. Multi-tree quantizers have been used for retrieval previously but were trained in an unsupervised setting [33].

Recently Doersch *et al*. [8] proposed a method to discover discriminative mid-level patches from urban imagery, but rely on discovering them from random patches sampled from the training data. In contrast, our method exploits robust matching in image collections and exploits SfM where after the first phase of pairwise image matching is performed, pairwise feature correspondences are linked over multiple images to produce long tracks *i.e*. larger sets of matched descriptors. The main focus of our paper however, is not on discovering the discriminative features but rather on learning a compact codebook from them in a way that retains their discriminative power.

3. Technical Approach

In this section, we give an overview of our method which is related to image categorization [6, 24], and is similar to the approach proposed in [16]. For training, we are given labeled images where each label represents a landmark.

We first perform SfM computation [29] independently for each landmark to obtain sparse reconstructions of 3D points $\{\mathcal{R}^k\}_{k=1}^{L}$ for the L landmarks[1]. Each 3D point in $\mathcal{R}^k$ is associated with a set of 2D keypoints extracted from different input images. We refer to the set of feature descriptors corresponding to these keypoints as a *track*. Thus, the i-th 3D point in $\mathcal{R}^k$ gives us a set of matching descriptors, $\mathcal{T}^{(i,k)} = \{\mathbf{x}_j^{(i,k)}\}_{j=1}^{m^{(i,k)}}$ where $\mathbf{x}_j^{(i,k)} \in \mathbb{R}^d$ and these $m^{(i,k)}$ descriptors share the same label (i,k). Considering all the tracks together, we define a training set of labeled descriptors $\mathcal{D}^{\mathcal{T}} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{n}$, where $y_j \in \{1, ..., N\}$ is the track label of the j-th descriptor $\mathbf{x}_j$ and N is the total number of unique tracks in all the reconstructions considered together. A codebook is then learnt from $\mathcal{D}^{\mathcal{T}}$ and subsequently used to encode the training images for learning a linear SVM. In Section 3.1, we describe our method to learn a discriminative random forest codebook that exploits track labels. In Section 3.2, we discuss how this codebook is used with popular encoding schemes and our proposed variants.

To classify an image, we extract local keypoints and compute feature descriptors which are then aggregated into a high-dimensional global image descriptor $\mathbf{f}$ by means of the learned codebook. The linear SVM is then used to predict the class label *i.e*. the *landmark-id*.

Track Selection. In practice, for hundreds of landmarks, the total number of 3D points in $\{\mathcal{R}^k\}_{k=1}^{L}$ can easily exceed several millions. Hence, in practice we decimate $\mathcal{D}^{\mathcal{T}}$, and learn our vocabulary on a subset $\mathcal{D}^s$. We prefer selecting longer tracks, since these features tend to be more repeatable and often more visually salient within the scene[2]. However, short tracks can contain discriminative visual features too, especially in images with rare viewpoints. To deal with potentially uneven viewpoint distributions common with landmark images, we adopt a randomized strategy to sample tracks from $\mathcal{D}^{\mathcal{T}}$. We randomly select a user specified number of tracks η per landmark, where the selection probability is proportional to the track length, which must be greater than a threshold τ; we set τ=10.

[1] if multiple connected components exist, we consider all of them

[2] the ratio test heuristic [19] in SfM [29] prunes ambiguous feature matches and retains distinctive ones.

3.1. Track-based Random Forest Codebook

Random forests have been used for a wide variety of tasks in the past, such as classification, regression, density estimation etc. [5]. Given the training set $\mathcal{D}^s$ with track labels, our approach to learn a random forest codebook from tracks (denoted as RFT) is now described. Unlike nearest neighbor quantizer codebooks, which are composed of a discrete set of codevectors $\in \mathbb{R}^d$, the RFT codebook contains one or more binary trees each of which represents a partition of feature space based on recursive binary splits. Each internal tree node has a binary split function denoted $h(\pi, \mathbf{x})$, where π represents a d-dimensional hyperplane and $h(\pi, \mathbf{x})$ tests which side of it, $\mathbf{x}$ lies on. The leaf nodes correspond to the actual codes in the RFT codebook.

We train each tree using bagging on a subset of $\mathcal{D}^s$, built by random sampling $\kappa\%$ of the data without replacement. Each decision tree is learnt independently in greedy fashion. The hyperplane parameters π at each tree node j are obtained by solving for the optimal $\pi_j^* = \arg\max_{\pi \in \mathbf{\Pi}} I_j$, where $\mathbf{\Pi}$ is a finite set of hyperplane hypotheses (splits) that are tested and I_j is the information gain defined in terms of the distribution of labels: $I_j = H(\mathcal{S}_j) - \sum_{i \in \{L,R\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i)$. Here, $\mathcal{S}_j$ denotes the set of training points associated with node j, and $\mathcal{S}_j^R$ and $\mathcal{S}_j^L$ denote the binary partition of $\mathcal{S}_j$ induced by a particular hyperplane at node j. $H(\mathcal{S})$ denotes an empirical entropy estimate given a certain distribution of labels in $\mathcal{S}$.

The tree count T and maximum tree height h are parameters that determine the codebook size ($T2^{h-1}$). For a target size, the optimal (h, T) pair is found using grid search. We now present two crucial aspects for training random forests with a very large set of labels – (1) generating the split hypotheses $\mathbf{\Pi}$ and (2) choice of $H(\mathcal{S})$.

Spatial Partitioning with Random Projection trees [7]: Decision trees are often trained with binary split functions that correspond to Axis-Aligned hyperplanes – a single coordinate of $\mathbf{x}$ is compared to a threshold. Although extremely efficient, axis aligned RFT codebooks require larger tree height and tree count to match the accuracy possible with fewer trees, when general hyperplanes or nonlinear split functions are used [5]. Larger tree count or tree height both increase codebook size which is undesirable.

Dasgupta *et al*. [7] proposed *Random Projection* (RP) trees, a data structure for spatial partitioning of high dimensional data that adapts to the *intrinsic dimensions* of the data without explicit learning, and applied it successfully to several unsupervised tasks on unlabeled data. RP trees are built using a special rule for recursive binary splits on the data. First, a random direction v in $\mathbb{R}^d$ is selected. Then, the median is computed for the set of points projected along v *i.e.* the set $\{z^\top v : z \in \mathbb{R}^d\}$. Adding a small perturbation to the median based on the data distribution [7], gives the offset that partitions the points about equally using a hyperplane orthogonal to v. To train our random forest, we use this algorithm to generate the set $\mathbf{\Pi}$, using ρ random projections and medians.

Improved Information Gain estimator: The Information Gain I_j is typically used with the standard entropy estimator, $H_S(\mathcal{S}) = -\sum_{c=1}^{|\mathcal{C}|} p(c) \log p(c)$ where $p(c)$ is the empirical probability distribution given the set of labels $\mathcal{C}$. As already noted in [23], H_S is a biased estimator, particularly when the number of classes is very large. In a such situation, the training of a Random Forest does not learn accurate models for the training data. A new unbiased estimator H_G was proposed in [23] that has been shown to produce better results in cases involving many labels (see [23] for the details). Note that in our scenario the number of tracks is also very large. Therefore we used the improved unbiased estimator H_G for all our experiments.

3.2. Feature encoding

Given a codebook that partitions feature space into regions called *visual words*, a local descriptor can be quantized into words and a histogram of all word occurrences in the image serves as its global descriptor. This is the bag of words histogram *BoW* encoding [6, 28, 22]. Codebooks such as *K-means* use nearest neighbor search to assign a code to a descriptor. Random forest codebooks (*ERC*, *RFT*) on the other hand, use tree traversal which takes time logarithmic in the number of codes. These codebooks assign multiple codes to a descriptor, one per tree, but then computes histograms similar to *BoW* methods.

An alternative encoding called *VLAD* [13] used with *K-means*, encodes the difference between the K centroids and the descriptors assigned to them. The assignment step is identical to *BoW*, but now, an average difference vector from each centroid to all its associated descriptors is computed. These K vectors are concatenated to obtain the global descriptor. An extra step is required during training, to use *RFT* codebooks with *VLAD* encoding. A centroid is estimated as the mean of all the examples assigned to the leaf. *RFT* codebooks with *VLAD* encoding allow fast assignment of descriptors to leaves via tree traversal and provides the benefit of multiple codes (one per tree).

We also tried a variant of *VLAD* using soft-assignment weights, which can be viewed as a combination of *VLAD* with *kernel codebook encoding* [10]. Given a *RFT* codebook forest with K leaves, for each leaf, we compute the mean difference vector between its centroid and all the descriptors, weighting each difference vector using a Gaussian kernel. More formally, we define K vectors $\{\mathbf{v}_k = \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{c}_k) e^{-\gamma \|\mathbf{x} - \mathbf{c}_k\|^2}\}_{k=1}^{K}$. The parameter γ is set to the mean of all per-component variance of all the training data assigned to all the leaves in our forest. As with *VLAD*,

the global descriptor is a concatenation of all K vectors.

We use L2-normalization with all *BoW* encodings [2], and square-root normalization followed by L2 normalization for *VLAD*.

3.3. Importance-weighted Aggregation

Most existing feature encoding schemes treat each local descriptor with equal importance. However, descriptors from parts of the image that contain *background* or irrelevant objects add noise to the global image descriptor which can hurt classification accuracy. In this section, we propose a technique that given a descriptor $\mathbf{x}$, predicts an *importance weight* $\omega(\mathbf{x})$ that will determine its contribution towards the global image descriptor. The idea is to learn a classifier that distinguishes *track data* (i.e our discriminative features on landmarks) from the rest, and use the predicted confidence value to derive this weighting. We implement this using a second random forest, trained using all the track descriptors $\mathcal{D}_s$ as positive examples and a large set of descriptors extracted from generic images without landmarks as negatives. Given a descriptor $\mathbf{x}$, this forest predicts the posterior probability $p(\mathbf{x})$ that $\mathbf{x}$ is a *track* (i.e. is similar to a landmark feature), by using the empirical distribution of the labels of the training examples assigned to the predicted leaf. We finally threshold this probability in a soft manner using a sigmoid. More formally, the importance weight is defined as $\omega(\mathbf{x}) = 1/(1 + e^{-\alpha p(\mathbf{x})+\beta})$. We call this *Importance-Weighted Aggregation* (*IWA*). The importance weighted *BoW* aggregation quantizes $\mathbf{x}$ as before, but now computes the histogram by accumulating the $\omega(\mathbf{x})$ contributions. Similarly, *VLAD* is modified to aggregate the scaled difference vector for a descriptor $\mathbf{x}$, obtained using the scale factor $\omega(\mathbf{x})$.

4. Experiments

We evaluated our approach on the public LANDMARK-3D benchmark [11] and a new large-scale dataset that we created and refer to as LANDMARK-620. These datasets will be described in detail in Sec. 4.3 and 4.4 respectively.

4.1. Methods

We have compared our novel codebook learning approach to the supervised and unsupervised codebook learning techniques listed below.

- **RFT**: Our random forest codebooks trained on descriptor tracks obtained from SfM (see Sec. 3).
- **KM** (K-means clustering): We vary the number of desired clusters K, to obtain codebooks of different sizes.
- **HKM** (Hierarchical K-means [22]): K-means clustering is done recursively to hierarchically partition feature space. *HKM* is faster than *KM* at quantizing features, as code assignment takes time logarithmic in tree depth. We tried several tree depth and branch factor combinations, and kept the best configurations.
- **ERC** (Extremely Randomized Clustering forests [21]). As mentioned earlier, this codebook learns a random forest using image category labels.
- **IL** (Info-Loss [15]): These codebooks are also trained discriminatively with image category labels, using empirical information loss minimization for learning a better nearest neighbor quantizer.

Some of these codebooks were used with bag of words histogram (**BoW**) and **VLAD** encoding methods. Different combinations of codebooks and encodings are denoted as *A-B*, where *A* and *B* refer to the codebook and encoding method respectively. In addition, our **RFT** codebook was used with **VLADsa**, the soft-assignment version of **VLAD** described in Sec. 3.2. The suffix **IWA** indicates that importance-weighted aggregation is used (e.g., *RFT-BoW-IWA* refers to the *RFT* codebook used with importance-weighted *BoW* encoding).

4.2. Experimental setup

Feature descriptors. In our experiments, all images exceeding 2 MPixels are resized to this maximum size. Scale-invariant DoG keypoints [19] are extracted and 32-dimensional DAISY descriptors [31] are computed for these keypoints. We also compute SIFT descriptors [19], using VlFeat [30]. In preliminary experiments on LANDMARK-3D, we found the classification accuracy using DAISY to be higher than SIFT by 10%. Hence, DAISY was used in all subsequent experiments (including all baseline methods).

Structure from Motion pipeline: We have implemented a state of the art incremental SfM algorithm similar to [29]. During the first phase of pairwise image matching, the local descriptors extracted from the images are geometrically verified, and pairwise feature matches are computed; this involves feature descriptor matching [19] followed by RANSAC-based robust epipolar geometry estimation between camera pairs. Given multi-view correspondences, a seed camera pair is first reconstructed and new cameras and 3d points are incrementally added to it by alternating between camera pose estimation and 3D point triangulation. Multiple rounds of sparse bundle adjustment is performed to refine the 3D structure and all camera parameters.

Learning model. We used one-vs-all linear SVMs as our multi-class classifier, using LIBLINEAR [9], performing the prediction with a winner-take-all strategy. We used hyper-parameter $C = 50$, found by using 5-fold cross validation in preliminary experiments. Finally, we calculate the classification accuracy as the mean of the diagonal of the confusion matrix. We tried the SVM formulation proposed in [4], also used by [16], but did not see much difference in performance. In our experiments, we assess the quality of

all the methods by comparing the classification accuracy as a function of the codebook size.

Parameters. We set $\kappa = 75\%$ and $\rho = 1000$ in all the experiments. The tree height h and tree count T is varied and best settings are kept for a given codebook size. For LANDMARK-3D, we tried setting η from 200 to 2000 and obtained best results with $\eta = 400$. Similarly, best results with *IWA* were obtained with $\alpha = 15.0$ and $\beta = 0.5$. For LANDMARK-620, the following settings were used; $\eta = 400$, $\alpha = 10$, $\beta = 0.1$.

4.3. Experiments on Landmark-3D

Dataset. LANDMARK-3D was introduced by Hao *et al*. [11] and contains 45K images of 25 landmarks. The images were gathered from Flickr and then manually filtered. We perform all our evaluation on a test set created by randomly sampling 200 images per landmark, and using the rest of the images as training set for the codebook. A subset of the latter set, consisting of 100 examples per category, is used to train the final landmark classifier.

Random Forest implementation. We performed preliminary experiments on LANDMARK-3D to assess the effect of the new extensions proposed in Sec. 3 for training random forest-based codebooks. We learned our *RFT* and the proposed *ERC* first using conventional axis-aligned splits, and then with the random projections-median splits [7]. As shown in Fig. 2c, the latter consistently outperforms the conventional method. Therefore, in all subsequent experiments, *RFT* codebooks were trained using this method.

We also empirically found that the improved entropy estimator H_G proposed in [23] provides some boost in accuracy over the standard entropy estimator H_S normally used to train Random Forests. Therefore, we use the H_G estimator for all the experiments involving *RFT*.

The number of trees and the tree depth affects the codebook size, and for a target size, various combinations were tested and the one with the best classification performance was retained. Finally, we found that the number of splits and the number of tracks used to train RFT, have only a moderate effect on the quality of the classifier. Again we tried different combinations and we keep the best performing one. Note that the training and testing of our codebook is very efficient and is easy to parallelize, as the trees can be trained and traversed at test time independently.

Recall from Sec. 4.1 that while using a RF codebook we perform the encoding by traversing the trees to determine the region of the feature space a given local descriptor belongs to, and then using the associated centroid. Note that we also tried a more traditional approach, that consists of calculating the L2 distance from the descriptor to all the centroids of the leafs, picking the closest one. However we empirically found that the performance of the latter method was $\sim 1\%$ lower in accuracy.

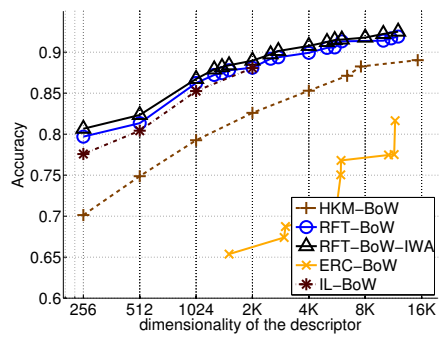
Landmark recognition. We first perform SfM reconstruction, then we randomly sample a variable range $200 - 1200$ tracks per landmark, and finally learn our codebook *RFT* and the Random Forest for the *IWA* as described above. We train the vocabularies based on *KM* and *HKM* using 2M local descriptors randomly sampled from 10K images; note that we tried to increase the quantity of training data but we did not see any additional boost in performance. *ERC* has been trained using 1M descriptors, sampled from 5K images. *IL* has been trained using only 100K descriptors, sampled from 1K images: we were not able to increase further the training set size and descriptor dimensionality D due to the high computational cost of this method, which is quadratic in D. Note that it has already been shown in[15] that *IL* outperforms *KM*, which is used as initialization for the training of *IL*. We perform evaluations using the encoding methods *BoW* and *VLAD*; the results of the experiments are shown respectively in Fig. 2a and 2b.

With a *BoW* encoding, the *RFT* codebook produces more accurate results compared to discriminative codebooks *ERC* and *IL*, as well as as unsupervised *HKM*. Moreover our method is considerably faster to train compared to *IL*, and the feature encoding step using our codebook is more efficient. With the *VLAD* encoding, all the results that make use of our codebook *RFT* are better then or equal to all the other comparisons, showing the effectiveness of our approach even in this scenario.

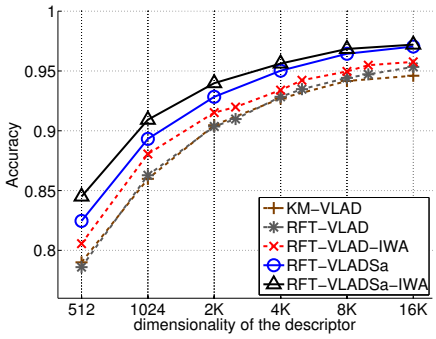
4.4. Experiments on Landmark-620

Dataset. LANDMARK-620 is a new dataset that we created for large-scale landmark recognition. We built the database by first manually listing 720 landmarks, each described with text phrases (e.g. "eiffel tower paris"). The landmarks we selected based on the rankings from [16, 37]. We downloaded up to 4000 images per landmark using multiple text queries on Flickr and web search engines. Unlike [16, 37, 12], we did not use geo-tags, since we found that geo-tags are not always representative of the image contents, which was also observed by [12]. A visual inspection of the images used by [16] (built using geo-tags) reveals the presence of many irrelevant images of people or other objects in the dataset, confirming that geo-tag queries did not yield a particular advantage over text queries.

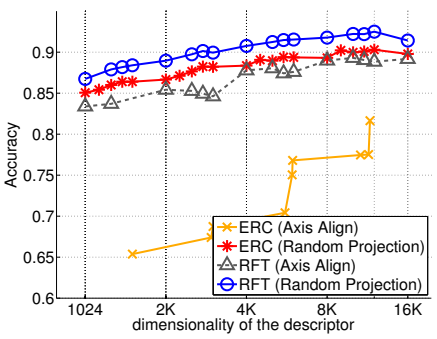
Since our images are also downloaded using text search, they were also contaminated with outliers. Thus, we found that it was crucial to filter out the irrelevant images from the database, as the unfiltered collection is simply too noisy to be used as ground truth and would provide unreliable training data and estimates of accuracy. Therefore we used state of the art SfM [29] to automatically remove outlier images from our contaminated dataset. We view this SfM filtering method as a surrogate for human labeling. Starting from an initial collection of 2.1M images, approximately 614K im-



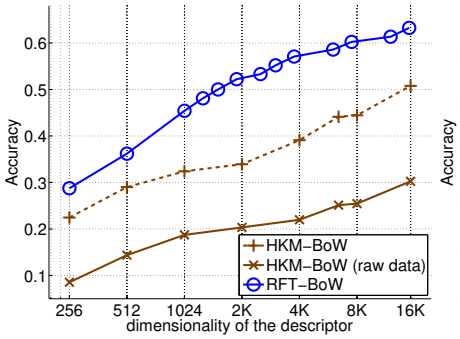
(a) LANDMARK-3D **(BOW)**



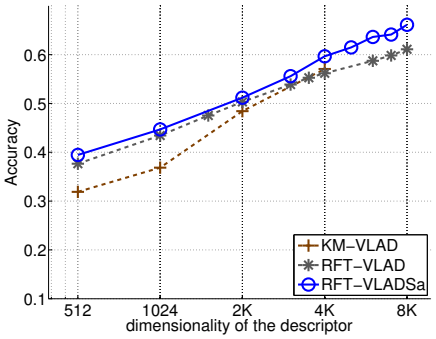
(b) LANDMARK-3D **(VLAD)**



(c) RFT+RP vs RFT+AA



(d) LANDMARK-620 **(BOW)**



(e) LANDMARK-620 **(VLAD)**

best 5 & *worst 5*	10 popular landmarks
Golden Gate Bridge (97%)	Trevi Fountain (77%)
Laocoonte Statue (97%)	Tyn Church (82%)
German-Fntn Istanbul (96%)	Piazza Navona (55%)
BowBridge CentralPark (95%)	Chichen Itza (71%)
Buland Darwaza (95%)	Arc de Triomphe (75%)
Brussels Grotemarkt (32%)	Taj Mahal (89%)
Bologna Duomo (31%)	Times Square (47%)
Catedral Cordoba (27%)	Pike Place Market (88%)
Capitolio la Havana (27%)	Trafalgar Square (51%)
Palma Cathedral (21%)	Pantheon (83%)

(f) PER-CLASS ACCURACY

Figure 2: *(a-b)*: Experiments on LANDMARK-3D, using various encodings. *(c)*: Classification accuracy with RF-based codebooks trained using different weak learners. *(d-e)*: Experiments on LANDMARK-620, using various encodings. *(f)*: Accuracies using *RFT+VLADSa* and dimensionality 8K on the top/worst five landmarks and a few popular landmarks.

ages were selected. Images that failed to match either other images or the 3D structure were discarded. We eliminated 100 landmarks for which we had fewer than 350 images after the filtering step. Finally, the filtered images were divided into training and test sets, that were used for all the comparisons (with the only exception of "HKM-Bow raw data" as described below). Although filtering was necessary for this dataset, it could seem to potentially introduce a bias in favor of our method, as the vocabulary construction relies on feature correspondences. However, even our experiments on Landmark3D, where no such filtering was applied and thus bias is certainly not present, show that our method outperforms all the baselines.

In summary, LANDMARK-620 has 620 landmarks and 614,315 images. We created a test set of 62,000 images, by randomly selecting 100 images per landmark, and used the remaining to train the codebook and the classifier that is trained using 100-200 images per category.

Landmark recognition. We now present the recognition results obtained on this challenging dataset. We first perform SfM again, this time *only* on the codebook training sets, obtaining a 3D reconstruction for each landmark. Note that each of them contains on average 172K 3D points; any landmark recognition system that requires to store the local descriptors of the models (e.g. [11]) would need 4.3 GB of memory using DAISY and 14.3 GB with SIFT (assuming a single feature vector for each 3D point), making these approaches not scalable to a large number of categories. We sample 400 tracks per category to learn our *RFT* codebook and the *IWA* random forest. We compare our codebook to *HKM* as its computational cost of aggregating local descriptor is comparable to ours. The method denoted as *[HKM-BoW (raw data)]* in Fig. 2d is our implementation of Li *et al*. [16], where the raw images downloaded from the Web, without *SfM* filtering, were used for training the classifier. Both versions of these *HKM* codebooks are trained using 5M local descriptors randomly sampled from 20K images. The results from experiments using the *BoW* and *VLAD* encodings, are shown in Figures 2d and 2e respectively.

We can see from the figures that we considerably outperform the *HKM-BoW (raw data)* baseline, showing the effectiveness of filtering the data by means of SfM, which creates a better training set for the recognition system. For both

BoW and *VLAD* we can see that our vocabulary outperforms all the other comparisons. In particular, with a descriptor of 15360 dimensions, *RFT+BoW* yields an accuracy of 63.25%, producing a +12.48% improvement over the baseline *HKM-BoW*. Increasing the size of the classifier training set from 100 examples per class to 200 we were able to further increase the accuracy of *RFT+BoW* to 68.76%. Using a more effective encoding method like VLAD, and a descriptor dimensionality of 8192, our method *RFT+VLAD* achieves an accuracy of 61.09%, and with *RFT+VLADSa* 66.10%. We provide 200 examples per class for the classifier, *RFT+VLADSa* achieves an **accuracy of 71.26%**, that represents the best result we provide.

5. Conclusions

We proposed a new, efficient algorithm to learn a random forest codebook from local feature descriptors, where feature correspondence information is given. It is applicable to large-scale landmark classification, where our approach uses discriminative feature tracks discovered using SfM. Our learned codebook outperforms existing discriminative codebooks. We believe this is due to our use of discriminative tracks obtained using SfM as well as our ability to exploit fine-grained descriptor labels during codebook learning. For memory-efficient landmark classification, our accuracy rates are much higher than previously reported in the literature [16]. Our codebook can be used with several feature encoding schemes, and the aggregation step is computationally efficient. While in this paper we have applied our codebook learning method to landmark classification, we expect that our approach can be used successfully in other scenarios where a set of correspondences among images is available. In the future, we will investigate this research direction by learning codebooks for image retrieval.

References

[1] H. Cai, F. Yan, and K. Mikolajczyk. Learning weights for codebook in image classification and retrieval. In *CVPR*, 2010.

[2] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[3] D. Chen, G. Baatz, Köser, Tsai, Vedantham, Pylvanainen, Roimela, Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, 2011.

[4] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In *COLT*, 2000.

[5] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, 2012.

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[7] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *STOC*, pages 537–546, 2008.

[8] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM Trans. Graph.*, 31(4), 2012.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9, 2008.

[10] J. C. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

[11] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu. 3d visual phrases for landmark recognition. In *CVPR*, pages 3594–3601, 2012.

[12] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[13] H. Jegou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[14] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. *ECCV*, pages 748–761, 2010.

[15] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *PAMI*, 31(7), 2009.

[16] Y. Li, D. Crandall, and D. Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, 2009.

[17] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, pages 791–804, 2010.

[18] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV (1)*, pages 15–29, 2012.

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

[20] A. Mikulík, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *ECCV*, pages 1–14, 2010.

[21] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *PAMI*, 30(9):1632–1646, Sept. 2008.

[22] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, pages 2161–2168, 2006.

[23] S. Nowozin. Improved information gain estimates for decision tree induction. In *ICML 2012*, 2012.

[24] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[25] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV (4)*, 2010.

[26] J. Philbin, Chum, Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[27] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *ECCV (1)*, 2012.

[28] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

[29] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from Internet photo collections. *IJCV*, 80(2):189–210, November 2008.

[30] A. Vedaldi and B. Fulkerson. VLFeat, http://www.vlfeat.org/.

[31] S. A. J. Winder, G. Hua, and M. Brown. Picking the best daisy. In *CVPR*, pages 178–185, 2009.

[32] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *ICCV*, 2005.

[33] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum. A multi-sample, multi-tree approach to bag-of-words image representation for image retrieval. In *ICCV*, pages 1992–1999, 2009.

[34] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *CVPR*, 2007.

[35] S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao. Generating descriptive visual words and visual phrases for large-scale image applications. *IEEE Transactions on Image Processing*, 2011.

[36] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, pages 809–816, 2011.

[37] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, 2009.