Predicting Good Features for Image Geo-Localization Using Per-Bundle VLAD

Hyo Jin Kim, Enrique Dunn, Jan-Michael Frahm
Department of Computer Science, University of North Carolina at Chapel Hill
{hyojin,dunn,jmf}@cs.unc.edu

Abstract

We address the problem of recognizing a place depicted in a query image by using a large database of geo-tagged images at a city-scale. In particular, we discover features that are useful for recognizing a place in a data-driven manner, and use this knowledge to predict useful features in a query image prior to the geo-localization process. This allows us to achieve better performance while reducing the number of features. Also, for both learning to predict features and retrieving geo-tagged images from the database, we propose per-bundle vector of locally aggregated descriptors (PBVLAD), where each maximally stable region is described by a vector of locally aggregated descriptors (VLAD) on multiple scale-invariant features detected within the region. Experimental results show the proposed approach achieves a significant improvement over other baseline methods.

1. Introduction

Image geo-localization is the process of determining the capturing viewpoint's positioning w.r.t. a geographic reference [43]. The recent availability of large scale geo-tagged image collections, enables the use of image retrieval frameworks to transfer geo-tag data from a reference dataset into an input query image. Applications of these capabilities include adding and refining geotags in image collections [15, 41], navigation [26], photo editing [44], and 3D reconstruction [11]. However, geo-localization of an image is a challenging task because the query image and the reference images in the database vary significantly due to changes in scale, illumination, viewpoint, and occlusion.

Image retrieval techniques based on local image features [27] can achieve increased robustness against photometric and geometric changes [24, 42]. However, not all local features are useful for geo-localization [21]. For example, features extracted from transient scene elements (pedestrians, cars, billboards) and ubiquitous objects (trees, fences, signage) can introduce obfuscating cues into the geo-localization process. Many approaches have been proposed to address this issue by focusing on the *uniqueness* of a feature by removing and reweighting non-unique features within the reference data [21, 32] or in the query image [2]. Indeed, unique features are helpful, but a non-unique feature may actually help increase the chance of correct localization, either by itself or in combination with others.

We exploit a data-driven notion of good features for geo-localization. That is, we aim to foster features having relatively high matching scores in correct localization outcomes, in contrast to their relatively low score for negative outcomes. Further, we cast feature score prediction as a classification problem, assuming the characteristics are shared in a reasonably-scaled geographic region. We use a separate set of geo-tagged Internet images to generate training data, computing matches against database images. To cope with noise and high intra-class variation among the training data, we adopt recent bottom-up clustering techniques for visual element discovery [8, 9] that involves iterative training of linear support vector machines (SVM). At the query phase, the algorithm selects features in a query image prior to the geo-localization process by accumulating predictions from the bank of linear SVMs. Our results show improved performance is achieved by using only features that are predicted as useful, while reducing the number of features significantly.

The feature representation for such a task should not only be robust to photometric and geometric changes, but also have a high discriminative power as we want to learn features over a large area, *e.g.* a city. Therefore, we avoid using low-level features for learning, which are hard to be discriminative over a large area. We propose a per-bundle vector of locally aggregated descriptors (PBVLAD) for feature representation, where each maximally stable (MSER) [28] region is described with a vector of locally aggregated descriptors (VLAD) on multiple scale-invariant features detected within the region. This allows us to represent multiple features with a fixed-size vector such that it can be used in various classification methods such as an SVM. We show in the experiments that this feature representation has significant improvement over low level features in both learning to predict features and retrieving images.

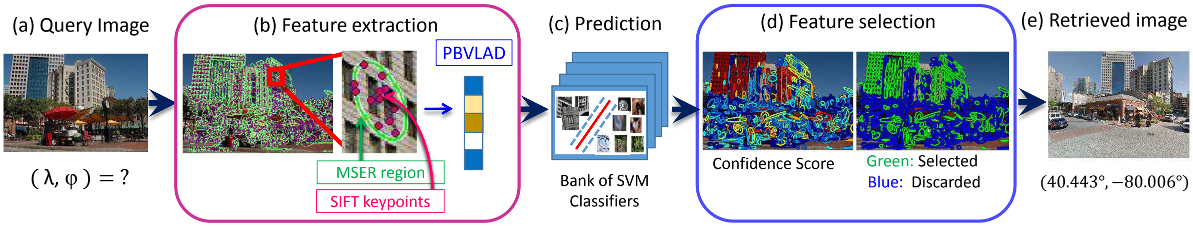

Figure 1: Overview of our approach. From an input query image with unknown geo-location (a), MSER regions and SIFT keypoints forms a bundled feature [40], and consequently represented by PBVLAD (b). Features go through a pre-trained bank of SVMs that outputs binary predictions about a feature being "good" for geolocalization (c). Predictions are accumulated to compute confidence scores for each feature (d, left). Features with high scores are selected for geo-localization (d, right). Retrieved geo-tagged image is shown in (e).

Our contribution is two-fold: (1) We offer a way to predict features that are good in a data-driven sense for geo-localization in a reasonably-scaled geographic region. We show that by selecting features based on predictions from learned classifiers, geo-localization performance can be improved. (2) We propose per-bundle vector of locally aggregated descriptors (PBVLAD) as a novel representation for bundled local features that is effective for both learning to predict features and image retrieval.

2. Related work

There are two main categories in image geo-localization for street-level input images. Our method falls into the category of image-retrieval-based methods where a geolocation of the image is approximated by identifying geo-tagged reference images depicting the same place [3, 5, 16, 35]. The other is to estimate the full camera pose of the query image using a 3D structure-from-motion model constructed from reference images [13, 17, 25, 31], which is limited to places with a dense distribution of reference images.

Our work is mostly related to recent works attempting to select features that are *geographically discriminative* by taking advantage of geotags in the database. Schindler *et al*. [32] build a vocabulary tree using only unique features that appear at each location. Arandjelovic and Zisserman [2] use distribution in the descriptor space as a measure for distinctiveness. Knopp *et al*. [21] refine the database by removing features that match to faraway places. Rather than finding features unique to specific places, Doersch *et al*. [9] find image patches that also occur frequently in a geographical region, and unique with respect to other geographic regions. While these methods focus on the uniqueness of a feature, we focus on features that explicitly contribute to geo-localization either positively or negatively. Although unique features do characterize a location, it may be risky to discard all non-unique features, some of which may contribute to correct retrieval by having high matching score in the correct location than in false positives.

Some cast the localization problem as a classification problem where visual words are weighed according to their importance to specific locations [4, 12]. Conversely, we train classifiers to predict whether a feature is useful for geo-localization over a larger scale of geographic region, utilizing a separate set of geo-tagged images from photo sharing websites taken in a city to generate our training data. Based on the predictions, we select features prior to geo-localization. We show that better performance is achieved without using all features. It is also more scalable as the training images can be much more sparse than the reference images, with the assumption that these characteristics are shared among images in the same geographic region.

In the fields of image retrieval, there is a large body of literature on feature selection and weighting [30, 36, 38, 45]. The closest work to ours is [33], which tries to find the importance of each feature by training a per-exemplar SVM on a given query image with hard negative mining. While this method can be effective, it is time consuming as a fresh model is trained every time. In constrast, we refine and organize the outcomes of geo-localizing training images in offline, and use this knowledge for selecting features.

In terms of selecting features in advance to matching in a data-driven way, our work is closely related to [14], but with different focuses. Whereas [14] tries to predict features that are likely to form a match, we predict features that contribute to correct geo-localization. As we show in our experiment, not all matches are useful for geo-localization.

Applying VLAD to local regions in previous work was either based on tiles from rectangular grids [1] (as in spatial pyramids [23]), or on bounding boxes [39], which are not robust to geometric changes. We propose to use VLAD for representing a bundled feature [40], which consists of SIFT keypoints and an MSER region that are both repeatable, thus resulting our PBVLAD to be robust to geometric and photometric changes.

3. Proposed approach

The overview of our approach is shown in Figure 1. In this section, we first introduce our proposed feature representation for image retrieval and training calssifiers (Sec. 3.1). We then illustrate our training framework for automatically generating training data and training a bank of SVMs for predicting good features for geo-localization (Sec. 3.2).

3.1. Per-bundle VLAD for feature representation

We want to identify parts of an image that are useful for geo-localization, using a discriminative classification method such as SVM. However, it is a hard problem to learn such characteristics given a low level description of a corner or a blob. Thus, we propose per-bundle vector of locally aggregated descriptors, namely PBVLAD. The key idea is to use groups of low level features, and describe them in a vector with a fixed-size that allows it to be compared in standard distance measures, and enables it to be used for various classification methods.

The concept of a *bundled feature* was proposed by Wu *et al.* [40] for retrieving partial-duplicate images. By bundling multiple SIFT features detected in the same MSER regions, the discriminative power is increased while still being repeatable, as both components are robust to photometric and geometric changes. The original representation was a concatenation of quantized SIFT features, which changes in length as a MSER region can contain different number of SIFT features. The similarity between two bundled features was measured by computing intersection between them. In this paper, we propose to describe a bundled feature with a vector of locally aggregated descriptors (VLAD) [19]. This representation produces sparse vectors with a fixed-size that is convenient for comparing distances and training classifiers such as SVM. Compared to the bag-of-words (BoW) representation, VLAD can have a much smaller dimension while maintaining high discriminative power, and it can be further quantized without significant loss in performance. Note that Min-hash sketches can also provide a compact representation [6], but it has a comparably low recall and a limited number of applicable classification methods as standard distance measures cannot be applied.

Let R and S denote the MSER regions and SIFT features detected in image I, respectively. Each MSER region $r \in R$ contains a set of SIFT features $B \subset S$ that are detected within that region $B = \{s = (d, l) | l \in r\}$, where d and l denote the descriptor and the location of the SIFT feature. B is called a bundled feature [40]. For a bundled feature B_a, its associated SIFT features $s_a = (d_a, l_a) \in B_a$ are each assigned to a visual word of a coarse vocabulary W via nearest neighbor search such that $NN(d_a) = \underset{w}{\operatorname{argmin}} ||d_a - c^w||$, where c^w is the centroid of the visual word w. The subvector of per-bundle VLAD that corresponds to the visual word w, denoted as p_a^w, is obtained as an accumulation of differences between d_a's that are assigned to w and the centroid c^w. As proposed in [7], we normalize the differences (*i.e., residuals*), so that each contribution of SIFT descriptor d_i to the vector p_a^w are equal. This is to limit the effect of possible noise, although bundled features are robust to photometric and geometric changes.

$$p_a^w = \sum_{d_i : NN(d_i) = w, d_i \in B_a} \frac{d_i - c^w}{\|d_i - c^w\|} \tag{1}$$

The final representation is the concatenation of the vectors p_a^w followed by L_2 normalization.

$$p_a = \left[p_a^1, p_a^2, ..., p_a^{|W|}\right] \tag{2}$$

We tested multiple normalization schemes [1, 19], but the combination of residual- and L_2- normalization performed the best in our data. The PBVLAD representation of corresponding bundled features are visualized in Figure 2.

Similarity metrics. The similarity between two PBVLAD is computed as their dot product $M(p_a, p_b) = p_a \cdot p_b$. Figure 3 depicts the matched feature regions of two corresponding images. We define the *matching score* f of a feature p_q in a query image I_q to a reference image I_r as the maximum possible similarity between p_q and features in I_r. The *image similarity* Sim between a query image I_q, and the reference image I_r becomes the sum of matching scores of individual features $p_q \in I_q$ with respect to I_r.

$$f(p_q, I_r) = \max_{p_r \in I_r} M(p_q, p_r), \tag{3}$$

$$Sim(I_q, I_r) = \sum_{p_q \in I_q} f(p_q, I_r) \tag{4}$$

We use above image similarity measure to retrieve reference images that best matches the query image.

For efficient nearest neighbor search in the reference data, we reduce the dimension of raw PBVLAD using principal component analysis (PCA). Instead of performing PCA on a whole vector, we do on a per-visual-word basis by performing PCA on subvectors p^w that are generated from each visual word w. We do this in order to preserve the characteristics of each visual words that might be lost due to the overall sparsity of the vector. In our implementation, a coarse vocabulary of 128 visual words was used, yielding 16,384-dimensional raw PBVLAD's. The dimension is then reduced to 2,048 by performing PCA on 128 visual words and taking the top 16 components of each. Note that PBVLAD matching can be efficiently indexed using product quantization [18]. Henceforth, the term *feature* will refer to PBVLAD representation of a bundled feature.

3.2. Predicting good features for geo-localization

Automatic training data generation. Given an arbitrary set of geo-tagged images $\mathcal{I}_t = \{I_t\}$, we want to *automatically* generate good/bad training examples of features for geo-localization using only their associated GPS locations. Rather than having assumptions about good and bad features for geo-localization, we want to find them in a data-driven way. This enables our method to adapt to various

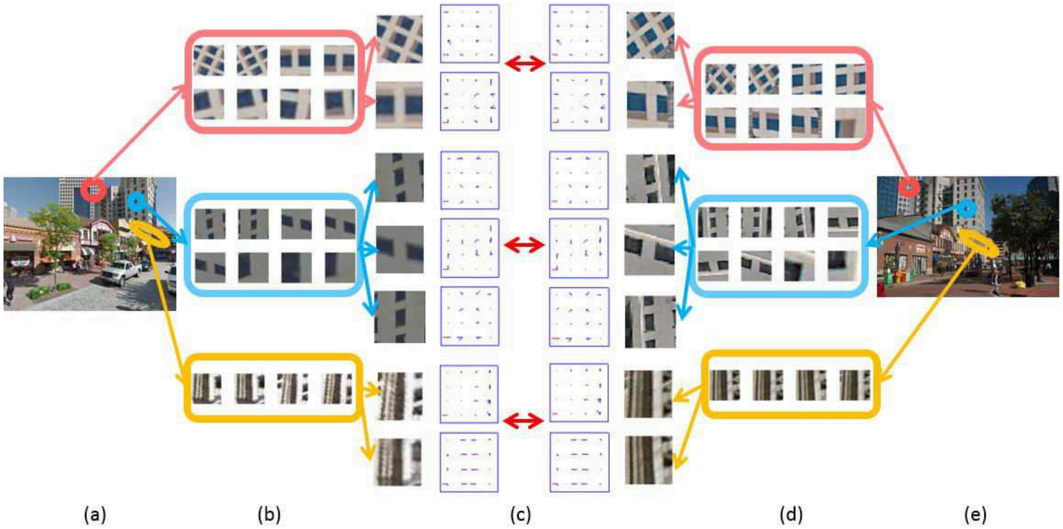

Figure 2: PBVLAD representation of corresponding bundled features. (a,e) Two different images depicting the same place. (b,d) Multiple SIFT features are bundled within MSER regions. (c) Each bundle is represented with VLAD. We follow the visualization scheme of [19] where subvectors are represented in 4x4 spatial grid with red representing negative values. Note that only non-sparse blocks that correspond to overlapping visual words of two bundles are visualized due to space limit.

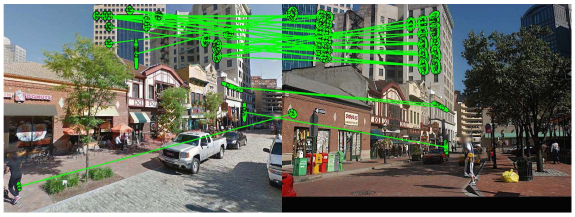

Figure 3: Matching with PBVLAD with similarity threshold 0.5

geographical regions. For each image in the training set, we retrieved top $n = 100$ images from the reference set $\mathcal{I}_r = \{I_r\}$ using image similarity defined in Eq. 4. We investigate whether a feature in a training image $p_t \in I_t$ is *explicitly* contributing to the correct retrieval of the ground truth image. To this end, we compare a feature's matching score to a ground truth reference image $f(p_t, I_{GT})$, against the matching score to a falsely retrieved images $f(p_t, I_{FP})$. Given that the overall image similarity between two images is the sum of individual matching scores (Eq. 4), this comparison helps us differentiate good features based on their individual contribution. If the difference between two values $|f(p_t, I_{GT}) - f(p_t, I_{FP})|$ is greater than a certain threshold, we include the feature into the training set, assigning positive label when $f(p_t, I_{GT}) > f(p_t, I_{FP})$, negative label otherwise. This process is depicted in Figure 5(a-d) and provides the initial positive and negative training feature set for data-driven visual component discovery.

Closed-loop training of SVM classifiers. The automatic labeling approach above can sometimes generate contradictory labels for the features with similar appearance. This commonly occurs in visual elements that appear in both the transient and the static objects. In Figure 4, for example, text on buses (b) and t-shirts (e) is assigned a negative label, while text on buildings and store signs (d) belongs to the positive set. A limited field-of-view overlap between a training image and a ground truth image can also lead to such contradictory labelings. Windows on the same building, for instance, can be assigned to different labels due to their visibility in the ground-truth reference image I_{GT}.

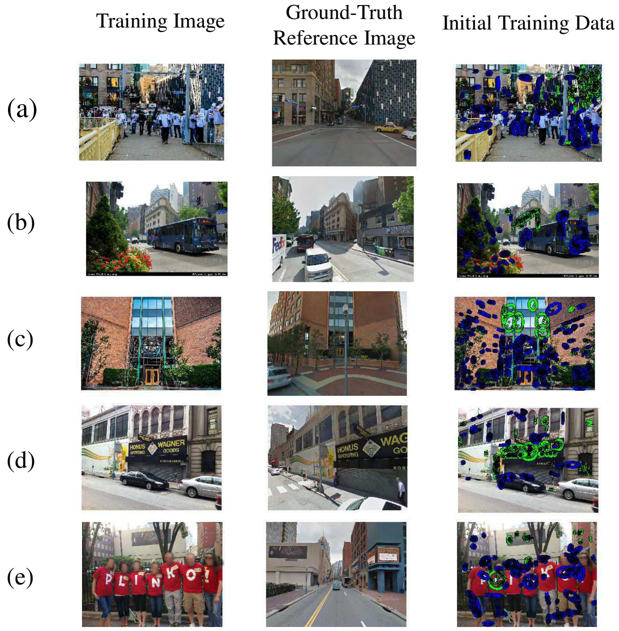

Figure 4: Initial training data generation. Positive and negative training examples are depicted in green and blue, respectively.

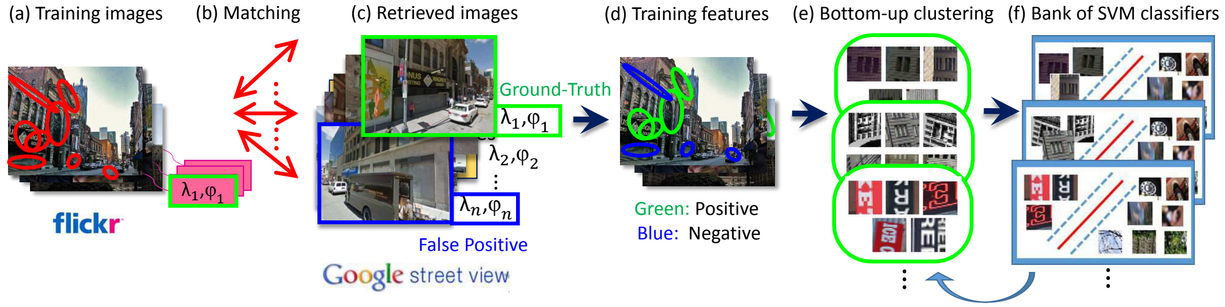

Figure 5: Overview of our training framework. For each training images that have GPS-tags (a), we retrieve top n images from the reference set (b-c). Positive labels are assigned to features that have higher matching score in the ground-truth reference image than in the falsely retrieved reference images, with a margin greater than $thres$. Negative labels are assigned in a similar manner. (d). To handle noise and high intra-class variation, we use bottom-up clustering technique, refining the positive set as well as training SVMs iteratively (e-f).

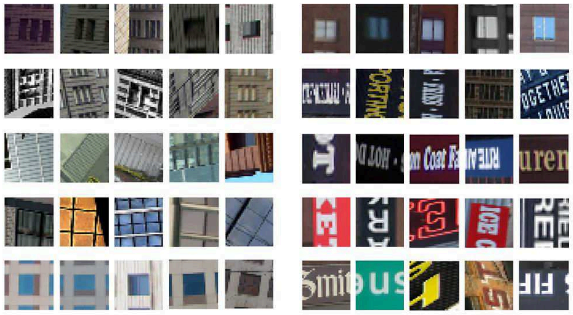

Figure 6: Top elements in the final clusters with a high ratio of positive labels. Each half row corresponds to different clusters.

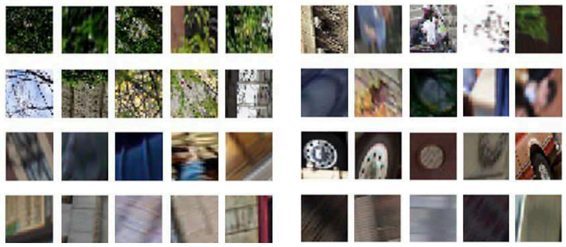

Figure 7: Final negative set elements aligned according to their initial clusters. Each half row corresponds to different clusters.

Such contradictory labeling on similar features limits the prediction accuracy.

On the other hand, there exists high intra-class variation in both the positive and negative classes: Windows have different appearances from text, for example, yet features from both appear in the same class. Training a single classifier over the entire data may be negatively affected by such intra-class variation.

To solve the problems of contradictory labelings and intra-class variation, we perform bottom-up clustering [9] on the initial training feature set. By doing so, we obtain clusters of training examples whose appearances and the labels are most consistent, as well as a bank of linear SVM classifiers that are trained within each cluster. Each training example constructs a cluster by finding k nearest neighbors in the training set. Redundant sets whose top ranked elements overlap with existing sets are eliminated. If a cluster has a high ratio of negative labels, the negative examples in that cluster are assigned to the final negative set $\mathcal{N}$, and the positive ones are discarded. For the remaining clusters C_i, a linear SVM is iteratively trained on the positive examples in each cluster, using $\mathcal{N}$ as the negative set for hard negative mining (Figure 5(e-f)). As the SVM uses its true-positive firings for the re-training in the iterative procedure, clusters are left with features having consistent appearances and labels. Similar to [9], the clusters and $\mathcal{N}$ are divided into three sets to avoid overfitting. We only keep the SVM classifiers with an accuracy rate greater than 0.8. Finally, we remove redundant classifiers whose weight vectors have a high cosine similarity with that of other classifiers as in [20]. Examples of top elements in $\mathcal{C}_i$ are shown in Figure 6. Figure 7 shows elements in $\mathcal{N}$, which are aligned according to their initial clusters. Interestingly, although our approach makes no assumption on features that are useful for geo-localization, we can observe semantic relationships emerge through the learning process. Namely, windows, characteristic wall patterns, and letters on signage are detected as positive elements, while features from trees, people, car wheels, pavements, and edges are considered as negative elements.

In the querying phase, we feed query image features into the bank of linear SVM classifiers. We accumulate predictions from each classifier to compute the confidence score of a feature being good for geo-localization (Figure 10 (b)), weighting them using the $discriminativeness$ [34] of the classifier, which is the ratio of number of firings in its cluster $\mathcal{C}_i$ over that in the entire training set, in order to compensate for the distribution of visual elements that each cluster spans. We discard features with a low confidence score and keep only the remaining features for performing geo-localization (Figure 10 (c)).

Implementation details. For generating the training set, we define the I_{GT} image set as reference images that are within 50 meters from the given GPS location and passed geometric verification w.r.t. the training image by fitting a fundamental matrix. For I_{FP}, we took reference images that are retrieved within the top n (n =100), and at least 270m away from the given GPS location. This accounts for both user-provided geo-tag errors and the fact that large, symmetric buildings are often observable from extended areas. Before comparing $f(p_t, I_{GT})$ and $f(p_t, I_{FP})$, we normalize the matching scores by multiplying $\frac{1}{\max(f)}$ to compensate for a non-uniform distribution of features. For training and predicting, we separated features into three scale levels based on the size of the MSER, as we observed that the distribution of positive and negative PBVLAD features varies in different scales. The number of SVM classifiers used in each level were 35, 150, and 25.

4. Experiments

4.1. Image Geo-localization

Dataset. For the reference image set $\mathcal{I}_r$, we collected 27,520 geo-registered Google Street View images covering the Pittsburgh (U.S.) area. These images contain 8 overlapping perspective views extracted from the spherical panoramas in two different yaw directions, to capture both eye-level street views and the higher parts of the building in urban environments. This setting is similar to those used in [9, 12, 37]. The co-located GPS-tagged training image set $\mathcal{I}_t$, comprising positive and negative training data $\mathcal{C}_i$'s and $\mathcal{N}$ for learning, was downloaded from Flickr and consisted of 850 images that were successfully registered to the nearest elements in $\mathcal{I}_r$ through geometric verification. The test image set $\mathcal{I}_q$ was formed by 145 internet collection images from the query set of [42] with manually verified GPS-tags.

Results. We compare the proportion of correctly localized image among a ranked list of top n candidates. All of our results are without post-processing such as geometric reranking [29]. We consider an image to be localized if it is within 35m from the ground truth location. For a baseline, we compare with our implemented version of [42] We also compare a variant of [42] with SIFT feature selection by pre-trained linear SVM in a procedure similar to our selection of PBVLAD features (SIFT Select).

Figure 8 depicts how our systems with selected PBVLAD (PBVLAD Select) and all PBVLAD (PBVLAD All) consistently outperform the baseline methods. Feature selection is more successful in PBVLAD than SIFT. The performance of using selected features is consistently better than using all features in PBVLAD, whereas this behavior alternates when considering SIFT features.

The performance at the top of the shortlist (n = 1) displayed in Table 1. Our method achieves a recall of 64.83% using all features and improves to 68.28% with selected features, while the best baseline method (SIFT Select) obtains 49.66%. We also tested the performance of the system using the same number of PBVLAD features as our selection framework, but that are picked randomly (PBVLAD Random). Its poor recall rate supports the effectiveness of our selection mechanism, illustrating how simply selecting fewer features does not generally improve the performance. Moreover, we also tested with the features that are *not* selected by our framework (PBVLAD Select$^{\mathsf{C}}$) to illustrate how discarded features are in general detrimental to the geo-localization. The random chance of retrieving correct images is 0.2 %, which reflects difficulty of the dataset.

Method	% Correct
PBVLAD All	64.83
PBVLAD Select	**68.28**
PBVLAD Random	33.38
PBVLAD Select$^{\mathsf{C}}$	19.31
SIFT All [42]	49.66
SIFT Select	46.90
Chance	0.20

Table 1: Proportion of correctly localized images at top 1

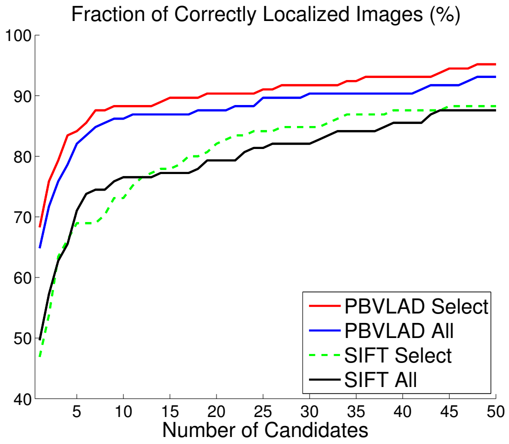

Figure 8: Geo-localization performance

Figure 9 shows examples of our results using PBVLAD. The top four retrieved images are shown for each query image. As can be seen, our method retrieves correct reference images despite partial occlusions and changes in viewpoint, illumination, and scale. Figure 10 depicts other examples where PBVLAD Select outperforms PBVLAD All.

We attribute the enhanced performance of PBVLAD-based retrieval to the increased discrimination power provided by aggregated features. Figure 11 (b) illustrates the maximum obtained feature similarity score for the features within a query image (a) w.r.t. the entire reference dataset. We can observe that PBVLAD features in foliage image regions are not highly matched to the reference set. Where individual SIFT features may have many similar features in

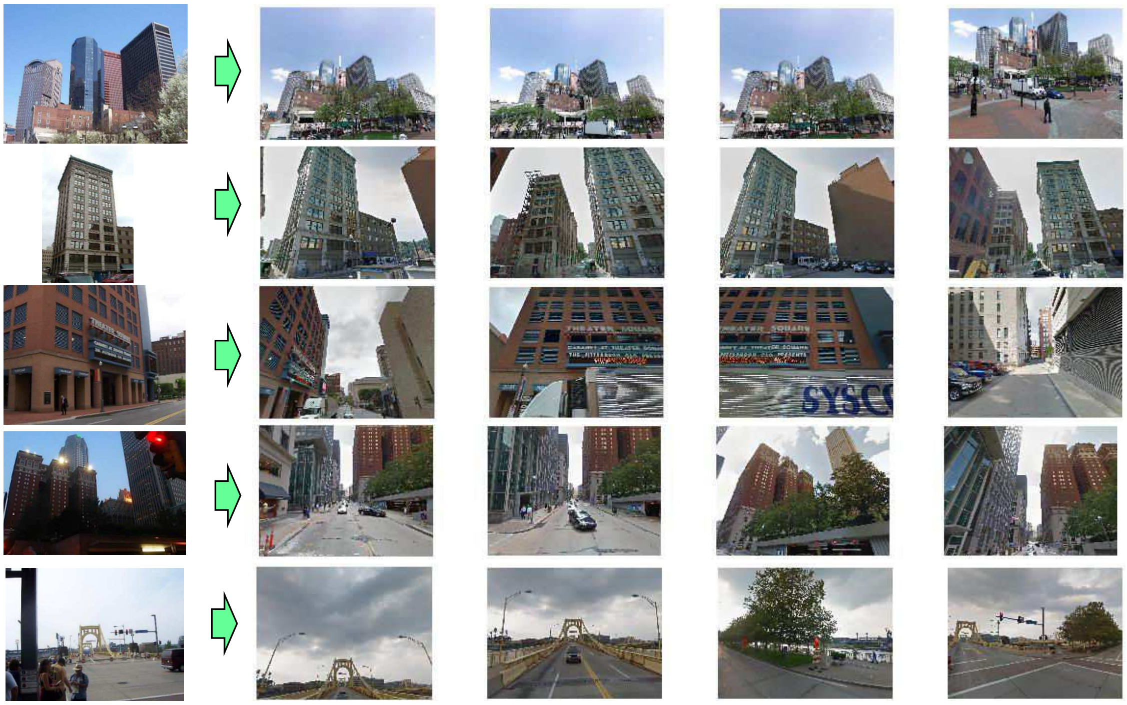

Figure 9: Example result (left) Query images, (right) Top four retrieved images using our proposed PBVLAD. Query images are of various sizes.

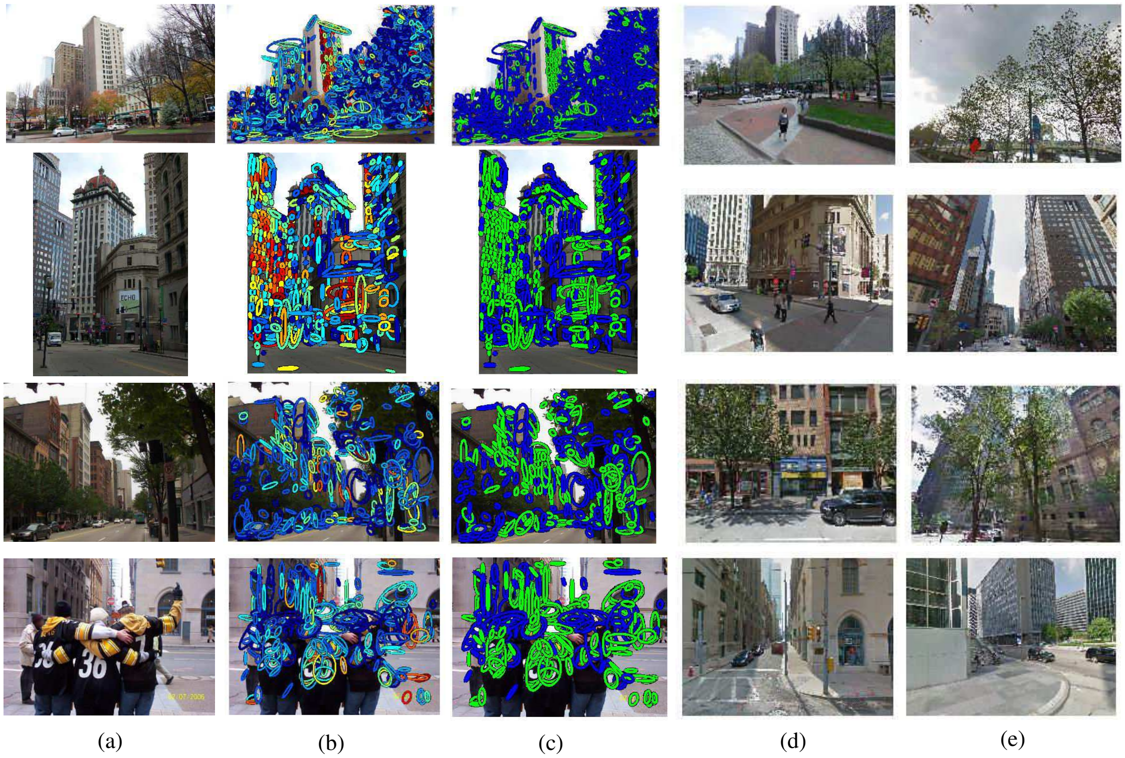

Figure 10: Qualitative comparison of retrieved image using selected PBVLAD and using all of the features. (a) Query image (b) Heat map representation of confidence being a good feature (c) Selected features (green:selected, blue:discarded.) (c) retrieved image using selected features (d) retrieved image using all features.

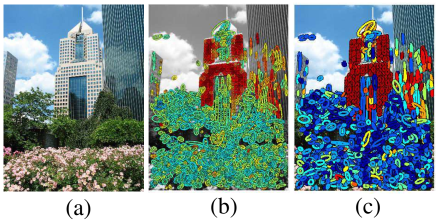

Figure 11: (a) Query image (b) Heat map of maximum matching scores $\max_{I_r}(f(p_q, I_r))$ of each features p_q. (c) Confidence scores

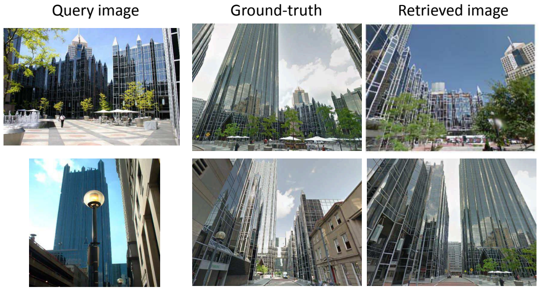

Figure 12: Failure cases. Retrieved images are more than 100m away from ground-truth locations.

the dataset, the analysis of their local ensembles is more discriminative. Moreover, our final predicted feature scores (c) illustrate how our framework discriminates good features prior to direct feature similarity estimation.

Failure cases. There are many cases where the ranked list contained the same building in the query image, but at different locations. The first and second row of Figure 12 show such examples. This occurs often for images depicting a large and symmetric buildings. In many cases, the building itself looked more similar to the retrieved image than the ground-truth reference image. Another observation is that when it comes to severe scale changes, the number of SIFT keypoints detected within the MSER region is reduced due to lack of details. In such cases, it becomes hard to match a PBVLAD as many of its group members are missing. This could be alleviated by using spectral SIFT [22], or by only including keypoints detected within some scale range from the MSER region similar to [6].

4.2. PBVLAD for general image retrieval

We evaluate PBVLAD as a descriptor for image retrieval on the Oxford5k Buildings dataset [29]. Table 2 compares our method against state-of-the-art image retrieval approaches [10, 19], which includes VLAD, Fisher vector (FV), and a bag-of-words baseline. The evaluation was performed without dimensionality reduction for all methods. PBVLAD shows competitive performance to other state-of-the-art descriptors. Table 3 shows the effect of dimension reduction using PCA. The decrease in the performance is not significant until the dimension is reduced to 12.5%.

Descriptor	# Vocabulary	mAP
BoW [19]	200,000	0.364
BoW [19]	20,000	0.319
Fisher [19]	64	0.317
VLAD [10]	128	0.339
PBVLAD	128	**0.369**

Table 2: Comparative image retrieval performance of PBVLAD on the Oxford 5k dataset. The accuracy is measured by the mean Average Precision (mAP). All descriptors are uncompressed.

	Full	Dim Reduced			
Dim	16384	8192	4096	2048	1024
mAP	**0.369**	0.364	0.334	0.264	0.210

Table 3: Retrieval performance of PBVLAD on Oxford 5k dataset, before and after the dimensionality reduction using PCA. The accuracy is measured by the mean Average Precision (mAP)

5. Conclusion

In this work, we proposed per-bundle vector of locally aggregated descriptors (PBVLAD) for maximally stable regions in an image. PBVLAD provides a convenient and effective representation for classification of grouped local features. Using this descriptor and a geo-tagged internet image collection, good/bad features for geo-localization were exploited with the notion of good/bad being explicitly defined in terms of the feature's contribution to the retrieval process. To remove noisy labels and deal with the large intra-class variation, bottom-up clustering was performed, generating a bank of SVM classifiers. At the query phase, outputs of each classifiers were accumulated to select good features. The experimental results show an improvement in the geo-localization accuracy when only good features predicted by our algorithm were used.

Acknowledgments Supported by the Intelligence Advance Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL), contract FA8650-12-C-7214. The U.S Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government. The authors would also like to thank Alex Berg and Amir Zamir for helpful discussions.

References

[1] R. Arandjelovic and A. Zisserman. All about vlad. In *CVPR*, 2013. 2, 3

[2] R. Arandjelović and A. Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*,

2014. 1, 2

[3] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large scale visual geo-localization of images in mountainous terrain. In *ECCV*. 2012. 2

[4] S. Cao and N. Snavely. Graph-based discriminative learning for location recognition. In *CVPR*, 2013. 2

[5] D. Chen, G. Baatz, K. Koser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *CVPR*, 2011. 2

[6] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, 2009. 3, 8

[7] J. Delhumeau, P. Gosselin, H. Jégou, and P. Pérez. Revisiting the vlad image representation. In *ACMMM*, 2013. 3

[8] C. Doersch, A. Gupta, and A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013. 1

[9] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros. What makes Paris look like Paris? In *ACMTOG*, 2012. 1, 2, 5, 6

[10] C. Eggert, S. Romberg, and R. Lienhart. Improving vlad: Hierarchical coding and a refined local coordinate system. In *ICIP*, 2014. 8

[11] J. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building Rome on a cloudless day. In *ECCV*. 2010. 1

[12] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *CVPR*, 2013. 2, 6

[13] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu. 3d visual phrases for landmark recognition. In *CVPR*, 2012. 2

[14] W. Hartmann, M. Havlena, and K. Schindler. Predicting matchability. In *CVPR*, 2014. 2

[15] J. Hays and A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 1

[16] J. Hays and A. Efros. Large-scale image geolocalization. In *Multimodal Location Estimation of Videos and Images*. 2015. 2

[17] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009. 2

[18] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 2011. 3

[19] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 2012. 3, 4, 8

[20] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*. 5

[21] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *ECCV*. 2010. 1, 2

[22] G. Koutaki and K. Uchimura. Scale-space processing using polynomial representations. In *CVPR*, 2014. 8

[23] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2

[24] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*. 2010. 1

[25] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*. 2012. 2

[26] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *CVPR*, 2012. 1

[27] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 1

[28] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 2004. 1

[29] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 6, 8

[30] D. Qin, C. Wengert, and L. Van Gool. Query adaptive similarity for large scale object retrieval. In *CVPR*, 2013. 2

[31] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 2

[32] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, 2007. 1, 2

[33] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. Efros. Data-driven visual similarity for cross-domain image matching. In *ACMTOG*, 2011. 2

[34] S. Singh, A. Gupta, and A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 5

[35] A. Taneja, L. Ballan, and M. Pollefeys. Never get lost again: Vision based navigation using streetview images. In *ACCV*, 2014. 2

[36] G. Tolias and H. Jégou. Visual query expansion with or without geometry: refining local descriptors by feature aggregation. *Pattern Recognition*, 2014. 2

[37] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *CVPR*, 2013. 6

[38] P. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshops*, 2009. 2

[39] K. van de Sande, C. Snoek, and A. Smeulders. Fisher and vlad with flair. In *CVPR*, 2014. 2

[40] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *CVPR*, 2009. 2, 3

[41] A. R. Zamir, S. Ardeshir, and M. Shah. Gps-tag refinement using random walks with an adaptive damping factor. In *CVPR*, 2014. 1

[42] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010. 1, 6

[43] A. R. Zamir and M. Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *TPAMI*, 2014. 1

[44] C. Zhang, J. Gao, O. Wang, P. Georgel, R. Yang, J. Davis, J. Frahm, and M. Pollefeys. Personal photograph enhancement using internet photo collections. *TVCG*, 2014. 1

[45] C.-Z. Zhu, H. Jégou, and S. Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV*, 2013. 2