Panorama to panorama matching for location recognition

Ahmet Iscen[1] Giorgos Tolias[2] Yannis Avrithis[1] Teddy Furon[1] Ondřej Chum[2]
[1] Inria, Rennes, France
[2] Visual Recognition Group, Faculty of Electrical Engineering, CTU in Prague, Czech Republic

ABSTRACT

Location recognition is commonly treated as visual instance retrieval on "street view" imagery. The dataset items and queries are panoramic views, *i.e.* groups of images taken at a single location. This work introduces a novel panorama-to-panorama matching process, either by aggregating features of individual images in a group or by explicitly constructing a larger panorama. In either case, multiple views are used as queries. We reach near perfect location recognition on a standard benchmark with only four query views.

CCS CONCEPTS

•Information systems → Image search; •Computing methodologies → Visual content-based indexing and retrieval;

KEYWORDS

image retrieval, location recognition

ACM Reference format:
A. Iscen, G. Tolias, Y. Avrithis, T. Furon and O. Chum. 2017. Panorama to panorama matching for location recognition. In *Proceedings of ICMR '17, Bucharest, Romania, June 06-09, 2017,* 5 pages.
DOI: http://dx.doi.org/10.1145/3078971.3079033

1 INTRODUCTION

Location recognition has been treated as a visual instance retrieval task for many years [1, 3, 11, 23, 25, 33]. Additional, task-specific approaches include ground truth locations to find informative features [25], regression for a more precise localization [18, 31], or representation of the dataset as a graph [7]. A dense collection of multiple views allows 3D representations are possible, *e.g.* structured from motion [12], searching 2D features in 3D models [19, 24], or simultaneous visual localization and mapping [8]. However, this does not apply to sparse "street-view" imagery [30, 32], where dataset items and queries are groups of images taken at a single location, in a panorama-like layout.

Several approaches on visual instance retrieval propose to jointly represent a set of images. These sets of images can appear at the query or at the database side. In the former case, these images are different views of the same object or scene [2, 27] and finally performance is improved. This joint representation, which commonly is an average query vector constructed via aggregation, is presumably more robust than each individual query vector. On the other hand, when aggregating images on the database side it is better to group them together by similarity [14]; images are assigned to sets, and a joint representation is created per set.

This work revisits location recognition by aggregating images both on query and database sides. Our method resembles implicit construction of a panorama, *i.e.* images are combined in the feature space and not in the image space, but we also experiment with an explicit construction. Contrary to the general case of visual instance retrieval, it is easy to obtain multiple query images, e.g. capturing them with a smartphone or with multiple cameras in the case of autonomous driving. On the database side, location provides a natural way of grouping images together. Thus, contrary to generic retrieval, the images to be aggregated on the query and database sides, may not be similar to each other; they rather depict whatever is visible around a particular location.

We significantly outperform the state of the art without any form of supervision other than the natural, location-based grouping of images, and without any costly offline process like 3D reconstruction. Indeed we are reaching near perfect location recognition on the Pittsburgh dataset [32] even when we use as few as four views on the query side.

2 BACKGROUND

This section describes the related work on Convolutional Neural Network (CNN) based descriptors for image retrieval and on image set joint representations. Our approach applies these methods on the dataset and query images.

2.1 CNN Descriptors for Retrieval

CNN-based global descriptors are becoming popular in image retrieval, especially for instance-level search. Existing works [4, 17, 22, 29] employ "off-the-shelf" networks, originally trained on ImageNet, to extract descriptors via various pooling strategies. This offers invariance to geometric transformation and robustness to background clutter. Other approaches [5, 9, 20] fine-tune such networks to obtain descriptor representations specifically adapted for instance search.

NetVLAD [1] is a recent work that trains a VLAD layer on top of convolutional layers in an end-to-end manner. It is tuned for the location recognition task. The training images are obtained from panoramas, fed to a triplet loss to make it more compatible with image retrieval. As a result, their representation outperforms existing works in standard location recognition benchmarks.

2.2 Representing Sets of Vectors

Two common scenarios aggregate a set of vectors into a single vector representation for image retrieval. The first case involves aggregation of a large number of local descriptors, either to reduce

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
ICMR '17, Bucharest, Romania
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-4701-3/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3078971.3079033

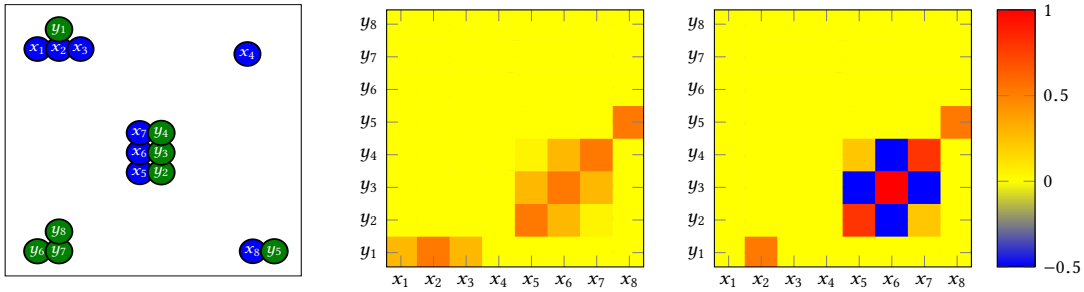

Figure 1: Left: Toy example of two vector sets X, Y on the 2D plane are shown on the left. Middle: Pairwise similarity between all vectors, cross-matching with sum-vectors, *i.e.* $\mathbf{X}^\top\mathbf{Y}$ (3). Only for visualization purposes, and since we are dealing with unnormalized 2D vectors, the similarity between vectors x, y is defined as $e^{\|\mathbf{x}-\mathbf{y}\|^2}$. Right: weighted pairwise-similarity between all vectors, cross-matching with pinv-vectors, *i.e.* $\mathbf{G}_\mathbf{X}^{-1}\mathbf{X}^\top\mathbf{Y}\mathbf{G}_\mathbf{Y}^{-1}$ (4).

the number of descriptors [26, 28], or to create a global descriptor [10, 16]. In the other case, which is exploited in this work, a set of global image descriptors is aggregated into a single vector representation to construct a joint representation for a set of images [14].

In particular, we follow the two *memory vector* construction strategies proposed by Iscen et al. [14]. The first method simply computes the sum of all vectors in a set. Given a set of vectors represented as the columns of a $d \times n$ matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ with $\mathbf{x}_i \in R^d$, the *sum* memory vector is defined as

$$m(\mathbf{X}) = \mathbf{X}\mathbf{1}_n. \tag{1}$$

Assuming linearly independent columns ($n < d$), the second method is based on the Moore-Penrose pseudo-inverse $\mathbf{X}^+$ [21], given by

$$m^+(\mathbf{X}) = (\mathbf{X}^+)^\top\mathbf{1}_n = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{1}_n. \tag{2}$$

It is theoretically optimized for high dimensional spaces and performs better in practice. This paper refers to the sum memory vector (1) as *sum-vector*, and to the pseudo-inverse memory vector (2) as *pinv-vector*.

Aggregating Dataset Images. The main purpose of aggregating dataset images is to reduce the computation cost of similarity search at query time [14]. Dataset vectors are assigned to sets in an offline process, and each set is represented by a single (memory) vector. At query time, the similarity between the query vector and each memory vector is computed, and memory vectors are ranked accordingly. Then the query is only compared to the database vectors belonging to the top ranked sets. This strategy eliminates the exhaustive computation of the similarities query vs. dataset vectors. Existing works use random assignments to create the sets, or weakly-supervised assignment based on k-means or kd-tree [13, 14].

Aggregating Query Images. Aggregation of query images has been also studied for instance-level object retrieval. Multiple images depicting the query object allow to better handle the problems of occlusion, view-point change, scale change and other variations. Arandjelovic et al. [2] investigate various scenarios, such as average or max pooling on query vectors and creating SVM models. Recently, Sicre and Jégou [27] have shown that aggregating query vectors with pinv-vector improves the search quality.

Aggregation of dataset images offers speed and memory improvements at the cost of performance loss. On the other hand, aggregation of query images is only applicable in the particular case of multiple available query images and offers performance improvements at no extra cost. Our approach adopts aggregation on both sides for the first time while enjoying speed, memory and performance improvements.

3 PANORAMA TO PANORAMA MATCHING

This section describes our contribution for location recognition. We assume that for each possible location we are given a set of images covering a full 360 degree view while consecutive images have an overlap (see Figure 2). We propose two ways to construct a *panoramic representation* of each location: an implicit way by vector aggregation and an explicit way by image stitching into a panorama and extraction of a single descriptor.

3.1 Implicit Panorama Construction

We form a panoramic representation by aggregating the descriptors of images from the same location. In this way, we implicitly construct a panorama in the descriptor space. In order to achieve this, we employ two approaches for creating memory vectors, *i.e.* sum-vector (1) and pinv-vector (2).

In contrast to previous works that aggregate the image vectors only on the dataset side [14] or only on the query side [27], we rather do it for both. This requires that the query is also defined by a set of images which offer a 360 degree view. A realistic scenario of this context is autonomous driving and auto-localization where the query is defined by such a set of images.

Assume that n images in a dataset location are represented by $d \times n$ matrix $\mathbf{X}$ and that k images in the query location by $d \times k$ matrix $\mathbf{Y}$. Analyzing the similarity between the two sum-vectors is straightforward. Panorama similarity is given by the inner product

$$s(\mathbf{X}, \mathbf{Y}) = m(\mathbf{X})^\top m(\mathbf{Y}) = \mathbf{1}_n^\top\mathbf{X}^\top\mathbf{Y}\mathbf{1}_k. \tag{3}$$

Similarly, panorama similarity for pinv-vectors is given by

$$s^+(\mathbf{X}, \mathbf{Y}) = m^+(\mathbf{X})^\top m^+(\mathbf{Y}) = \mathbf{1}_n^\top\mathbf{G}_\mathbf{X}^{-1}\mathbf{X}^\top\mathbf{Y}\mathbf{G}_\mathbf{Y}^{-1}\mathbf{1}_k, \tag{4}$$

where $\mathbf{G}_\mathbf{X} = \mathbf{X}^\top\mathbf{X}$ is the Gram matrix for $\mathbf{X}$. Compared to (3), the sum after cross-matching is weighted now, and the weights are given by $\mathbf{G}_\mathbf{X}^{-1}$ and $\mathbf{G}_\mathbf{Y}^{-1}$. This is interpreted as "democratizing" the result of cross-matching; the contribution of vectors that are similar within the same set are down-weighted, just as in handling the burstiness phenomenon for local descriptors [16]. We visualize this with a toy example in Figure 1. Unweighted cross-matching is

Figure 2: Example of all images assigned to a single location (first two rows) and the corresponding panorama (last row) covering a full 360 degree view, constructed by automatic stitching.

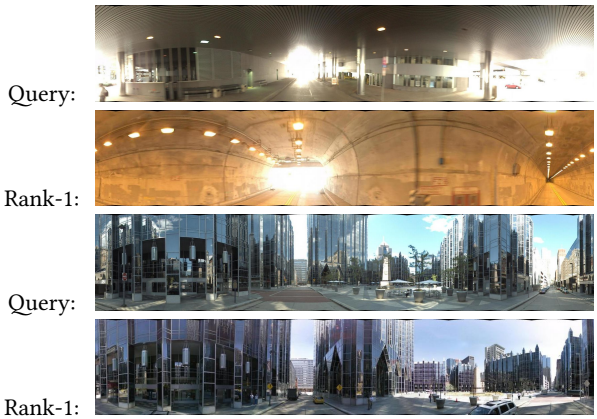

Figure 3: Two examples of failures with *pan2pan/net*. We show the query and the top ranked image from the dataset.

dominated by "bursty" vectors in the same cluster. Democratization down-weights these contributions.

3.2 Explicit Panorama Construction

Our second approach explicitly creates a panoramic image. The descriptors are then extracted from the panorama. Given that images of a location are overlapping, we construct a panoramic image using an existing stitching method. In particular, we use the work of Brown and Lowe [6], which aligns, stitches, and blends images automatically based on their local SIFT descriptors and inlier correspondences. Figure 2 shows a stitched panoramic image. Once stitching is complete, we extract a single global descriptor from the panorama image, capturing the entire scene.

4 EXPERIMENTS

In this section, we describe our experimental setup, and compare our method to a number of baselines using the state-of-the-art NetVLAD network in a popular location recognition benchmark.

4.1 Experimental Setup

The methods are evaluated on the Pittsburgh dataset [32] referred to as Pitt250k. It contains 250k database and 24k query images from Google Street View. It is split into training, validation, and test sets [1]. We evaluate our approach on the test set, which consists of 83,952 dataset images and 8,280 query images. Each image is associated with a GPS location and 24 images are associated with the same GPS location. Therefore, each panoramic representation aggregates 24 images. There is a total of 345 query locations and 3,498 dataset locations. We use NetVLAD for our descriptor representation in all experiments. While the original representation is $d = 4{,}096$ dimensional, we also experiment with reducing dimensionality to $d = 256$ by PCA.

The standard evaluation metric is Recall@N. It is defined to equal 1 if at least one of the top N retrieved dataset images is within 25 meters from the spatial location of the query. Average is reported over all queries. We follow this protocol for the baseline and other cases where the query images are used individually.

Aggregating on the query side implies that there is a single query per location: the number of queries decreases from 8,280 to 345. We report the average recall@N from these 345 panorama queries. Section 4.3 also experiments with a larger number of random queries, each capturing only a fraction of the panoramic view. In this case, recall@N is averaged over those random queries. Aggregating on the dataset side does not affect the standard evaluation.

4.2 Panorama Matching

We refer to our proposed method as panorama to panorama or *pan2pan* matching, in particular *pan2pan/sum* and *pan2pan/pinv* when aggregating descriptors with sum-vector and pinv-vector respectively; and as *pan2pan/net* when using a NetVLAD descriptor from an explicit panorama. We compare against the following baselines: image to image matching (*im2im*) as in the work by Arandjelovic et al. [1], image to panorama matching (*im2pan*) corresponding to dataset-side aggregation as in the work by Iscen et al. [14], and panorama to image matching (*pan2im*) corresponding to query-side aggregation as in the work by Sicre and Jégou [27].

Figure 4 compares all methods for different descriptor dimensions. Clearly, panorama to panorama matching outperforms all other methods. The improvement is consistent for all N and significant for low N: pan2pan/net obtains 98% recall@1! There are only 7 failure queries. Two of them are shown in Figure 3. One is a challenging query depicting an indoor parking lot and the other

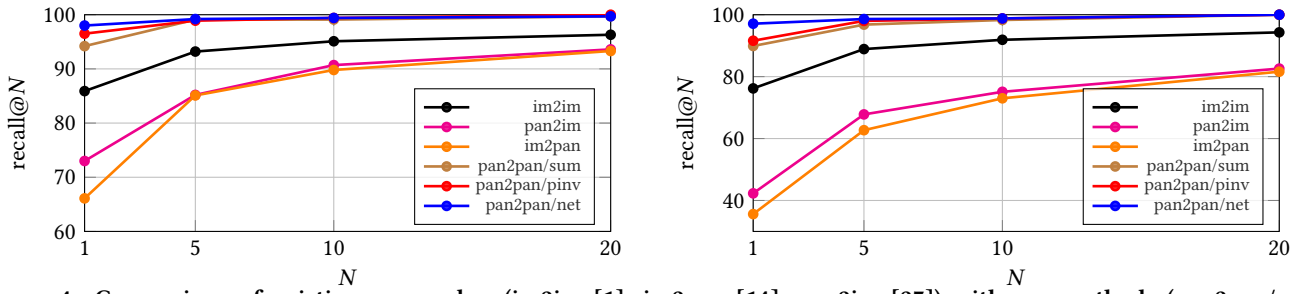

Figure 4: Comparison of existing approaches (im2im [1], im2pan [14], pan2im [27]) with our methods (pan2pan/sum, pan2pan/pinv and pan2pan/net) for the full 4096D (left) and for reduced dimensionality to 256D (right).

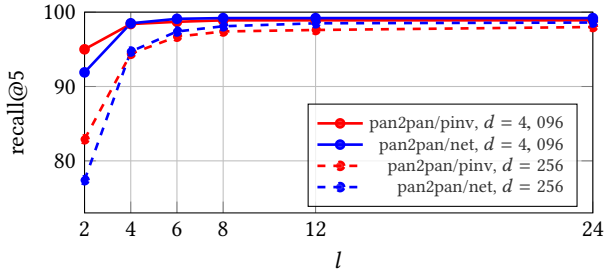

Figure 5: Recall@5 on Pitt250k, sampling l images from each query panorama and using NetVLAD descriptors of two different dimensionalities d. We report average measurements over 10 random experiments and compare our methods pan2pan/pinv and pan2pan/net.

actually retrieves the same building, which is incorrectly marked in the dataset's ground truth.

The recall is not only improved, but the search is also more efficient both speed-wise ($24^2\times$ faster) and memory-wise (24× less memory). Instead of comparing a given query image against 83k vectors, we only make 3.5k comparisons. Additional operations are introduced when aggregating the set of query images, but this cost is fixed and small compared to the savings from the dataset side.

Comparing to results in prior work, im2pan behaves as in the work of Iscen et al. [14] when compared to the baseline im2im. That is, memory compression and speed up at the cost of reduced performance. However, pan2im does not appear to be effective in our case, in contrast to the work of Sicre and Jégou [27]. On the contrary, pan2pan significantly improves the performance while enjoying both memory compression and search efficiency.

4.3 Sparse Panorama Matching

Aggregating on the dataset side is performed off-line. However, the user is required to capture images and to construct a full panorama (24 images in our case) at query time. Even though this is not a daunting task given the advances of smartphones and tablets, we additionally investigate a scenario where the user only captures a partial panoramic view.

In particular, we randomly sample a subset of l images from the query location and consider them as the query image set. Explicit panorama construction is no longer possible because the sampled images may not overlap and so we cannot stitch them. In this case, we feed sampled images through the convolutional layers only, and stack together all activations before pooling them through the NetVLAD layer (pan2pan/net for sparse panoramas).

Figure 5 shows the results. Our methods have near-perfect performance even for a small number of sampled images. When the user only takes four random photos, we are able to locate them up to 99% recall@5. Another interesting observation is that pan2pan/pinv outperforms pan2pan/net for $l = 2$, which is expected due to the nature of pinv-vec construction. It is theoretically shown to perform well even if all the vectors in the set are random, as shown in the original paper [14].

4.4 Comparison to Diffusion-based Retrieval

This work casts location recognition as a retrieval task. Query expansion techniques significantly improve retrieval performance. We compare to the state-of-the-art retrieval method by Iscen et al. [15], a kind of query expansion based on graph diffusion. In this method, an image is represented by individual region descriptors and at query time all query regions are processed.We compare to this method by considering that regions and images in [15] correspond to images and panoramas respectively in our scenario.

Our pan2pan/pinv and pan2pan/net approaches achieve 96.5% and 98% recall@1 respectively, while the approach [15] gives 91.9%. Even though query expansion improves the baseline, it does not help as much as our methods. This can be expected because [15] is based on many instances of the same object, which is not the case for location recognition on street view imagery.

5 CONCLUSIONS

Our method is unsupervised and conceptually very simple, yet very effective. Besides the performance gain, we make significant savings in space by aggregating descriptors of individual images over each group. The need for multiple query views is not very demanding because only four views are enough—an entire query panorama is definitely not needed.

Although our aggregation methods have been used for instance retrieval in the past, we are the first to successfully aggregate on both dataset and query-side for location recognition (which in fact has failed for instance retrieval [26]). An interesting finding is that although the NetVLAD descriptor has been explicitly optimized to aggregate CNN activations on the location recognition task, in some cases it is preferable to aggregate individual views into a pinv-vector rather than extracting a single NetVLAD descriptor from an explicit panorama.

Acknowledgments. The authors were supported by the MSMT LL1303 ERC-CZ grant. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

REFERENCES

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

[2] R. Arandjelovic and A. Zisserman. Multiple queries for large scale specific object retrieval. In *BMVC*, 2012.

[3] R. Arandjelovic and A. Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *ACCV*, 2014.

[4] A. Babenko and V. Lempitsky. Aggregating deep convolutional features for image retrieval. In *ICCV*, 2015.

[5] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014.

[6] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 74:59–73, 2007.

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

[8] M. Cummins and P. Newman. Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *ICML*, 2010.

[9] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. *ECCV*, 2016.

[10] P.-H. Gosselin, N. Murray, H. Jégou, and F. Perronnin. Revisiting the fisher vector for fine-grained classification. *Pattern recognition letters*, 49, 2014.

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

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

[13] A. Iscen, L. Amsaleg, and T. Furon. Scaling group testing similarity search. In *ICMR*, 2016.

[14] A. Iscen, T. Furon, V. Gripon, M. Rabbat, and H. Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE Trans. Big Data*, 2017.

[15] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*, 2017.

[16] H. Jégou and A. Zisserman. Triangulation embedding and democratic kernels for image search. In *CVPR*, June 2014.

[17] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCVW*, 2016.

[18] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015.

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

[20] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. *ECCV*, 2016.

[21] C. R. Rao and S. K. Mitra. Generalized inverse of a matrix and its applications. In *Sixth Berkeley Symposium on Mathematical Statistics and Probability*, 1972.

[22] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4, 2016.

[23] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *CVPR*, 2016.

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

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

[26] M. Shi, Y. Avrithis, and H. Jégou. Early burst detection for memory-efficient image retrieval. In *CVPR*, 2015.

[27] R. Sicre and H. Jégou. Memory vectors for particular object retrieval with multiple queries. In *ICMR*, 2015.

[28] G. Tolias, Y. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, December 2013.

[29] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *ICLR*, 2016.

[30] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.

[31] A. Torii, J. Sivic, and T. Pajdla. Visual localization by linear combination of image descriptors. In *ICCVW*, 2011.

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

[33] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006.