Efficient Search in a Panoramic Image Database for Long-term Visual Localization

Semih Orhan
Department of Computer Engineering
Izmir Institute of Technology
semihorhan@iyte.edu.tr

Yalın Başstanlar
Department of Computer Engineering
Izmir Institute of Technology
yalinbastanlar@iyte.edu.tr

Abstract

In this work, we focus on a localization technique that is based on image retrieval. In this technique, database images are kept with GPS coordinates and the geographic location of the retrieved database image serves as an approximate position of the query image. In our scenario, database consists of panoramic images (e.g. Google Street View) and query images are collected with a standard field-of-view camera in a different time. While searching the match of a perspective query image in a panoramic image database, unlike previous studies, we do not generate a number of perspective images from the panoramic image. Instead, taking advantage of CNNs, we slide a search window in the last convolutional layer belonging to the panoramic image and compute the similarity with the descriptor extracted from the query image. In this way, more locations are visited in less amount of time. We conducted experiments with state-of-the-art descriptors and results reveal that the proposed sliding window approach reaches higher accuracy than generating 4 or 8 perspective images.

1. Introduction

Visual localization can be defined as estimating the position of a visual query material within a known environment. Visual localization approaches have attracted an increasing attention [18] especially due to the limitation of GPS-based localization in urban environment (e.g. signal failure due to tall buildings or a cluttered environment).

The localization technique that we employ is based on image retrieval, in which the geographic location of the retrieved image serves as an approximate position of the query image. Last decade witnessed many computer vision techniques proposed to solve this type of visual localization problem ([30, 29, 1]). Especially if the database and query images are collected at different seasons/years, referred as long-term localization ([27, 16, 25]), there are numerous challenges such as illumination variations, weather conditions, seasonal changes, viewpoint variations and changing objects in the scene. Any method to solve city-scale localization problem should address these long-term appearance changes.

In our setting, the environment is represented by a set of images acquired at different locations. Thus, rather than a metric localization, it is a topological localization (also called as location recognition or place recognition [10, 2, 7, 14]) that could help the navigation of mobile agents. In our work, we use 360 degree vision, panoramic images in particular. The reason to use panoramic images is to exploit their wide field-of-view (FOV) advantage. With this advantage, recognition of correct location can be achieved in some scenarios where standard FOV cameras fail due to their non-overlapping fields of view.

Our main contribution is that, while matching a perspective (standard FOV) image in a database consisting of panoramic images, unlike previous studies, we do not produce a number of virtual perspective images from the panoramic image. Instead, taking advantage of employing CNNs, we slide a search window in feature maps obtained from the panoramic image. Similarity between the descriptors extracted from the feature maps of database and query images is computed. In this way, many more positions are visited in a panoramic image which increases the probability of a healthy match with the query image. We conducted experiments with three different state-of-the-art descriptors and showed the superiority of the proposed approach.

The paper is organized as follows. In Section 2, the related work is summarized and our novelty is clarified. In Section 3, our novel sliding window approach is outlined, as well as the preparation of the dataset used in the experiments is explained. The design and the results of the experiments are presented in Section 4. Section 5 concludes the paper.

2. Related Work

We will present the related work in two parts. First, we will review image-based localization before and after CNN era. Second, we will focus on localization studies that exploit panoramic images.

Before the invasion of CNN based methods, image retrieval based localization techniques mostly depend on Bag-of-Features approaches [17], where SIFT-like [13] descriptors extracted from all images in the database are clustered to define a set of 'visual words', then the images are represented with those visual words. VLAD (Vector of Locally Aggregated Descriptors) [11] managed to do the same task with compact representations which enabled us to use large datasets. In time, researchers proposed techniques that are more robust to repetitive structures [30], illumination and viewpoint changes [29], and even changes over time such as seasonal changes.

Recent studies consider using features from the deep convolutional layers of CNNs. [2] compares different feature extraction techniques for CNNs, it also gives a comparison with non-CNN methods. NetVLAD [1] adds a layer to a standard CNN that converts last convolutional layer to a compact descriptor to mimic the behavior of VLAD [11]. Recently, a region similarity based method (SFRS, [29]) has outperformed NetVLAD and other previous approaches on several visual localization benchmark datasets.

Another family of CNN-based methods have especially focused on image retrieval (E.g. Is this the Eiffel Tower?) rather than accurate localization. These methods also relate to us since ours is a topological localization task. Tolias *et al.* [28] proposed to use regional max-pooling of CNN activations for instance retrieval (R-MAC) and Radenovic *et al.* [20] improved this idea by proposing a trainable Generalized-Mean (GeM) pooling layer rather than using maximum activations. An extensive comparison provided in [19] depicts the success of GeM on image retrieval benchmark datasets.

The studies mentioned above did not use 360° imagery, they matched perspective (standard FOV) query images with perspective images in the database. However, wide FOV can overcome the difficulties when viewing angles of perspective images do not overlap. An example scenario is given in Figure 1, where a perspective camera is attached on top of a car. In this case, database consists of images taken with a single orientation (moving direction of the car). If the query image is captured with a different orientation, scene will be completely different. This scenario is actually very common, benchmark datasets that researchers use for image-based localization are mostly constructed in this manner [22].

When we consider the previous work on localization/place recognition with 360 degree imagery, we can group them into two. In the first group, both database and query images are panoramic ([7, 15, 8, 14, 31, 3, 10, 12]). Some researchers directly work on omnidirectional images (dough-nut images obtained with an omnidirectional sensor), others work after converting them to panoramic images. In this first group of studies, problem turns into panorama-to-panorama image matching. Some use SIFT, GIST or originally proposed descriptors, whereas recent ones use features extracted via CNNs ([31, 3, 10, 12]).

(a) (b)

Figure 1. Typical scenario that database images are collected with a perspective camera attached on top of a car. a) An image in database. b) An image taken by a car moving in the opposite direction but at the same point where image in (a) is taken. When query image is (b), it is not possible to correctly match it with database images

For the studies in the second group, database is composed of panoramic images and query images are taken with a standard FOV camera ([24, 32, 9]). We find this scenario more realistic since panoramic database images can be collected offline (e.g. Google Street View panoramas), whereas query images can be taken by a standard camera in a car or another mobile agent. In the previous studies, panoramic database images were used to generate 4 or 8 gnomonic projections along the equator. A gnomonic projection is a virtual perspective image, where image plane is tangent to the sphere of the observant. Thus, 4 non-overlapping gnomonic projections (each having 90° FOV) corresponds to the cubemap representation (used for localization in [32]). Figure 2b shows 4 images generated in this way. In another study, 8 non-overlapping gnomonic projections (each with 45° FOV) were used [9].

The approach of generating perspective views actually converts the problem into a perspective-to-perspective matching, but with increased number of images. This strategy is unable to provide good matching since quite a number of query images (e.g. Figures 2c and 2d) do not have a good overlap ratio with virtual perspective images. Generating more perspective images with overlapping FOV would alleviate the overlap problem in Figures 2c and 2d. However, that also increases the computation time. Different from the previous work, we propose to search query image in the panoramic image without generating gnomonic images. As will be explained in Section III, we do this via a

(a)

(b)

(c) (d)

Figure 2. a) An equirectangular panoramic image. b) 90° FOV perspective images generated from the panoramic image at four different orientations (0°, 90°, 180° and 270°). c) A query image belonging to 45° orientation. d) Another query image belonging to 225° orientation. Query images (c) and (d) are considerably different from the closest cubemap images. This will result in a poor matching performance. Also notice the illumination changes in (d) which makes the problem even harder.

search window on the CNN feature maps obtained from the panoramic image. In this way, we check many locations in the panoramic image with a very low computational cost.

3. Methodology and Dataset

3.1. Preparing the dataset

Panoramic images in our dataset were obtained from Google Street View (images of 2019) and downloaded via Street View Download 360 application[1]. Perspective query images belong to the same spots and they are a subset of a larger dataset provided by UCF [33] which were collected from Google Street View before 2014. This time gap results in not only illumination changes but also some seasonal and structural changes (e.g. change of a facade of a building) between database and query images and conforms better to the long-term localization scenario ([27, 16, 22]). The images are from the downtown area of Pittsburgh, PA. We collected database and query images at 80 different positions

[1] iStreetView.com

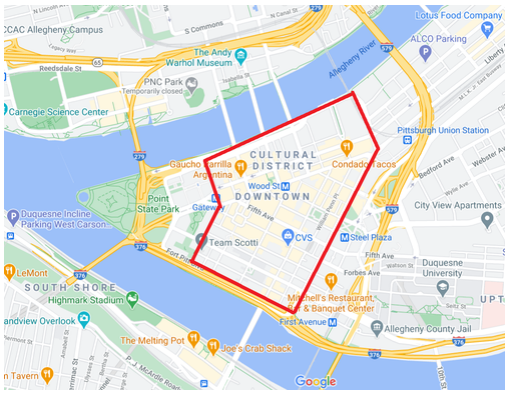


Figure 3. The area in central Pittsburgh, PA where database and query images in our dataset were collected (source: Google Maps).

which are spread to 0.6 km^2 (Figure 3). Each position has one panoramic database image and four perspective query images (UCF dataset) each covering a 90° FOV (without overlap).

3.2. Searching a perspective query image within a database of panoramic images

We search the perspective query image within the regions of panoramic images using convolutional layers of CNN (Figure 4). In other words, we encode regions of CNN feature maps into feature vectors (descriptors). This is performed in a sliding windows manner. This way, a panoramic image is processed without converting it into several virtual perspective images and it will not be fed into CNN several times. Our panoramic images are equirectangular. I.e. from top to bottom, it covers +90° to -90° vertical viewing angle and from left to right, it covers -180° to 180° horizontal viewing angle. Since query images are vertically centered around horizon, we only slide our search window along the equator (see Figure 4a).

To encode CNN feature maps into descriptors, several techniques can be used. First family of these techniques pool the features in various ways ([28, 20, 21]). Let X be a WxHxK tensor corresponding to the feature map in the last convolution layer of CNN. X_i represents a single 2D activation plane in the feature map, where $i = 1, ..., K$ and $X_i(p)$ is the response at position p. If we select the maximum value in X_i (Eq.1), this results in a K-size feature vector for the image (MAC, [21]).

$$\mathbf{f} = [f_1, ...f_i, ...f_K]^T, \quad f_i = \max_{p \in X_i} X_i(p) \qquad (1)$$

This simple idea was improved in [28] (R-MAC) by encoding multiple image regions into the same feature vector and in [20] by proposing a trainable Generalized-Mean (GeM) pooling layer rather than using maximum or average

of activations (Eq. 2).

$$\mathbf{f} = [f_1, ...f_i, ...f_K]^T, \quad f_i = \left(\frac{1}{|X_i|} \sum_{p \in X_i} X_i(p)^{c_i} \right)^{\frac{1}{c_i}} \tag{2}$$

It is shown in [20] that c_i is learnable and GeM pooling behaves as max-pooling [21] when $c_i \rightarrow \infty$. Radenovic *et al.* [20] trained GeM with a structure-from-motion based approach defined in [23]. Training samples are derived from 7.4 million Flicker images which consist of popular landmarks, cities, and country images. Since it's well suited for most image retrieval tasks, while employing GeM we did not perform an additional training with our dataset.

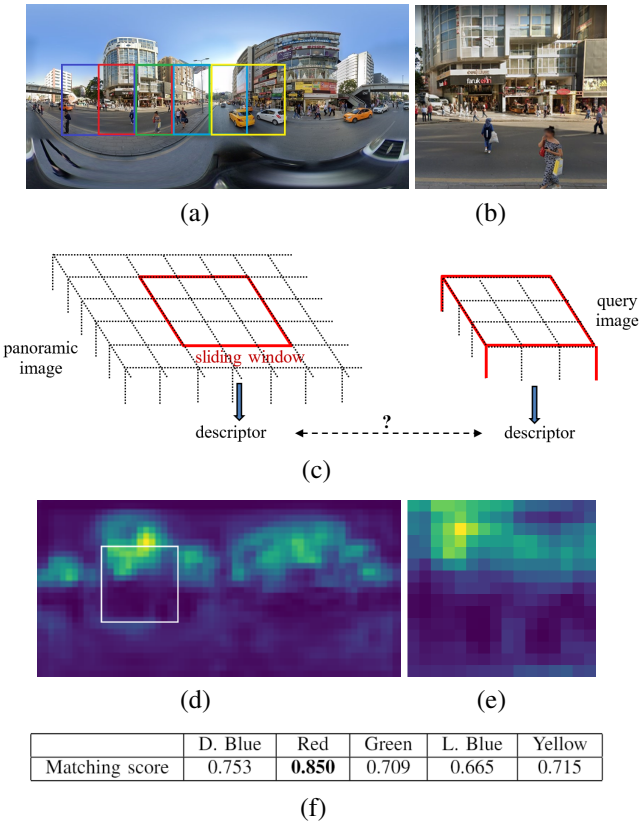
A second family of feature extraction techniques use a training set of images taken at different times at closeby spatial locations. This is cast as a weak supervision since the annotations can be noisy due to position shifts and limited overlap between views. NetVLAD [1] uses a triplet ranking loss where positive and negative samples are arranged according to their distance to the anchor image. Recently, a self-supervised method benefits from image-to-region similarities (SFRS[6]), especially designed to deal with noisy labels, has outperformed previous approaches on visual localization benchmark datasets. Section IV will present our results with R-MAC[28], GeM[20] and SFRS[6].

The descriptors have backbone CNNs that accept varying size input and produce convolution layers accordingly. Thus, we are able to give panoramic and perspective images to the same CNN and receive large feature maps for panoramic database images whereas small feature maps for queries. Figure 4d shows the sum of the values in the 31x62x512 feature map of a 16-layer CNN for the panoramic image. Figure 4e shows the same for the query image and its feature map size is 16x16x512.

Some previous studies (e.g. [26, 4, 5]) that worked on semantic segmentation or object detection in panoramic images trained special CNNs to handle the distortions which become substantial especially towards the top and bottom of the panoramic image (north and south poles). This is crucial, for instance, if you detect objects close to upper or lower side of the image. However, it is not crucial for our task since we only search along the equator (since query images cover that area) where little distortion exists. Moreover, training a CNN with limited number of panoramic images cannot reach the performance of CNNs that was trained with huge datasets. Thus, we employed the pretrained descriptors as they are.

4. Experimental Results and Discussions

As mentioned in Section 2, previous work on searching perspective query images in a panoramic image database converts panoramas into 4 or more gnomonic projections, then performs perspective-to-perspective matching. Therefore, we compare our sliding window method with that approach. Initially, we compare with cubemap approach that corresponds to non-overlapping 4 gnomonic 90° projections (cf. Figure 2b). For each query image, a best match (providing highest matching score) panoramic image is obtained. If this panoramic image belongs to the same position, retrieval is considered as successful. For the same query image, the best matching database cubemap image is obtained. If this cubemap is from the same position, retrieval is considered as successful. We do not check how much overlap actually



	D. Blue	Red	Green	L. Blue	Yellow
Matching score	0.753	**0.850**	0.709	0.665	0.715

(f)

Figure 4. a) A 500x1000 panoramic image in the database, five different colored sliding windows which have the same size with the query image in (b). Red window actually matches with the query image. b) Perspective query image (250x250). In this example (a) is the correct image to be retrieved when (b) is given as query. c) Feature maps in CNN convolution layers are represented when panoramic and query images are given as input. Different size feature maps are obtained when different size images are given to CNN. d) Sum of the values in 31x62 feature map of the image shown in (a). e) Sum of the values in 16x16 feature map of the image shown in (b). Although these are not the descriptor vectors, one can observe the high values in corresponding regions. f) Descriptor (GeM, [20]) matching score between the query and the sliding window in (a). Highest score comes from the correct position in the panoramic image (red window).

	Accuracy Part-1 (%)	Accuracy Part-2 (%)	Accuracy Merged (%)
4 gnomonic views (Cubemap)	56.9	64.4	**54.7**
8 overlapping gnomonic views	70.0	80.0	**67.5**
Sliding window 14x14 stride=2	70.0	81.9	**72.2**
Sliding window 14x14 stride=3	67.5	81.9	70.6
Sliding window 16x16 stride=2	67.5	76.9	69.7
Sliding window 16x16 stride=3	67.5	74.4	67.8

Table 1. Retrieval Accuracies with GeM Pooling [20] (Database: GoogleStreetView and Query: UCF Pittsburg)

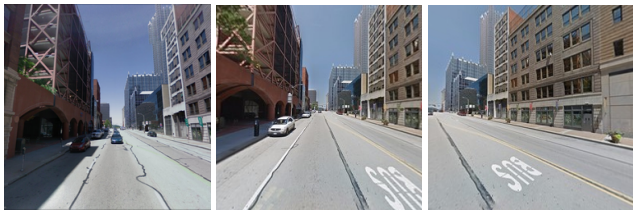
exist between query and retrieved cubemap.

Accuracy of the cubemap approach depends on how much of the query image's field of view (FOV) overlaps with the cubemap image's FOV. In a lucky case, there is a perfect overlap, but in an unlucky case, only 45° of 90° FOV overlap (Figure 5). To be fair, while generating cubemap projections we equally covered a range of overlaps from the best (90°) to the worst (45°). More specifically, for 80 positions in the experiment, 20 of them have 90° overlap, 20 have 75° overlap, 20 have 60° overlap and 20 have 45° overlap.

Images of part-1 and part-2 datasets are collected from 40 positions each. The merged dataset is created by merging part-1 and part-2 adding up to 80 positions. For the merged dataset, the last column of Table 1 reports the retrieval accuracy of the cubemap approach together with the accuracies of the proposed sliding window approach, where GeM was used as the descriptor for both approaches. Table 1 also depicts the effect of the sliding window size and the stride size for our approach. The best performance (72.2%) is obtained with 14x14 windows and stride=2, but even with other parameters, sliding window approach significantly outperforms cubemap approach (54.7%).

Accuracies are generally higher for small datasets (part-1 and part-2) as less number of alternatives exist to compete with the true match (first two columns of Table 1). However, relative success of the proposed approach stays same.

Another comparison is made with 8 overlapping gnomonic projections (90° FOV images, each overlaps 45° with the next one). This alleviates the reduced overlap problem in Figure 5c because it guarantees to have a considerable overlap with the query images (cf. Figure 6). While generating 8 gnomonic views from our dataset, we prepared equal number of examples with the best (90°) and the worst



(a) (b) (c)

Figure 5. a) Query image (UCF dataset), b) A 90-degree overlapping cubemap (best case), c) A 45-degree overlapping cubemap image (worst case)

(67.5°) overlaps. Results are in the second row of Table 1. As expected, accuracy increased (67.5%) when compared to cubemap approach, however it is still below the proposed method. In Figure 6, we see two examples from the dataset where 4-gnomonic and 8-gnomonic methods fail but the proposed method retrieves the correct image.

We also evaluated the compared methods according to Recall@N metric (Figure 7a), where retrieval is considered successful if at least one of the top N retrieved dataset locations is correct. The proposed method remained on top while success increases with N.

Other Descriptors.

Results presented so far were obtained with GeM[20]. Figures 7b and 7c show the results for R-MAC[28] and SFRS[6] respectively. Please observe that the relative performance of the compared methods did not change, which proves the superiority of the proposed method. We also observe a significant performance increase for all methods with SFRS. This success is partly due to the fact that SFRS model we employed was trained with a dataset of Pittsburgh images, namely Pitts30k-train (cf. [6] for details).

Computation Cost.

Table 2 shows average descriptor extraction costs for the proposed sliding window, cubemap and 8-gnomonic approaches on our database of 80 positions. The proposed sliding window approach runs 2 times faster than 8-gnomonic approach and performs better. Please note that the reported times increase proportionally with larger dataset (e.g. there are 3500 positions in Pitts250k-test [30]) and with higher image resolutions. The search time of both methods took milliseconds, hence it was not added to the table.

Experiments were run on a computer with an Intel i7-8700K processor, a memory of 16 GB and an NVIDIA GeForce GTX 1080 graphics processing unit.

5. Conclusions and Future Work

In this work, we search perspective query images in a panoramic image database for long-term localization in an urban environment. In our setting, the environment is rep-

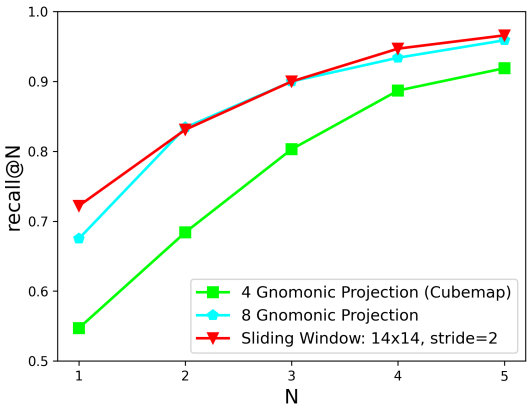
(a)

(b)

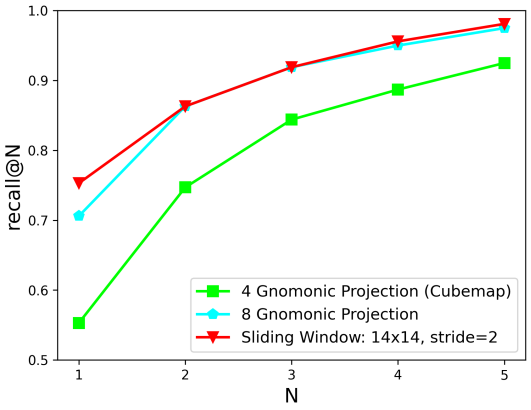
Figure 6. Two examples from the dataset where 4-gnomonic and 8-gnomonic approaches fail but the proposed method retrieves the correct image. In both (a) and (b), upper-right corner shows the query image and the bottom row shows generated 8 gnomonic views. In (a), overlap between query and gnomonic images is not perfect, also there is an illumination difference. In (b), overlap between query and gnomonic images is perfect but there are illumination, viewpoint and long-term changes (e.g. vegetation).

resented by a set of images acquired at different locations. Thus the developed method can be seen as an assistive localization technique rather than a complete navigation system of a mobile agent.

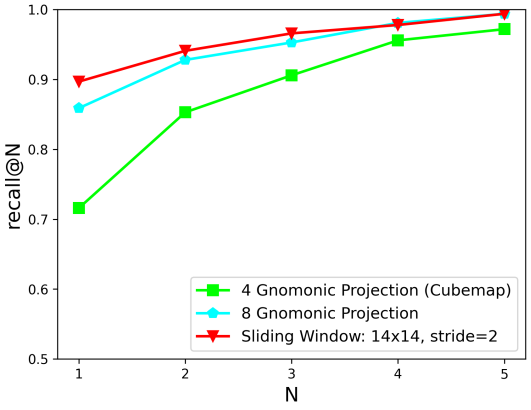
Database and query images are collected from Google Street View and covers a variety of appearance changes such as illumination variations and seasonal changes. As a novelty, we proposed to search query images in a panoramic image database via sliding windows on feature maps of CNN. We compared our approach with the classical approach of converting panoramic images into several gnomonic projections before searching. We conducted experiments with three state-of-the-art descriptors and showed that the sliding window approach performs better than producing cubemap (4 non-overlapping) or more frequent (8 overlapping) images while requiring less computation time. One can suggest to generate a higher number of overlapping perspective images to increase the chance of good matching. However, that would also increase the cost. As conclusion, while matching query images and panoramic database



(a)



(b)



(c)

Figure 7. Recall@N graphs for the merged dataset (80 locations) with descriptors a) GeM[20], b) R-MAC[28] and c) SFRS [6].

images, we advise to perform the search on the feature maps of CNN. Actually, most of the applications that require search of a narrow FOV image in large FOV images can benefit from the proposed approach.

In the future, depth maps or semantic information extracted from the images can be exploited to increase the localization accuracy.

Process	Computation Cost
Descriptor extraction from our database for 500x1000 panoramic images	2.04 sec.
Descriptor extraction from our database for 4 gnomonic images, 250x250 each (Cubemap)	1.96 sec.
Descriptor extraction from our database for 8 gnomonic images, 250x250 each	3.97 sec.

Table 2. Descriptor Extraction Cost of Sliding Window and Gnomonic Projection Approaches

Acknowledgement

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), Grant No: 120E500.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 1, 2, 4

[2] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230, 2017. 1, 2

[3] R. Cheng, K. Wang, S. Lin, W. Hu, K. Yang, X. Huang, H. Li, D. Sun, and J. Bai. Panoramic annular localizer: Tackling the variation challenges of outdoor localization using panoramic annular images and active deep descriptors. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 920–925, 2019. 2

[4] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018. 4

[5] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, and Jose J Guerrero. Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, 5(2):1255–1262, 2020. 4

[6] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *European Conference on Computer Vision*, pages 369–386, 2020. 4, 5, 7

[7] Toon Goedemé, Marnix Nuttin, Tinne Tuytelaars, and Luc Van Gool. Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236, 2007. 1, 2

[8] Peter Hansen and Brett Browning. Omnidirectional visual place recognition using rotation invariant sequence matching. *Technical Report*, 2015. 2

[9] Jiung-Yao Huang, Su-Hui Lee, and Chung-Hsien Tsai. A fast image matching technique for the panoramic-based localization. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016. 2

[10] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Panorama to panorama matching for location recognition. In *ACM International Conference on Multimedia Retrieval (ICMR)*, 2017. 1, 2

[11] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2011. 2

[12] Peter Karkus, Anelia Angelova, Vincent Vanhoucke, and Riko Jonschkowski. Differentiable mapping networks: Learning structured map representations for sparse visual localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 2

[13] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999. 2

[14] Huimin Lu, Xun Li, Hui Zhang, and Zhiqiang Zheng. Robust place recognition based on omnidirectional vision and real-time local visual features for mobile robots. *Advanced Robotics*, 27(18):1439–1453, 2013. 1, 2

[15] Ana C Murillo, Gautam Singh, Jana Kosecka, and José Jesús Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics*, 29(1):146–160, 2012. 2

[16] Y. Naiming, T. Kanji, F. Yichu, F. Xiaoxiao, I. Kazunori, and I. Yuuki. Long-term vehicle localization using compressed visual experiences. In *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018. 1, 3

[17] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 2

[18] Nathan Piasco, Desire Sidibe, Cedric Demonceaux, and Gouet-Brunet Valerie. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74:90–109, 2018. 1

[19] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5706–5715, 2018. 2

[20] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1655–1668, 2018. 2, 3, 4, 5, 7

[21] A.S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 2016. 3, 4

[22] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6DOF outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 2, 3

[23] Johannes L Schonberger, Filip Radenovic, Ondrej Chum, and Jan-Michael Frahm. From single image query to detailed 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5126–5134, 2015. 4

[24] Georg Schroth, Robert Huitl, David Chen, Mohammad Abu-Alqumsan, Anas Al-Nuaimi, and Eckehard Steinbach. Mobile visual location recognition. *IEEE Signal Processing Magazine*, 28(4):77–89, 2011. 2

[25] E. Stenborg, C. Toft, and L. Hammarstrand. Long-term visual localization using semantically segmented images. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 1

[26] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *European Conference on Computer Vision (ECCV)*, pages 707–722, 2018. 4

[27] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. Semantic match consistency for long-term visual localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 3

[28] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. *International Conference on Learning Representations (ICLR)*, 2016. 2, 3, 4, 5, 7

[29] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015. 1, 2

[30] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:2346–2359, 2015. 1, 2, 5

[31] T.H. Wang, H.J. Huang, J.T. Lin, C.W. Hu, K.H. Zeng, and M. Sun. Omnidirectional CNN for visual place recognition and navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 2

[32] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *European Conference on Computer Vision*, pages 255–268. Springer, 2010. 2

[33] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1546–1558, 2014. 3