Visual Place Recognition with Repetitive Structures

Akihiko Torii
Tokyo Tech*
torii@ctrl.titech.ac.jp

Josef Sivic
INRIA†
Josef.Sivic@ens.fr

Tomas Pajdla
CTU in Prague‡
pajdla@cmp.felk.cvut.cz

Masatoshi Okutomi
Tokyo Tech*
mxo@ctrl.titech.ac.jp

Abstract

Repeated structures such as building facades, fences or road markings often represent a significant challenge for place recognition. Repeated structures are notoriously hard for establishing correspondences using multi-view geometry. Even more importantly, they violate the feature independence assumed in the bag-of-visual-words representation which often leads to over-counting evidence and significant degradation of retrieval performance. In this work we show that repeated structures are not a nuisance but, when appropriately represented, they form an important distinguishing feature for many places. We describe a representation of repeated structures suitable for scalable retrieval. It is based on robust detection of repeated image structures and a simple modification of weights in the bag-of-visual-word model. Place recognition results are shown on datasets of street-level imagery from Pittsburgh and San Francisco demonstrating significant gains in recognition performance compared to the standard bag-of-visual-words baseline and more recently proposed burstiness weighting.

1. Introduction

Given a query image of a particular street or a building, we seek to find one or more images in the geotagged database *depicting the same place*. The ability to visually recognize a place depicted in an image has a range of potential applications including automatic registration of images taken by a mobile phone for augmented reality applications [1] and accurate visual localization for robotics [7]. Scalable place recognition methods [3, 7, 18, 31, 37] often build on the efficient bag-of-visual-words representation developed for object and image retrieval [6, 13, 15, 24, 26, 40]. In an offline pre-processing stage, local invariant descriptors are
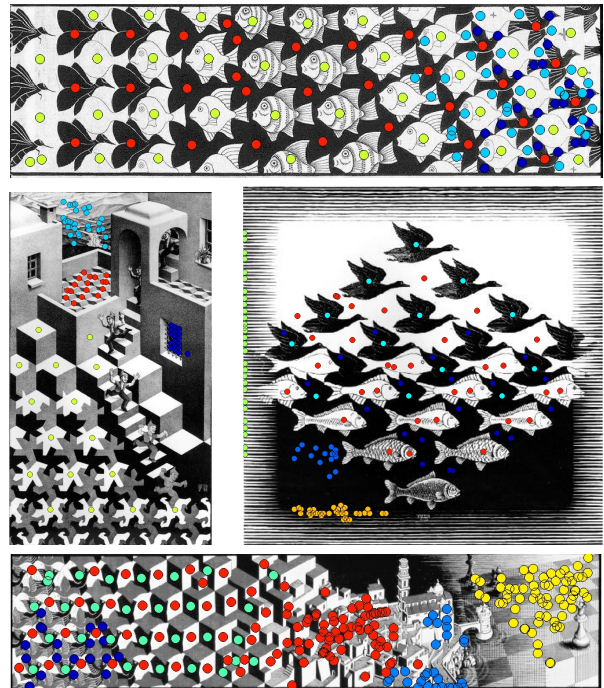

Figure 1. We detect groups of repeated local features (overlaid in colors). The detection is robust against local deformation of the repeated element and makes only weak assumptions on the spatial structure of the repetition. We develop a representation of repeated structures for efficient place recognition based on a simple modification of weights in the bag-of-visual-word model.

extracted from each image in the database and quantized into a pre-computed vocabulary of visual words. Each image is represented by a sparse (weighted) frequency vector of visual words, which can be stored in an efficient inverted file indexing structure. At query time, after the visual words are extracted from the query image, the retrieval proceeds in two steps. First a short-list of ranked candidate images is obtained from the database using the bag-of-visual-words representation. Then, in the second verification stage, candidates are re-ranked based on the spatial layout of visual words.

A number of extensions of this basic architecture have

*Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology

†WILLOW project, Laboratoire d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

‡Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Enginnering, Czech Technical University in Prague

been proposed. Examples include: (i) learning better visual vocabularies [21, 28]; (ii) developing quantization methods less prone to quantization errors [14, 27, 44]; (iii) combining returns from multiple query images depicting the same scene [4, 6]; (iv) exploiting the 3D or graph structure of the database [11, 20, 29, 42, 43, 47]; or (v) indexing on spatial relations between visual words [5, 12, 48].

In this work we develop a scalable representation for large-scale matching of repeated structures. While repeated structures often occur in man-made environments – examples include building facades, fences, or road markings – they are usually treated as nuisance and downweighted at the indexing stage [13, 18, 36, 39]. In contrast, we develop a simple but efficient representation of repeated structures and demonstrate its benefits for place recognition in urban environments. In detail, we first robustly detect repeated structures in images by finding spatially localized groups of visual words with similar appearance. Next, we modify the weights of the detected repeated visual words in the bag-of-visual-word model, where multiple occurrences of repeated elements in the same image provide a *natural soft-assignment* of features to visual words. In addition the contribution of repetitive structures is controlled to prevent dominating the matching score.

The rest of the paper is organized as follows. After describing related work on finding and matching repeated structures (Section 1), we review in detail (Section 2) the common tf-idf visual word weighting scheme and its extensions to soft-assignment [27] and repeated structure suppression [13]. In Section 3 we describe our method for detecting repeated visual words in images. In Section 4, we describe the proposed model for scalable matching of repeated structures, and demonstrate its benefits for place recognition in section 5.

Related work.   Detecting repeated patterns in images is a well-studied problem. Repetitions are often detected based on an assumption of a single pattern repeated on a 2D (deformed) lattice [10, 19, 25]. Special attention has been paid to detecting planar patterns [35, 38] and in particular building facades [3, 9, 45], for which highly specialized grammar models, learnt from labelled data, were developed [23, 41].

Detecting planar repeated patterns can be useful for single view facade rectification [3] or even single-view 3D reconstruction [46]. However, the local ambiguity of repeated patterns often presents a significant challenge for geometric image matching [33, 38] and image retrieval [13].

Schindler *et al.* [38] detect repeated patterns on building facades and then use the rectified repetition elements together with the spatial layout of the repetition grid to estimate the camera pose of a query image, given a database of building facades. Results are reported on a dataset of 5 query images and 9 building facades. In a similar spirit,

Doubek *et al.* [8] detect the repeated patterns in each image and represent the pattern using a single shift-invariant descriptor of the repeated element together with a simple descriptor of the 2D spatial layout. Their matching method is not scalable as they have to exhaustively compare repeated patterns in all images. In scalable image retrieval, Jegou *et al* [13] observe that repeated structures violate the feature independence assumption in the bag-of-visual-word model and test several schemes for down-weighting the influence of repeated patterns.

2. Review of visual word weighting strategies

In this section we first review the basic tf-idf weighting scheme proposed in text retrieval [32] and also commonly used for the bag-of-visual-words retrieval and place recognition [3, 6, 12, 13, 18, 24, 26, 40]. Then, we discuss the soft-assignment weighting [27] to reduce quantization errors and the 'burstiness' model recently proposed by Jegou *et al.* [13], which explicitly downweights repeated visual words in an image.

Term frequency–inverse document frequency weighting. The standard 'term frequency–inverse document frequency' (*tf–idf*) weighting [32], is computed as follows. Suppose there is a vocabulary of V visual words, then each image is represented by a vector

$$\mathbf{v}_d = (t_1, ..., t_i, ..., t_V)^\top \qquad (1)$$

of weighted visual word frequencies with components

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{N_i}, \qquad (2)$$

where n_{id} is the number of occurrences of visual word i in image d, n_d is the total number of visual words in the image d, N_i is the number of images containing term i, and N is the number of images in the whole database. The weighting is a product of two terms: the *visual word frequency*, n_{id}/n_d, and the *inverse document (image) frequency*, $\log N/N_i$. The word frequency weights words occurring more often in a particular image higher (compared to visual word present/absent), whilst the inverse document frequency downweights visual words that appear often in the database, and therefore do not help to discriminate between different images. At the retrieval stage, images are ranked by the normalized scalar product (cosine of angle)

$$f_d = \frac{\mathbf{v}_q^\top \mathbf{v}_d}{\|\mathbf{v}_q\|_2 \, \|\mathbf{v}_d\|_2} \qquad (3)$$

between the query vector $\mathbf{v}_q$ and all image vectors $\mathbf{v}_d$ in the database, where $\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^\top \mathbf{v}}$ is the L_2 norm of $\mathbf{v}$. When both the query and database vectors are pre-normalized to unit L_2 norm, equation (3) simplifies to the standard scalar product, which can be implemented efficiently using inverted file indexing schemes.

Soft-assignment weighting. Visual words generated through descriptor clustering often suffer from quantization errors, where local feature descriptors that should be matched but lie close to the Voronoi boundary are incorrectly assigned to different visual words. To overcome this issue, Philbin *et al.* [27] soft-assign each descriptor to several (typically 3) closest cluster centers with weights set according to $\exp -\frac{d^2}{2\sigma^2}$, where d is the Euclidean distance of the descriptor from the cluster center and σ is a parameter of the method.

Burstiness weighting. Jegou *et al.* [13] study the effect of visual "burstiness", i.e. that a visual-word is much more likely to appear in an image, if it has appeared in the image already. Burstiness has been also studied for words in text [17]. Jegou *et al.* observe by counting visual word occurrences in a large corpus of 1M images that visual words occurring multiple times in an image (e.g. on repeated structures) violate the assumption that visual word occurrences in an image are independent. Further they observe that the bursted visual words can negatively affect retrieval results. The intuition is that the contribution of visual words with a high number of occurrences towards the scalar product in equation (3) is too high. In the voting interpretation of the bag-of-visual-words model [12], bursted visual words vote multiple times for the same image. To see this, consider an example where a particular visual word occurs twice in the query and five times in a database image. Ignoring the normalization of the visual word vectors for simplicity, multiplying the number of occurrences as in (3) would result in 10 votes, whereas in practice only up to two matches (correspondences) can exist.

To address this problem Jegou *et al.* propose to downweight the contribution of visual words occurring multiple times in an image, which is referred to as intra-image burrstiness. They experiment with different weighting strategies and empirically observe that down-weighting repeated visual words by multiplying the term frequency in equation (3) by factor $\frac{1}{\sqrt{n_{id}}}$, where n_{id} is the number of occurrences, performs best. Similar strategies to discount repeated structures when matching images were also used in [36, 39].

Note that Jegou *et al.* also consider a more precise description of local invariant regions quantized into visual words using an additional binary signature [12] more precisely localizing the descriptor in the visual word Voronoi cell. For simplicity, we do not consider this representation here.

In contrast to downweighting repeated structures based on globally counting feature repetitions across the entire image, we (i) explicitly detect localized image areas with repetitive structures, and (ii) use the detected local repetitions to adaptively adjust the visual word weights in the

soft-assigned bag-of-visual words model. The two steps are described next.

3. Detection of repetitive structures

The goal is to segment local invariant features detected in an image into localized groups of repetitive patterns and a layer of non-repeated features. Examples include detecting repeated patterns of windows on different building facades, as well as fences, road markings or trees in an image (see figure 2). We will operate directly on the extracted local features (rather than using specially designed features [9]) as the detected groups will be used to adjust feature weights in the bag-of-visual-words model for efficient indexing. The feature segmentation problem is posed as finding connected components in a graph.

In detail, we build an (undirected) feature graph $G = (V, E)$ with N vertices $V = \{(\mathbf{x}_i, s_i, \mathbf{d}_i)\}_{i=1}^N$ consisting of local invariant features at locations $\mathbf{x}_i$, scales s_i and with corresponding SIFT descriptors $\mathbf{d}_i$. Each SIFT descriptor is further assigned to the top $K = 50$ nearest visual words from a pre-computed visual vocabulary (see section 5 for details). Two vertices (features) are connected by an edge if they have close-by image position as well as similar scale and appearance. More formally, a pair of vertices V_i and V_j is connected by an edge if the following three conditions are satisfied:

1. The spatial L_2 distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ between features satisfies $\|\mathbf{x}_i - \mathbf{x}_j\| < c(s_i + s_j)$ where c is a constant (we set $c = 10$ throughout experiments);
2. The ratio σ of scales of the two features is in $0.5 < \sigma < 1.5$;
3. The features share at least one common visual word in their individual top K visual word assignments. Note that this condition avoids directly thresholding the distance between the SIFT descriptors of the two features, which we found unreliable.

Having built the graph, we group the vertices (image features) into disjoint groups by finding connected components of the graph [30]. These connected components group together features that are spatially close, and are also similar in appearance as well as in scale. In the following, we will call the detected feature groups "repttiles" for "tiles (regions) of repetitive features".

Figures 1 and 2 show a variety of examples of detected patterns of repeated features. Only connected components with more than 20 image features are shown as colored dots. Note that the proposed method makes only weak assumptions on the type and spatial structure of repetitions, not requiring or attempting to detect, for example, feature symmetry or an underlying spatial lattice.
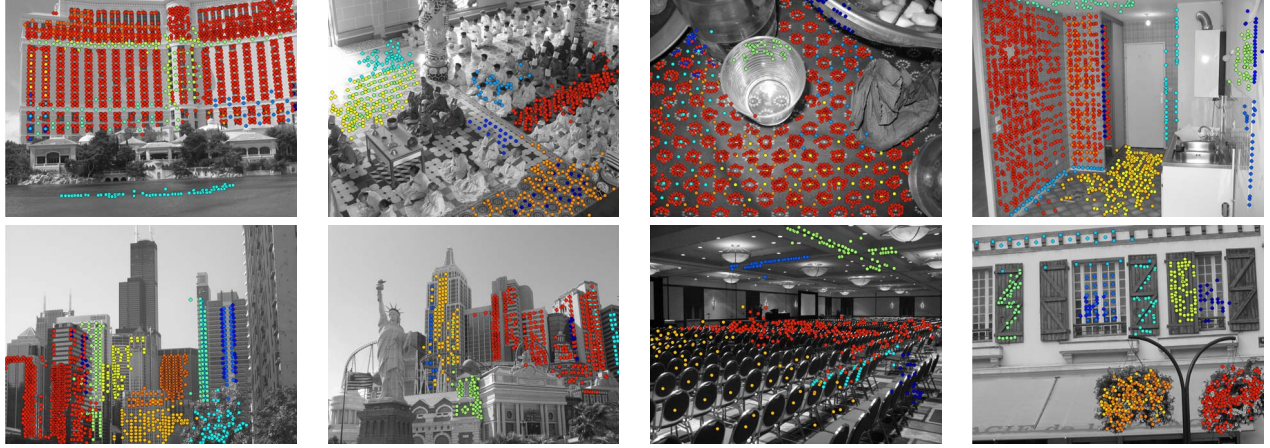
Figure 2. Examples of detected repetitive patterns of local invariant features ("repttiles") in images from the INRIA Holidays dataset [13]. The different repetitive patterns detected in each image are shown in different colors. The color indicates the number of features in each group (red indicates large and blue indicates small groups). Note the variety of detected repetitive structures such as different building facades, trees, indoor objects, window tiles or floor patterns.

4. Representing repetitive structures for scalable retrieval

In this section we describe our image representation for efficient indexing taking into account the repetitive patterns. The proposed representation is built on two ideas. First, we aim at representing *the presence* of a repetition, rather than measuring the actual number of matching repeated elements. Second, we note that different occurrences of the same visual element (such as a facade window) are often quantized to different visual words naturally representing the noise in the description and quantization process as well as other non-modeled effects such as complex illumination (shadows) or perspective deformation. We take advantage of this fact and design a descriptor quantization procedure that *adaptively soft-assigns* local features with more repetitions in the image to fewer nearest cluster centers. The intuition is that the multiple examples of a repeated feature provide a natural and accurate soft-assignment to multiple visual words.

Formally, an image d is represented by a bag-of-visual-words vector

$$\mathbf{r}_d = (r_1, ..., r_i, ..., r_V)^\top \qquad (4)$$

where the i-th visual word weight

$$r_i = \begin{cases} w_{id} & \text{if } 0 \le w_{id} < T \\ T & \text{if } T \le w_{id} \end{cases} \qquad (5)$$

is obtained by thresholding weights w_{id} by a threshold T. Note that the weighting described in equation (5) is similar to burstiness weighting, which down-weights repeating visual words. Here, however, we represent highly weighted (repeating) visual words with a constant T as the goal is

to represent the occurrence (presence/absence) of the visual word, rather than measuring the actual number of occurrences (matches).

Weight w_{id} of the i-th visual word in image d is obtained by aggregating weights from adaptively soft-assigned features across the image taking into account the repeated image patterns. In particular, each feature f from the set F_d of all features detected in image d is assigned to a k_f-tuple V_f of indices of the k_f nearest (in the feature space) visual words. Thus, $V_f(k)$ for $1 \le k \le k_f$ is the index of the k-th nearest visual word to f. The number k_f, which varies between 1 and $k_{\max}$, will be defined below. Weight w_{id} is computed as

$$w_{id} = \sum_{f \in F_d} \sum_{k=1}^{k_f} 1[V_f(k) = i] \frac{1}{2^{k-1}} \qquad (6)$$

where the indicator function $1[V_f(k) = i]$ is equal to 1 if visual word i is present at the k-th position in V_f. This means that weight w_{id} is obtained as the sum of contributions from all assignments of visual word i over all features in F_d. The contribution of an individual assignment depends on the order k of the assignment in V_f by the weight $1/(2^{k-1})$. The number k_f is computed by the following formula

$$k_f = \left\lceil k_{\max} \frac{\log(\frac{n_d+1}{m_f})}{\max_{f \in F_d} \log(\frac{n_d+1}{m_f})} \right\rceil \qquad (7)$$

where k_{max} is the maximum number of assignments ($k_{max} = 3$ in all our experiments), and m_f is the number of features in the repttile of f. We use $\lceil a \rceil = ceiling(a)$, i.e. $\lceil a \rceil$ is the smallest integer greater than or equal to a. Note that image features belonging to relatively larger repttiles are soft-assigned to fewer visual words as image repetitions provide a natural soft-assignment of the particular

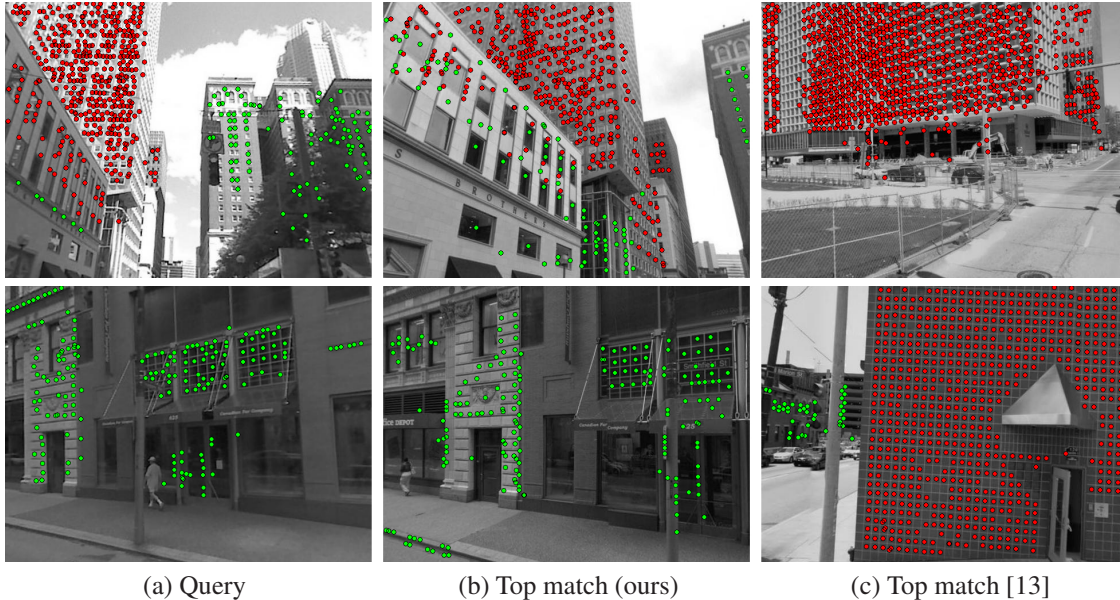|		
(a) Query	(b) Top match (ours)	(c) Top match [13]

Figure 3. **Examples of place recognition results on the Pittsburgh dataset.** Each row shows the query image (a), the best matching database image (b) correctly matched by the proposed method, and the best matching image (incorrect) using the baseline burstiness method [13] (c). The detected groups of repetitive features ("repttiles") are overlaid over the image and color-coded according to the number of visual word assignments k_f (red $k_f = 2$, green $k_f = 1$). Non-repetitive features ($k_f = 3$) are not plotted for the clarity. Note that the number of soft-assignments for each feature is adapted to the size of the repttile, where features in bigger repttiles are assigned to a smaller number of nearest visual words.

repeating scene element to multiple visual words. This natural soft-assignment is more precise and less ambiguous than the standard soft-assignment to multiple nearest visual words [27] as will be demonstrated in the next section.

5. Experiments

In this section we describe the experimental validation of our approach. First, we describe the experimental set-up and give the implementation details. Then we compare the place recognition performance of the proposed method with several baseline methods.
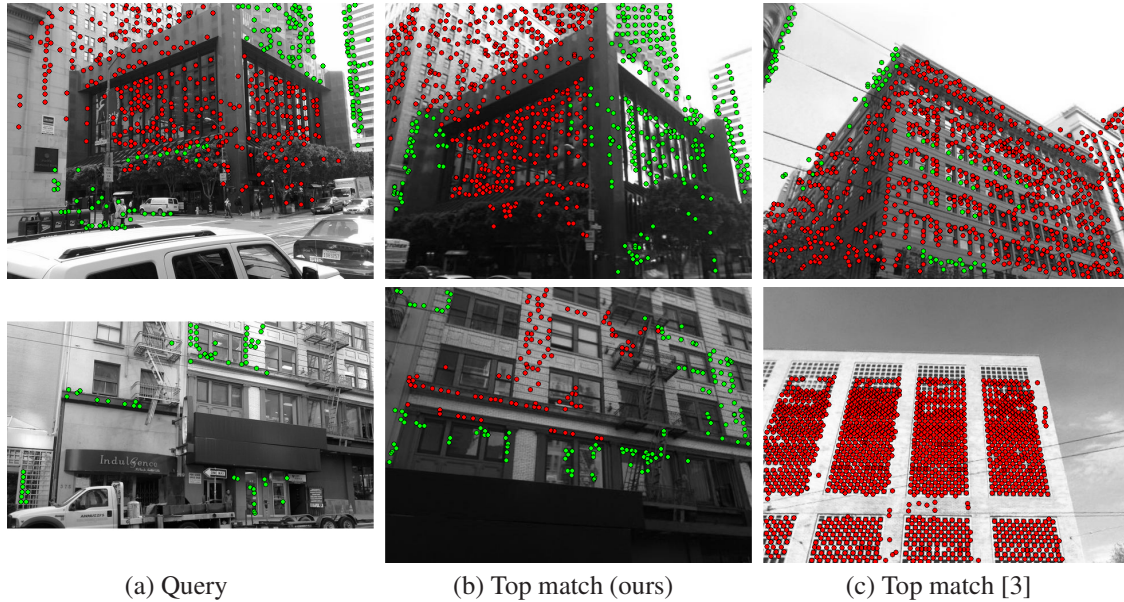
Experimental set-up. The geotagged image database is formed by $254,064$ perspective images generated from $10,586$ Google Street View panoramas of the Pittsburgh area downloaded from the the Internet. From each panorama of $6,656 \times 3,328$ pixels, we generate 24 perspective images of 640×480 pixels (corresponding to 60 degrees of horizontal FOV) with two yaw directions $[4, 26.5]$ and 12 pitch $[0, 30, ..., 360]$ directions. This is a similar setup to [3]. As testing query images, we use $24,000$ perspective images generated from $1,000$ panoramas randomly selected from $8,999$ panoramas of the Google Pittsburgh Research Data Set[1]. The datasets are visualized on a map in figure 5(a). This is a very challenging place recognition set-

[1]Provided and copyrighted by Google.

up as the query images were captured in a different session than the database images and depict the same places from different viewpoints, under very different illumination conditions and, in some cases, in a different season. But at the same time the ground truth GPS positions for the query test images are known. Note also the high number of test query images compared to other existing datasets [3, 18].

Implementation details. We build a visual vocabulary of 100,000 visual words by approximate k-means clustering [22, 26]. The vocabulary is built from features detected in a subset of $10,000$ randomly selected database images. We use the SIFT descriptors with estimated orientation for each feature (not assuming the upright image gravity vector) followed by the RootSIFT normalization [2].

Place recognition performance. We compare results of the proposed adaptive (soft-)assignment approach (Adaptive weights) with several baselines: the standard tf-idf weighting (tf-idf) [26], burstiness weights (brst-idf) [13], standard soft-assignment weights [27] (SA) and Fisher vector matching (FV) [16]. Following [16], we constructed Fisher vectors from SIFT descriptors reduced to 64 dimensions by PCA, and used 512 Gaussian mixture components. The Gaussian mixture models were trained on the same dataset, which was used to build the visual vocabulary. As in [16], resulting 512x64 dimensional descriptors
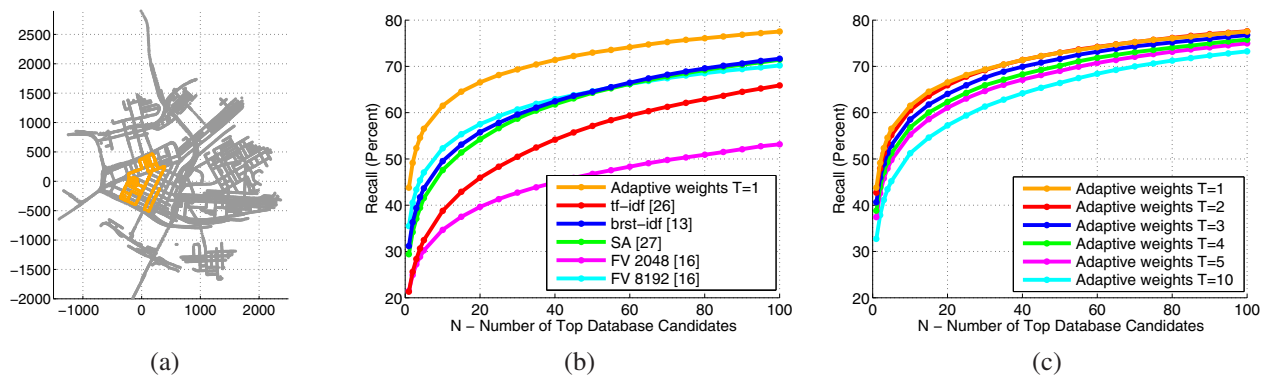
(a) Query   (b) Top match (ours)   (c) Top match [3]

Figure 4. **Examples of place recognition results on the San Francisco dataset.** Each row shows the query image (a), the best matching database image (b) correctly matched by the proposed method, and the best matching image (incorrect) using [3] (c). See the caption of figure 3 for details of feature coloring.

(a)   (b)   (c)

Figure 5. **Evaluation on the Pittsburgh dataset.** (a) Locations of query (yellow dots) and database (gray dots) images. (b-c) The fraction of correctly recognized queries (Recall, y-axis) vs. the number of top N retrieved database images (x-axis) for the proposed method (Adaptive weights) compared to several baselines.

are then reduced to 2048 (FV2048) or 8192 (FV8192) dimensions using PCA. For each method, we measure the percentage of correctly recognized queries (Recall) similarly to, e.g. [3, 18, 34]. The query is correctly localized if at least one of the top N retrieved database images is within m meters from the ground truth position of the query.

The ground truth is derived from the (known) GPS positions of the query images. We have observed that GPS positions of Street View panoramas are often snapped to the middle of the street. The accuracy of the GPS positions hence seems to be somewhere between 7 and 15 meters. Results for different methods for $m = 25$ meters and varying value of N are shown in figure 5 (b). Figure 3 shows examples of place recognition results.

Sensitivity to parameters. The weight threshold T in eq. (5) is an important parameter of the method and its setting may depend on the dataset and size of the visual vocabulary. In the Pittsburgh database, since 97 % of w_{id} are less or equal to 1, $T = 1$ effectively downweights unnecessary bursted visual words. Figure 5 (c) shows the evaluation of place recognition performance for different values of T. In the following we use $T = 1$ (unless stated otherwise).

Next, we evaluate separately the benefits of the two components of the proposed method with respect to the baseline burstiness weights: (i) thresholding using eq. (5) results in $+8.92\%$ and (ii) adaptive soft-assignment using eq. (6) and (7) results in $+10.30\%$. When the two are combined the improvement is $+11.97\%$. This is measured for the dis-
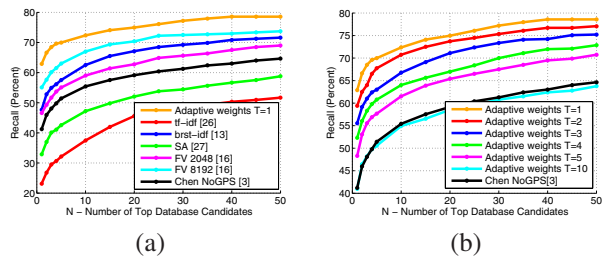
Figure 6. **Evaluation on the San Francisco [3] dataset.** The fraction of correctly recognized queries (Recall, y-axis) vs. the number of top N retrieved database images (x-axis) for the proposed method (Adaptive weights) compared to several baselines.

Table 1. mAP on INRIA Holidays and Oxford Building datasets. Here we use 200K visual vocabulary built from RootSIFT [2] features and $T = 5$ (different choices of T had small effect on the result).

	tf-idf [26]	brst-idf [13]	SA [27]	Proposed
INRIA	0.7364	0.7199	0.74838	**0.7495**
Oxford	0.6128	0.6031	0.6336	**0.6565**

tance threshold $m = 25$ meters and for the top $N = 10$ but we have observed that the improvements are consistent over a range of N (not shown).

Finally, we have also tested different parameters of the adaptive soft-assignment (eq. (6) and (7)). The method is fairly insensitive to the choice of the maximum number of assignments k_{max}, where values of 2 to 5 result in a similar performance. We use $k_{max} = 3$ following [27]. The base of the exponential in eq. (6) is chosen so that weights decrease with increasing k and we found $1/2$ work well. In general, this value needs to be set experimentally, similarly to the sigma parameter in the standard soft-assignment [27].

Scalability. Our adaptive soft-assignment can be indexed using standard inverted files and in terms of memory requirements compares favorably with respect to the standard soft-assignment and Fisher vector representation. Our tf-idf vectors are about 7.2% sparser than for [27] and the memory footprint is about 6.2% smaller than for the FV2048 representation while achieving better place recognition performance.

Evaluation on different datasets. We have also evaluated the proposed method on the San Francisco visual place recognition benchmark [3]. We have built a vocabulary of 100,000 visual words from upright RootSIFT [2] features extracted from 10,000 images randomly sampled from the San Francisco 1M image database [3]. We have not used the histogram equalization suggested by [3] as it did not improve results using our visual word setup. Performance is measured by the recall versus the number of top N database candidates in the shortlist as in figure 7(a) in [3]. Results

for the different methods are shown in figure 6. The results of [3] were obtained directly from the authors but to remove the effect of geometric verification we ignored the threshold on the minimum number of inliers by setting $T_{PCI} = 0$. Note also that the GPS position of the query image was not used for any of the compared methods. The pattern of results is similar to the Pittsburgh data with our adaptive soft-assignment method (Adaptive weights) performing best and significantly better than the method of [3] underlying the importance of handling repetitive structures for place recognition in urban environments. Example place recognition results demonstrating benefits of the proposed approach are shown in figure 4.

We have also evaluated the proposed method for retrieval on the standard INRIA Holidays [13] and Oxford Buildings datasets [26], where performance is measured by the mean Average Precision (mAP). Results are summarized in table 1 and demonstrate the benefits of the proposed approach over the baseline methods.

6. Conclusion

In this work we have demonstrated that repeated structures in images are not a nuisance but can form a distinguishing feature for many places. We treat repeated visual words as significant visual events, which can be detected and matched. This is achieved by robustly detecting repeated patterns of visual words in images, and adjusting their weights in the bag-of-visual-word representation. Multiple occurrences of repeated elements are used to provide a natural soft-assignment of features to visual words. The contribution of repetitive structures is controlled to prevent dominating the matching score. We have shown that the proposed representation achieves consistent improvements in place recognition performance in an urban environment. In addition, the proposed method is simple and can be easily incorporated into existing large scale place recognition architectures.

References

[1] B. Aguera y Arcas. Augmented reality using Bing maps., 2010. Talk at TED 2010.

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[3] D. Chen, G. Baatz, et al. City-scale landmark identification on mobile devices. In *CVPR*, 2011.

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

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

[6] 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.

[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.

[8] P. Doubek, J. Matas, M. Perdoch, and O. Chum. Image matching and retrieval by repetitive patterns. In *ICPR*, 2010.

[9] D. Hauagge and N. Snavely. Image matching using local symmetry features. In *CVPR*, 2012.

[10] J. Hays, M. Leordeanu, A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. In *ECCV*, 2006.

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

[12] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large-scale image search. In *ECCV*, 2008.

[13] H. Jegou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[14] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *PAMI*, 33(1):117–128, 2011.

[15] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, 2007.

[16] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *PAMI*, 34(9):1704–1716, 2012.

[17] S. Katz. Distribution of content words and phrases in text and language modelling. *Natural Language Engineering*, 2(1):15–59, 1996.

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

[19] T. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. In *ECCV*, 1996.

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

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

[22] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[23] P. Muller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. *ACM TOG*, 26(3):85, 2007.

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

[25] M. Park, K. Brocklehurst, R. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *PAMI*, 31(10):1804–1816, 2009.

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

[27] 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.

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

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

[30] A. Pothen and C.-J. Fan. Computing the block triangular form of a sparse matrix. *ACM Transactions on Mathematical Software*, 16(4):303–324, 1990.

[31] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *Proc. CIVR*, 2008.

[32] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), 1988.

[33] T. Sattler, B. Leibe, and L. Kobbelt. SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter. In *ICCV*, 2009.

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

[35] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In *BMVC*, 1998.

[36] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *CVIU*, 92:236–264, 2003.

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

[38] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*, 2008.

[39] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *PAMI*, 19(5):530–534, 1997.

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

[41] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *CVPR*, 2010.

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

[43] 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.

[44] J. C. van Gemert, C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *PAMI*, 32(7):1271–1283, 2010.

[45] C. Wu, J. Frahm, and M. Pollefeys. Detecting large repetitive structures with salient boundaries. In *ECCV*, 2010.

[46] C. Wu, J.-M. Frahm, and M. Pollefeys. Repetition-based dense single-view reconstruction. In *CVPR*, 2011.

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

[48] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011.