Learning and calibrating per-location classifiers for visual place recognition

Petr Gronát[1,2,3] Guillaume Obozinski[4] Josef Sivic[1,3] Tomáš Pajdla[2]
[1]INRIA [2]Czech Technical University in Prague [4]Ecole des Ponts – ParisTech

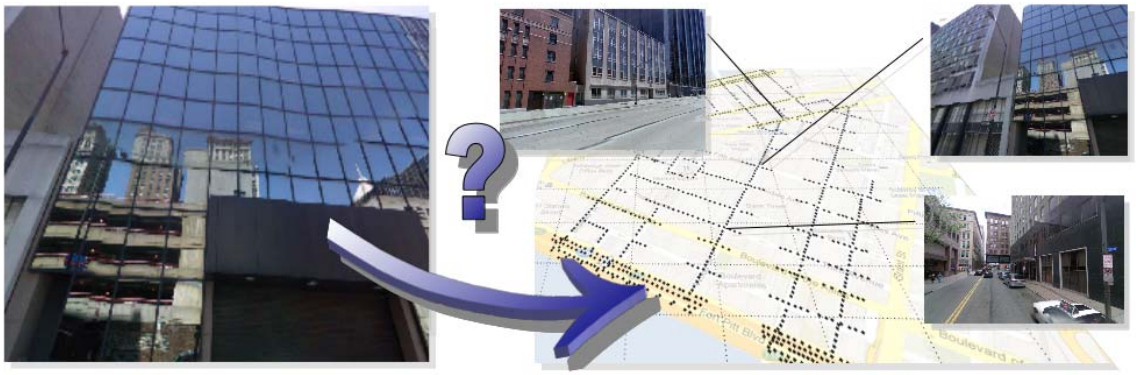


Figure 1: The goal of this work is to localize a query photograph (left) by finding other images of the same place in a large geotagged image database (right). We cast the problem as a classification task and learn a classifier for each location in the database. We develop a non-parametric procedure to calibrate the outputs of the large number of per-location classifiers without the need for additional positive training data.

Abstract

The aim of this work is to localize a query photograph by finding other images depicting the same place in a large geotagged image database. This is a challenging task due to changes in viewpoint, imaging conditions and the large size of the image database. The contribution of this work is two-fold. First, we cast the place recognition problem as a classification task and use the available geotags to train a classifier for each location in the database in a similar manner to per-exemplar SVMs in object recognition. Second, as only few positive training examples are available for each location, we propose a new approach to calibrate all the per-location SVM classifiers using only *the negative examples. The calibration we propose relies on a significance measure essentially equivalent to the p-values classically used in statistical hypothesis testing. Experiments are performed on a database of 25,000 geotagged street view images of Pittsburgh and demonstrate improved place recognition accuracy of the proposed approach over the previous work.*

1. Introduction

Visual place recognition [7, 13, 27] is a challenging task as the query and database images may depict the same 3D structure (e.g. a building) from a different camera viewpoint, under different illumination, or the building can be partially occluded. In addition, the geotagged database may be very large. For example, we estimate that Google street-view of France alone contains more than 60 million panoramic images.

Similar to other work in large scale place recognition [7, 13, 27] and image retrieval [20, 21, 28], we build on the bag-of-visual-words representation [6, 28] and describe each image by a set of quantized local invariant features, such as SURF [1] or SIFT [17]. Each image is then represented by a weighted histogram of visual words, called the "tf-idf vector" due to the commonly used tf-idf weighting scheme [28]. The vectors are usually normalized to have unit L_2 norm and the similarity between the query and a database vector is then measured by their dot product. This representation has some desirable properties such as robustness to background clutter and partial occlusion. Efficient retrieval is then achieved using inverted file indexing.

Recent work has looked at different ways to improve the retrieval accuracy and speed of the bag-of-visual-words model for image and object retrieval. Examples include: (i) learning better visual vocabularies from training examples with matched/non-matched descriptors [19, 23]; (ii) developing quantization methods less prone to quantization er-

[2]Center for Machine Perception, Faculty of Electrical Engineering
[3]WILLOW project, Laboratoire d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.
[4]Universit Paris-Est, LIGM (UMR CNRS 8049), Center for Visual Computing, Ecole des Ponts - ParisTech, 77455 Marne-la-Valle, France

rors [11, 22] or (iii) combining results from multiple query images depicting the same scene [4, 5].

While in image retrieval databases are typically unstructured collections of images, place recognition databases are usually structured: images have geotags, are localized on a map and depict a consistent 3D world. Knowing the structure of the database can lead to significant improvements in both speed and accuracy of place recognition. Examples include: (i) building an explicit 3D reconstruction of the scene [10, 15, 16]; (ii) constructing an image graph [24, 30], where images are nodes and edges connect close-by images on the map [29], or (iii) using the geotagged data as a form of supervision to select local features that characterize a certain location [13, 27] or re-rank retrieved images [32].

In this work, we also take advantage of geotags as an available form of supervision and investigate whether the place recognition problem can be cast as a classification task. While visual classifiers were investigated for landmark recognition [14], where many photographs are available for each of the landmarks, in this work we wish to train a classifier *for each location on the map* in a similar manner to per-exemplar classification in object recognition [18]. This is beneficial as each classifier can learn which features are discriminative for a particular place. The classifiers are learnt offline. At query time, the query photograph is localized by transferring the GPS tag of the best scoring location classifier.

While learning classifiers for each place may be appealing, calibrating outputs of the individual classifiers is a critical issue. In object recognition [18], it is addressed in a separate calibration stage on a held-out set of training data. This is not possible in the place recognition set-up as only a small number, typically one to five, of positive training images are available for each location (e.g. street-view images viewing the same building facade). To address this issue, we propose a calibration procedure inspired by the use of p-values in statistics and based on ranking the score of a query image amongst scores of other images in the database.

The rest of the paper is organized as follows: Section 2 describes how per-location classifiers are learnt. Section 3 details the classifier calibration procedure. Implementation details and experimental results are given in Section 4.

2. Per-location classifiers for place recognition

We are given tf-idf vectors d_j, one for each database image j. The goal is to learn a score f_j for each database image j, so that, at test time, given the descriptor q of the query image, we can either retrieve the correct target image as the image j^* with the highest score

$$j^* = \arg\max_j f_j(q) \tag{1}$$

or use these scores to rank candidate images and use geometric verification to try and identify the correct location in an n-best list. Instead of approaching the problem directly as a large multiclass classification problem, we tackle the problem by learning a per-exemplar linear SVM classifier [18] for each database image j. Similar to [13], we use the available geotags to construct the negative set $\mathcal{N}_j$ for each image j. The negative set is constructed so as to concentrate difficult negative examples, i.e. from images that are far away from the location of image j and at the same time similar to the target image as measured by the dot product between their feature vectors. The details of the construction procedure will be given in section 4. The positive set $\mathcal{P}_j$ is represented by the only positive example, which is d_j itself.

Each SVM classifier produces a score s_j which is a priori not comparable with the score of the other classifiers. A calibration of these scores will therefore be key to convert them to comparable scores f_j. This calibration problem is more difficult than usual given that we only have a single positive example and will be addressed in section 3.

Learning per-location SVM classifiers. Each linear SVM classifier learns a score s_j of the form

$$s_j(q) = w_j^T q + b_j \tag{2}$$

where w_j is a weight vector re-weighting contributions of individual visual words and b_j is the bias specific for image j. Given the training sets $\mathcal{P}_j$ and $\mathcal{N}_j$, the aim is to find a vector w_j and bias b_j such that the score difference between d_j and the closest neighbor from its negative set $\mathcal{N}_j$ is *maximized*. Learning the weight vector w_j and bias b_j is formulated as a minimization of the convex objective

$$\Omega(w_j, b_j) = ||w_j||^2 + C_1 \sum_{x \in \mathcal{P}_j} h(w_j^T x + b_j) + C_2 \sum_{x \in \mathcal{N}_j} h(-w_j^T x - b_j), \tag{3}$$

where the first term is the regularizer, the second term is the loss on the positive training data weighted by scalar parameter C_1, and the third term is the loss on the negative training data weighted by scalar parameter C_2. This is a standard SVM formulation (3), also used in exemplar-SVM [18]. In our case h is the squared hinge loss, which we found to work better in our setting than the standard hinge-loss. w_j and b_j are learned separately for each database image j in turn. In our case (details in section 4), we use about 1-5 positive examples, and 200 negative examples. As the dimensionality of w is $100,000$ all training data points are typically support vectors.

Expanding the positive set. A typical geotagged database may contain several images depicting a particular location. For example, neighboring street-view panoramas depict the same store front from different viewpoints. However, a specific place is often imaged only in a small number (2-5) of neighboring panoramas. If such images are identified, they may provide a few additional positive examples for the particular place and improve the quality of that per-location classifier. Moreover, treating them erroneously as negatives is likely to bias the learnt classifier. We automatically identify such images as geo-graphically close-by images to the location j. These images can be further verified using geometric verification [21] and included in the positive training data for location j. Details are given in section 4.

3. Non-parametric calibration of the SVM-scores from negative examples only

Since the classification scores s_j are learned independently for each location j, they cannot be directly used as the scores f_j from eq. (1). As illustrated in figure 2, for a given query q, a classifier from an incorrect location (b) can have a higher score (2) than the classifier from the target location (a). Indeed, the SVM score is a signed distance from the discriminating hyperplane and is a priori not comparable between different classifiers. This issue is addressed by calibrating scores of the learnt classifiers. The goal of the calibration is to convert the output of each classifier into a probability (or in general a "universal" score), which can be meaningfully compared across classifiers.

Several calibration approaches have been proposed in the literature (see [9] and references therein for a review). The most known consists of fitting a logistic regression to the output of the SVM [25]. This approach, however, has a major drawback as it imposes a parametric form (the logistic a.k.a. sigmoid function) of the likelihood ratio of the two classes, which typically leads to biased estimates of the calibrated scores. Another important calibration method is the isotonic regression [31], which allows for a non-parametric estimate of the output probability. Unfortunately, the fact that we have only a single positive example (or only very few of them, and which are all used for training) essentially prevents us from using any of these methods. However, given the availability of negative data, it is easy to estimate the significance of the score of a test example compared to the typical score of (plentifully available) negative examples. Intuitively, we will use a large dataset of negative examples to calibrate the individual classifiers so that they *reject the same number of negative examples* at each level of the calibrated score. We will expand this idea in detail and use concepts from hypothesis testing to propose a calibration method.

Calibration *via* significance levels. In the following, we view the problem of deciding whether a query image matches a given location based on the corresponding SVM score as a hypothesis testing problem. In particular, we appeal to ideas from the traditional frequentist hypothesis testing framework also known as Neyman-Pearson (NP) framework (see e.g. [2], chap. 8).

We define the null hypothesis as $H_0 = \{\text{the image is a random image}\}$ and the alternative as $H_1 = \{\text{the image matches the particular location}\}$. The NP framework focuses on the case where the distribution of the data under H_0 is well known, whereas the distribution under H_1 is not accessible or too complicated to model, which matches perfectly our setting.

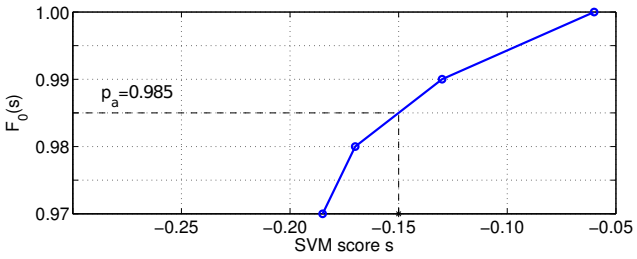
In the NP framework, the *significance level* of a score is measured by the p-value or equivalently by the value of the cumulative density function (cdf) of the distribution of the negatives at a given score value. The cdf is the function F_0 defined by $F_0(s) = \mathbb{P}(S_0 \leq s)$, where S_0 is the random variable corresponding to the scores of negative data (see figure 3 for an illustration of the relation between the cdf and the density of the function). The cdf (or the corresponding p-value[1]) is naturally estimated by the empirical cumulative density function $\hat{F}_0$, which is computed as:

$$\hat{F}_0(s) = \frac{1}{N_c} \sum_{n=1}^{N_c} 1_{\{s_n \leq s\}},$$

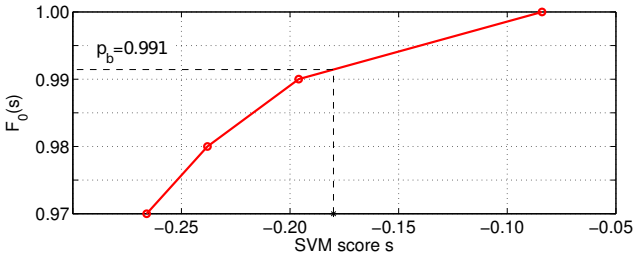
where $(s_n)_{1 \leq n \leq N_c}$ are the SVM scores associated with N_c negative examples used for calibration. $\hat{F}_0(s)$ is the fraction of the negative examples used for calibration (ideally held out negative examples) that have a score below a given value s. Computing $\hat{F}_0$ exactly would require to store all the SVM scores for all the calibration data for all classifiers, so in practice, we only keep a fraction of the larger scores. We also interpolate the empirical cdf between consecutive datapoints so that instead of being a staircase function it is a continuous piecewise linear function such as illustrated on figure 2. Given a query, we first compute its SVM score s_q and then compute the calibrated probability $f(q) = \hat{F}_0(s_q)$. We obtain a similar calibrated probability $f_j(q)$ for each of the SVMs associated with each of the target locations, which can now be ranked.

Summary of the calibration procedure. For each place, keep N_c scores from negative examples $(s_n)_{1 \leq n \leq N_c}$ used for calibration together with the associated cumulative

[1] The notion most commonly used in statistics is in fact the p-value. The p-value associated to a score is the quantity $\alpha(s)$ defined by $\alpha(s) = 1 - F_0(s)$; so the more significant the score is, the closer to 1 the cdf value is, and the closer to 0 the p-value is. To keep the presentation simple, we avoid the formulation in terms of p-values and we only talk of the probabilistic calibrated values obtained from the cdf F_0.

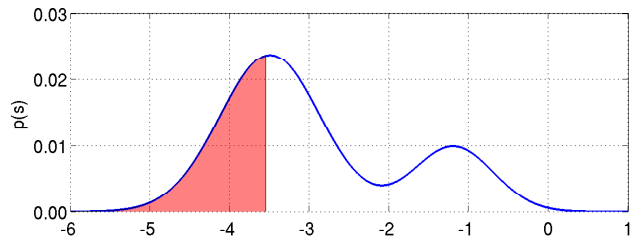


(a)

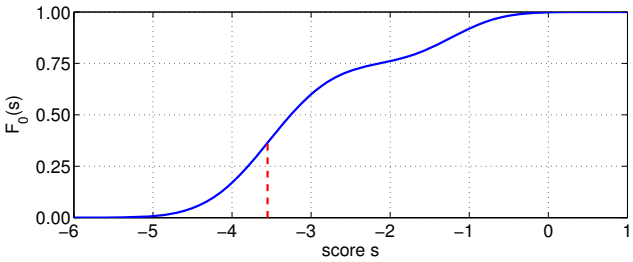


(b)

Figure 2: An illustration of the proposed normalization of SVM scores for two different database images. In each plot, the x-axis shows the raw SVM score. The y-axis shows the calibrated output. For the given query, the raw SVM score of image (b) is lower than for image (a), but the calibrated score of image (b) is higher than for image (a).



(a)



(b)

Figure 3: A figure showing the relation between (a) the probability density of the random variable S_0 modeling the scores of the negative examples and (b) the corresponding cumulative density function $F_0(s) = \mathbb{P}(S_0 \leq s)$.

probability values $\hat{F}_0(s_n)$. Given the score of the query s_q:
1. Find n such that $s_n \leq s_q < s_{n+1}$
2. Compute the interpolated empirical cdf value

$$\hat{F}_0(s_q) \approx \hat{F}_0(s_n) + \frac{s_q - s_n}{s_{n+1} - s_n}(\hat{F}_0(s_{n+1}) - \hat{F}_0(s_n)).$$

Discussion. It should be noted that basing the calibration only on the negative data has the advantage that we privilege precision over recall, which is justified given the imbalance of the available training data (much more negatives than positives). Indeed, since we are learning with a single positive example, intuitively, we cannot guarantee that the learned partition of the space will generalize well to other positives, whose scores in the test set can potentially drop significantly (this is indeed what we observe in practice). By contrast, since we are learning from a comparatively large number of negative examples, we can trust the fact that new negative examples will stay in the half-space containing the negative training set, so that their scores are very unlikely to be large. Our method is therefore based on the fact that we can measure reliably how surprising a high score would be if it was the score of a negative example. This exactly means that we can control false positives (type I error) reasonably well but not false negatives (type II error or equivalently the power of our test/classifier), exactly as in the Neyman-Pearson framework. An additional reason for not relying on positive examples for the calibration in our case is that (even if we had sufficiently many of them) the positive examples that we collect using location and geometric verification from the geotagged database typically have illumination conditions that are extremely similar to each other and not representative of the distribution of test positives which can have very different illuminations. This is because of the controlled nature of the capturing process of geotagged street-level imagery (e.g. Google street-view) used for experiments in this work. Close-by images are typically captured at a similar time (e.g. on the same day) and under similar imaging conditions.

Scheirer et al. [26] propose a method, which is related to ours, and calibrate SVM scores by computing the corresponding cdf value of a Weibull distribution fitted to the top negative scores. The main difficulty is that the Weibull model should be fitted only to the tail of the distribution of the negatives, which is in general difficult to identify. As a heuristic, Scheirer et al. propose to fit the Weibul model to false positives (i.e. the negative samples classified incorrectly as positives). But in our case, most of the exemplar SVMs that we are training have 0 false positives in a held out set, which precludes the application of their method. To avoid that issue our approach forgoes any parametric form of the distribution and instead relies directly on a standard non-parametric estimate of the cumulative density function.

Finally, we should remark that we are not doing here cal-

ibration in the same sense of the word as the calibration based on logistic regression (or isotonic regression), since logistic regression estimates a probability of making a correct prediction by assigning a new data to class 1, while we are estimating how unlikely it would be for a negative example to have such a high score. The calibration with either methods yields "universal" scores in the sense that they are comparable from one SVM to another, but the calibrated values obtained from logistic regression are not comparable to the values obtained from our approach.

4. Experiments

In this section we first give implementation details, then describe the experimental datasets and finally compare performance of the proposed approach with several baseline methods.

Implementation details. All images are described using the bag-of-visual-words representation [28]. First, SURF [1] descriptors are extracted. Second, a vocabulary of 100k visual words is learnt by approximate k-means clustering [21] from a subset of features from 2,000 randomly selected images. Third, a tf-idf vector is computed for each image by assigning each descriptor to the nearest cluster center. Finally, all tf-idf vectors are normalized to have unit L_2 norm.

To learn the classifier for database image j, the positive and negative training data is constructed as follows. The *negative training set* N_j is obtained by: (i) finding the set of images with geographical distance greater than $200\mathrm{m}$; (ii) sorting the images by decreasing value of similarity to image j measured by the dot product between their respective tf-idf vectors; (iii) taking the top $N = 200$ ranked images as the negative set. In other words, the negative training data consists of the hard negative images, i.e. those that are very similar to image j but are far away from its geographical position, hence, cannot have the same visual content.

The *positive training set* P_j initially consist of the image j itself and can be expanded by: (i) finding the adjacent images (e.g. images located within $< 20m$ of image j), (ii) identifying adjacent images with the same visual content using geometric verification, and (iii) adding these verified images to the positive set P_j.

For SVM training we use `libsvm` [8]. We set the value of the regularization parameters to $C_1 = 1 \cdot n_P$ for positive data and $C_2 = 10^{-3} \cdot n_N$ for negative data where n_P and n_N denote the number of examples in the positive and the negative set, respectively. These parameters were found by cross-validation and work well on various datasets.

The calibration with significance levels is done for each classifier in turn as follows: (i) given image j and learnt SVM we construct a set of images consisting of the whole database without the positive set P_j; (ii) for this image set, SVM scores are computed; (iii) empirical cdf $\hat{F}_0$ is estimated from sorted SVM scores.

To use a reasonable amount of memory, for each classifier, we store only the first 1000 largest negative scores (the number of negative scores stored could be reduced further using interpolation).

Image dataset. We performed experiments on a database of Google Streetview images from the Internet. We downloaded panoramas from Pittsburgh (U.S.) covering roughly an area of 1.3×1.2 km^2. Similar to [3], we generate for each panorama 12 overlapping perspective views corresponding to two different elevation angles to capture both the street-level scene and the building façades, resulting in a total of 24 perspective views each with 90° FOV and resolution of 960×720 pixels. This dataset contains 25,000 perspective images.

As a query set with known ground truth GPS positions, we use the 8999 panoramas from the Google Streetview research dataset, which cover approximately the same area, but were captured at a different time, and typically depict the same places from different viewpoints and under different illumination conditions. For each test panorama, we generate perspective images as described above. Finally, we randomly select out of all generated perspective views a subset of 4k images, which is used as a test set to evaluate the performance of the proposed approach.

Results. We compare the performance of the proposed approach (SVM p-val) with the following baseline methods: (a) Training per-location classifiers without any calibration step (SVM). (b) Calibrating per-location classifiers using the standard logistic regression[2] as in exemplar SVM [18] (SVM logistic). (c) The standard bag-of-visual words retrieval (BOW) [21]. (d) Our implementation of the confuser suppression approach (Conf. supp.) of [13] that, in each database image, detects and removes features that frequently appear at other far-away locations (using parameters $t = 3.5$ and $w = 70$).

For all methods, we implemented a two-stage place recognition approach. Given a query image, the aim of the first stage is to efficiently find a small subset (20) of candidates that are likely to depict the same place as the query image. In the second stage, we search for restricted homographies between candidates and the query image using RANSAC [21]. The candidates are finally re-ranked by decreasing number of inliers.

Since the ground truth GPS position for each query image is available, we measure the overall recognition perfor-

[2]The calibration of SVM scores with logistic regression is based on a subset of 30 hard negatives from N_j and 1-15 available positive examples from P_j.

Method	% correct *init. retrieval*	% correct *with geom. verif.*
SVM	00.0	12.7
SVM logistic	03.6	10.3
BOW	32.0	53.1
Conf. supp. [13]	36.5	58.1
SVM p-val	**41.9**	**60.8**

Table 1: The percentage of correctly localized test queries for which the top-ranked database image is within 20 meters from the ground truth query position. The proposed method (SVM p-val) outperforms the baseline methods. Results are shown for the initial retrieval (left column) and after re-ranking the top 20 retrieved images using geometric verification. Notice that SVM output without calibration gives 0% of correctly localized queries.

mance by the percentage of query test images for which the top-ranked database image was located within a distance of 20 meters from the ground truth query location. Results are summarized in table 1 and clearly demonstrate the benefits of careful calibration of the per-location classifiers. In addition, the proposed per-location classifier method outperforms the baseline bag-of-visual-word approach [21] including confuser suppression [13].

Examples of correctly and incorrectly localized queries are shown in figure 4. Figure 5 illustrates the weights learnt for one database image applied to three different query images.

Scalability. The linear SVM classifiers trained for each database image are currently non-sparse, which increases the computational and memory requirements at query time compared to the original bag-of-visual-words representation. For a database of 25,000 images, applying all classifiers on a query image takes currently on average 1.72s. The method could be further sped-up by, for example: (i) reducing the dimensionality of the input vectors [12], or (ii) enforcing additional sparsity constraints on learnt weight vectors w.

5. Conclusions

We have shown that place recognition can be cast as a classification problem and have used geotags as a readily-available supervision to train an ensemble of classifiers, one for each location in the database. As only few positive examples are available for each location, we have proposed a non-parametric procedure to calibrate the output of each classifier without the need for additional positive training data. The results show improved place recognition performance over baseline methods and demonstrate that careful calibration is critical to achieve competitive place recognition performance. The developed calibration method is not specific to place recognition and can be useful for other per-exemplar classification tasks, where only a small number of positive examples are available [18].

Acknowledgements This work was supported by the MSR-INRIA laboratory, the EIT-ICT labs, PRoViDE EU FP7-SPACE-312377 project and SGS13/140/OHK3/2T/13. Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: 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.

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. 1, 5

[2] G. Casella and R. Berger. Statistical inference. 2001. 3

[3] D. Chen, G. Baatz, Köser, S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, 2011. 5

[4] O. Chum, A. Mikulik, M. Perdroch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, 2011. 2

[5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007. 2

[6] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004. 1

[7] M. Cummins and P. Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. 1

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Machine Learning Research*, 9:1871–1874, 2008. 5

[9] M. Gebel and C. Weihs. Calibrating classifier scores into probabilities. *Advances in Data Analysis*, pages 141–148, 2007. 3

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

[11] H. Jégou, M. Douze, and C. Schmid. Product Quantization for Nearest Neighbor Search. *IEEE PAMI*, 33(1):117–128, 2011. 2

[12] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE PAMI*, 34:1704–1716, 2012. 6

[13] J. Knopp, J. Sivic, and T. Pajdla. Avoidng confusing features in place recognition. In *ECCV*, 2010. 1, 2, 5, 6, 7

[14] Y. Li, D. Crandall, and D. Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, 2009. 2

[15] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010. 2

Figure 4: **Examples of query images (gray) correctly (green) and incorrectly (red) localized by different methods.** (a) query image. (b) the top-ranked image retrieved by per-location classifiers (proposed method). (c) the top-ranked image retrieved by the baseline confuser suppression method [13]. (d) the top-ranked image retrieved by the baseline bag-of-visual-words method. Bottom two rows: the proposed method is sometimes confused by high-scoring similar repeated texture patterns on facades.

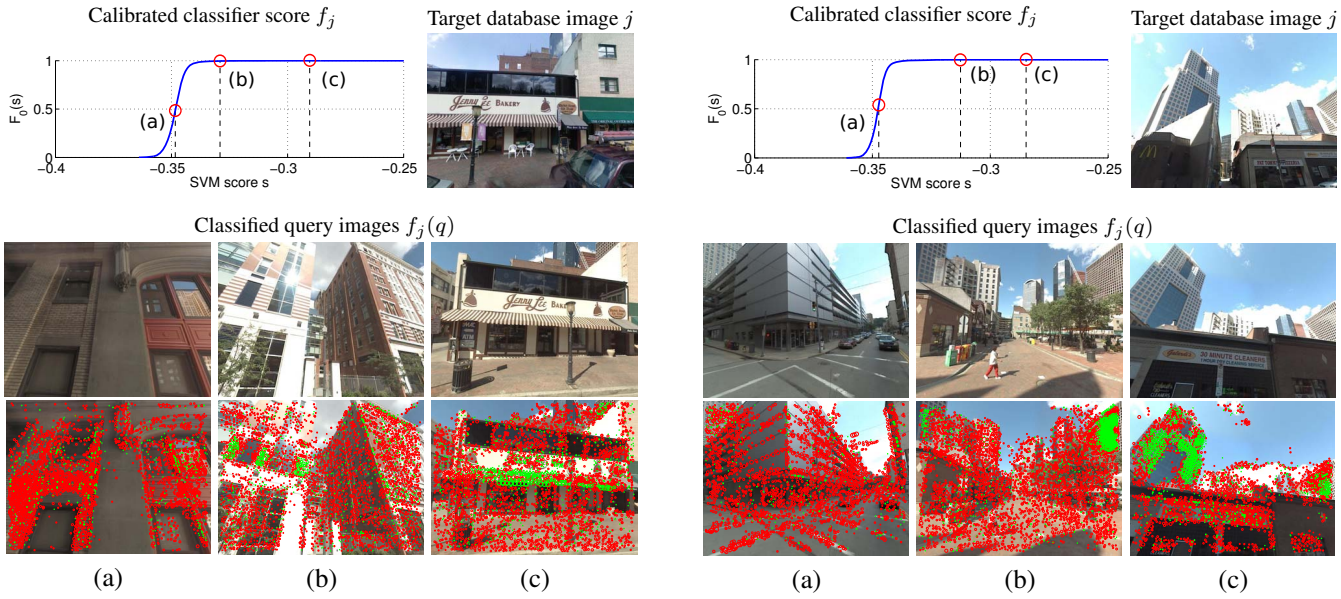


Figure 5: **A visualization of learnt feature wights for two database images. In each panel:** *first row:* (Right) Target database image j. (Left) Cumulative density function (or calibrated score) learnt for the SVM scores of the corresponding classifier f_j; three query images displayed on the *second row* are represented by their SVM scores and cdf values $F_0(s)$, denoted (a)-(c) on the graph. *Third row:* A visualization of the contribution of each feature to the SVM score for the corresponding query image. Red circles represent features with negative weights while green circles correspond to features with positive weights. The area of each circle is proportional to the contribution of the corresponding feature to the SVM score. Notice that the correctly localized queries (c) contain more green colored features than queries from other places (b) and (a). *Left panel:* Query (b) gets a high score because the building has orange and white stripes similar to the the sun-blinds of the bakery, which are features that also have large positive weights in the query image (c) of the correct place. *Right panel:* Query (b) is in fact also an image of the same location with a portion of the left skyscraper in the target image detected in the upper left corner and the side of the rightmost building in the target image detected in the top right corner. Both are clearly detected by the method as indicated by a large quantity of green circles in the corresponding regions.

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

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[18] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 2, 5, 6

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

[20] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 1

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 1, 3, 5, 6

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 2

[23] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010. 1

[24] J. Philbin, J. Sivic, and A. Zisserman. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *IJCV*, 2010. 2

[25] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999. 3

[26] W. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012. 4

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

[28] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1, 5

[29] A. Torii, J. Sivic, and T. Pajdla. Visual localization by linear combination of image descriptors. In *IEEE Workshop on Mobile Vision*, 2011. 2

[30] P. Turcot and D. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problem. In *WS-LAVD, ICCV*, 2009. 2

[31] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD*, 2002. 3

[32] A. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010. 2