IPSJ Transactions on Computer Vision and Applications Vol.6 58–62 (July 2014)

[DOI: 10.2197/ipsjtcva.6.58]

Express Paper

Efficient Localization of Panoramic Images Using Tiled Image Descriptors

Akihiko Torii[1,a)] Yafei Dong[1,b)] Masatoshi Okutomi[1,c)] Josef Sivic[2,d)] Tomas Pajdla[3,e)]

Received: March 14, 2014, Accepted: April 24, 2014, Released: July 25, 2014

Abstract: We seek to localize a query panorama with a wide field of view given a large database of street-level geotagged imagery. This is a challenging task because of significant changes in appearance due to viewpoint, season, occluding people or newly constructed buildings. An additional key challenge is the computational and memory efficiency due to the planet-scale size of the available geotagged image databases. The contributions of this paper are two-fold. First, we develop a compact image representation for scalable retrieval of panoramic images that represents each panorama as an ordered set of vertical image tiles. Two panoramas are matched by efficiently searching for their optimal horizontal alignment, while respecting the tile ordering constraint. Second, we collect a new challenging query test dataset from Shibuya, Tokyo containing more than thousand panoramic and perspective query images with manually verified ground truth geolocation. We demonstrate significant improvements of the proposed method compared to the standard bag-of-visual-words and VLAD baselines.

Keywords: visual place recognition, bag of visual words and VLAD image representations, panorama image localization

1. Introduction

We seek to localize a query image given a large database of street-level geotagged imagery. Solving this problem would have significant practical applications in robotics, augmented reality or navigation. However, this task is difficult as the appearance of the query can be very different from the appearance depicted in the database due to changes in viewpoint, illumination, different season, partial occlusion by objects and people, or even structural changes in the scene such destroyed or newly constructed buildings. In addition, with the emergence of planet scale geolocalized image databases such as Google Street View or Bing maps, one of the key challenges becomes the computational and memory efficiency. What is the appropriate image representation that is compact, efficient, yet sufficiently rich to enable accurate visual localization?

Several successful methods [4], [6], [16], [25], [27], [33] treat the visual localization problem as large-scale instance-level retrieval, where images are represented using local invariant features [19], aggregated into an image-level representation such as the bag-of-visual-words [5], [26] or the VLAD descriptor [13]. The image database can be further augmented by 3D point clouds [15], automatically reconstructed by large-scale structure from motion (SfM) [1], [15], which enables accurate prediction of query image camera position [18], [23].

In this work, we specifically address localization of query images with a large field of view. Having a (partial) panorama query image is not a non-realistic situation because most cameras and smartphones have an easy to use swing panorama function and specialized software is also available [29], [30], [31]. Panoramic cameras are also popular in mobile robotics [6]. Having a large field of view query offers the additional benefit of significantly reducing the ambiguity of place recognition as it captures a bigger portion of the scene.

Yet, most recent large-scale place recognition methods focus on localizing narrow field of view query images. Furthermore, even though the geotagged database imagery usually contains full panoramas, they are usually transformed into a set of overlapping (perspective) cutouts and treated as a collection of unrelated perspective views [4], [7], [16], [27]. In this work we develop a compact yet accurate image representation specifically suited for large scale visual localization of panoramic imagery.

Related work. An early attempt to solve panorama localization was presented in Ref. [21]. The main contribution was the normalization of panoramas removing possibly different panorama rotations due to changing orientations of acquisition devices. Several improvements were later added by Ref. [14] to make the matching based on image correlation robust and efficient. The lack of robustness to illumination changes and acquisition devices, however, remained an issue due to using raw image intensity values.

Recent work has developed [6] scalable methods for localizing panoramic images based on the bag-of-visual-words model followed by geometric verification. We go beyond this work

[1] Department of Mechanical and Control Engineering, Graduate School of Science and Engineering, Tokyo Institute of Technology, Meguro, Tokyo 152–8550, Japan
[2] WILLOW project, Laboratoire d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548, 75214 Paris Cedex 13, France
[3] Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Enginnering, Czech Technical University in Prague, 121–35 Praha 2, Czech Republic
[a)] torii@ctrl.titech.ac.jp
[b)] ydong@ctrl.titech.ac.jp
[c)] mxo@ctrl.titech.ac.jp
[d)] Josef.Sivic@ens.fr
[e)] pajdla@cmp.felk.cvut.cz

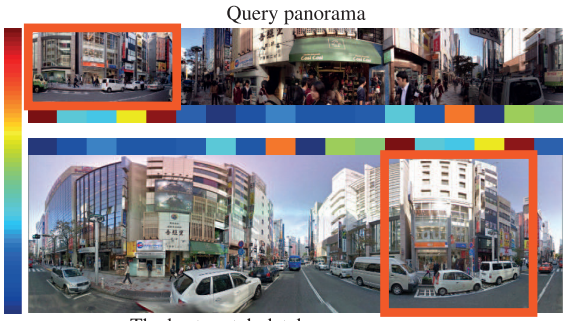


Fig. 1 Matching panoramic images with the circular ordering constraint. Example query panorama and the best (correct) matching database panorama from the Shibuya dataset. The colored horizontal bars indicate the similarity scores between the best matched vertical tile descriptors with the circular ordering constraint (red:high, blue:low). The orange bounding boxes are the best matched image areas. Notice that this is a very challenging image pair to localize due to occlusions and repeated patterns.

and incorporate geometric constraints, specifically developed for matching panoramic imagery, into the indexing stage in a compact and scalable manner.

Our work is also related to the spatial pyramid representation used in category-level recognition [17] and multiple overlapping VLAD descriptors [3] in retrieval, which construct an image descriptor by concatenating blocks of a spatial image pyramid. In contrast, our method exploits stronger geometric constraints, available for panoramic imagery.

Contributions. First, we develop a compact image representation for scalable retrieval of panoramic images that represents each panorama as an ordered set of vertical image tiles, where each tile is represented by a compact visual descriptor. Two panoramas are matched by efficiently searching for their optimal horizontal alignment, while respecting the ordering constraint (**Fig. 1**). Second, we show that this representation can be applied to two different compact visual descriptors: the bag-of-visual-words [22] and VLAD [13]. Third, we collect a new challenging test query dataset from Shibuya, Tokyo containing 366 panorama and 947 perspective images with manually verified ground truth geolocation. Finally, we demonstrate on this data significant improvements in place recognition performance compared to the standard bag-of-visual-words and VLAD baselines.

2. Tiled Image Representation

In this section we first review the bag-of-visual-words (BoVW) representation [5], [26] commonly used for place recognition [4], [7], [16], [27]. Then, we describe how to make the tiled BoVW and discuss the efficiency of storage and computation. Next, we also review the vector of locally aggregated descriptors (VLAD) [13] which is a recent popular compact image descriptor and describe its tiled representation. For simplicity, we assume that the query images are 360° horizontal FoV panoramas but the method is extendable for any FoV with no extra effort.

Tiled bag-of-visual-words representation. In the standard BoVW representation the position of the visual words in the image is lost. In the tiled representation, we split the panorama in a set of vertical tiles and compute a separate histogram of visual words in each tile.

In more detail, we denote features extracted from an image as $D = \{\mathbf{d}_i\}_{i=1}^{N_f}$ and $X = \{\mathbf{x}_i = (x_i, y_i)^\top\}_{i=1}^{N_f}$ where $\mathbf{d}_i$ is the descriptor, $\mathbf{x}_i$ is the keypoint position [32] and N_f is the number of features in the image. The visual words (centroids) pre-computed on the training data are denoted $C = \{\mathbf{c}_i\}_{i=1}^{N_c}$ where N_c is the number of visual words in the vocabulary. The image tiles are constructed as follows. We consider vertical image tile of width τ (in degrees), which results $N_\tau = 360°/\tau$ tiles for a 360 degree panorama. This is implemented by assigning a tile index t_i to each extracted local feature i in the image depending on which tile the feature falls into. We can generate the tiled BoVW for any tile width τ by only changing the tile indices $T = \{t_i\}_{i=1}^{N_f}$ without the need to re-extract the features or to re-assign their descriptors to the visual words.

There are two principal ways of storing the tiled BoVW for images in the database. First, we can store only the visual word indices W and the keypoint positions X, and re-compute the tile histograms on the fly. This does not require any additional disk space since both W and X are stored for the standard BoVW with spatial reranking [22]. This can be further sped-up by pre-computing several sets of tile indices T for some preferred tile widths. Second, we can explicitly pre-compute and store the sparse histograms for all tiles in the panorama. Interestingly, even though the number of histograms increases with the number of tiles N_τ, the total number of non-zero elements, which is the main factor determining the memory complexity, does not increase significantly in practice. This is because for large visual vocabularies most visual words occur only once in the image and hence are assigned typically to only one tile. Similarly, repeated visual words (such as windows on a facade) are typically spatially close [27] and often fall into the same tile.

Tiled VLAD representation. The tiled VLAD is constructed similarly to the tiled BoVW. We prepare the tile indices $T = \{t_i\}_{i=1}^{N_f}$ for each extracted local feature. Then, the tiled VLAD is computed by separately aggregating the quantization residual vectors using the corresponding tile indices T. Similar to the BoVW, the feature-to-centroid assignment and the residual vector computation are required only once for generating the tiled VLAD at many different tile widths and circular sliding angles. However, in contrast to the tiled BoVW, the memory footprint is proportional to the number of tile widths because the VLAD descriptor is a dense vector. To apply PCA compression, the tiled representation has to be explicitly pre-computed at each different tile width in the database. The optimal tile width depends on the type of scenes and the application.

3. Matching Tiled Image Descriptors with a Circular Ordering Constraint

In this section we describe the matching strategy for tiled representations of BoVW and VLAD, respectively, while taking into account the computational efficiency.

We assume the query q and database panoramas d are related by an unknown rotation and wish to recover the best possible rotation between the two. This will be achieved by trying all possible rotations, i.e., circularly "sliding" the query panorama by differ-

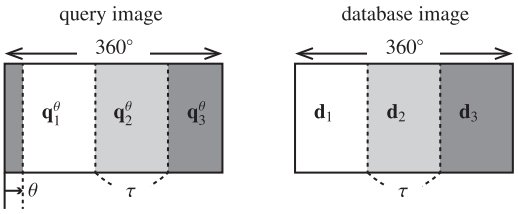


Fig. 2 Tiled representation of image descriptors and matching with the ordering constraint. Given tiled representation of image descriptors, e.g., the tile width 120°, we seek for the best match between the query (left) and database (right) images by searching over the circular sliding angle θ in the query image.

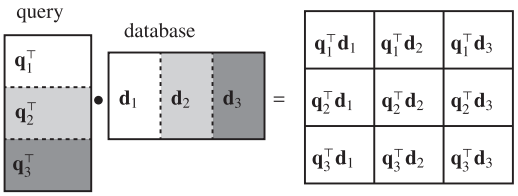


Fig. 3 Matching with the circular ordering constraint for the tiled BoVW. Illustration of matching between a query q and a database image d when the tile width τ and the sliding interval α. Here $\tau = \alpha = 120°$. This is achieved by explicitly enumerating all tile orderings of the query and taking the ordering with the highest matching score.

ent amounts (denoted as θ) as illustrated in **Fig. 2** and taking the rotation with the highest similarity.

Matching tiled BoVW. In the case of matching BoVW, finding the best rotation θ is formulated as

$$\underset{\theta\in(0,360°]}{\operatorname{argmax}} \left(\sum_{i}^{N_\tau} \mathbf{q}_i^{\theta\top} \mathbf{d}_i \right), \tag{1}$$

where $\mathbf{q}_i$ and $\mathbf{d}_i$ are the L2-normalized BoVW vectors of the i-th tile. Explicitly enumerating the different orientations θ considers the matching with all the tile configurations while preserving the ordering of the tiles. The individual per-tile dot products can be computed efficiently using an inverted file indexing structure as in the standard BoVW model. In addition, as the tiled-BoVW vectors are generally much sparser than the non-tiled BoVW vectors computing the per-tile matching score is also very fast. Note that for a given rotation the panoramas are matched tile-by-tile respecting the coarse position of visual words within the panorama and hence encoding some spatial information, similar to spatial pyramids [17].

In detail, the computational overhead of matching with the circular ordering constraint is proportional to the sampling interval α of the circular sliding angle θ. Therefore, it is necessary to find the tile width τ and the interval α to achieve a scalable, robust and accurate retrieval and localization. The naive but effective choice of them is $\tau = \alpha$. **Figure 3** shows an example of computing the similarity scores between query and database tiled-BoVW descriptors for $\tau = \alpha = 120°$. We take the final similarity score of this pair of images as the maximum of the scores $(s_{\theta=0°}, s_{\theta=120°}, s_{\theta=240°})$, where

$$s_{\theta=0°} = \mathbf{q}_1^\top \mathbf{d}_1 + \mathbf{q}_2^\top \mathbf{d}_2 + \mathbf{q}_3^\top \mathbf{d}_3 \tag{2}$$

$$s_{\theta=120°} = \mathbf{q}_2^\top \mathbf{d}_1 + \mathbf{q}_3^\top \mathbf{d}_2 + \mathbf{q}_1^\top \mathbf{d}_3 \tag{3}$$

$$s_{\theta=240°} = \mathbf{q}_3^\top \mathbf{d}_1 + \mathbf{q}_1^\top \mathbf{d}_2 + \mathbf{q}_2^\top \mathbf{d}_3. \tag{4}$$

Matching tiled VLAD. It is possible to conduct the same matching process for the tiled VLAD representation. Although dot products among PCA-compressed tiled-VLAD descriptors can be efficiently computed using multi-core parallelization, we cannot use the inverted indexing due to the non-sparseness of the VLAD. Instead of computing the similarity between the descriptors by dot products, we can compute for each orientation θ of the query the sum of distances between individual tile descriptors

$$\underset{\theta\in(0,360°]}{\operatorname{argmin}} \sum_{i}^{N_\tau} \| \mathbf{q}_i^\theta - \mathbf{d}_i \|. \tag{5}$$

Efficient matching can be achieved by using approximate nearest neighbor search [20], [22] or product quantization [12].

4. Experiments

In this section we describe the experimental validation of our approach on the Pittsburgh and Shibuya datasets. We first describe the experimental set-up of the two datasets and give the implementation details. Then, we show and discuss the results.

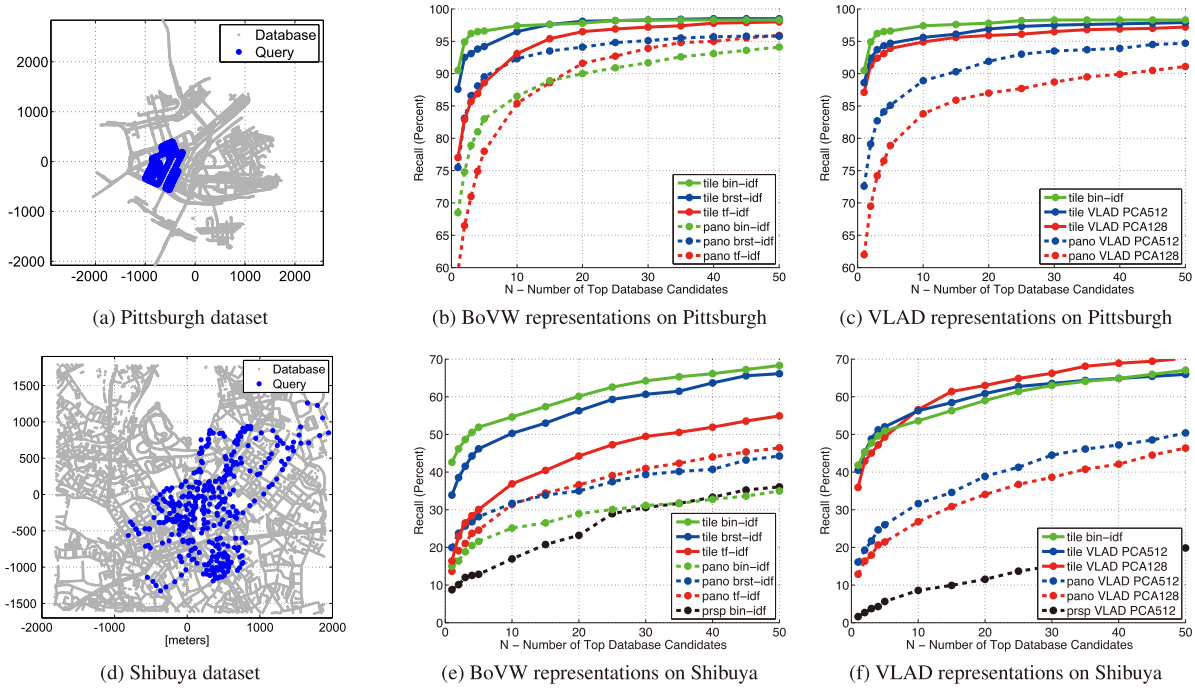
Pittsburgh dataset. The geotagged image database is formed by 10,586 Google Street View panoramas of the Pittsburgh area downloaded from the Internet. As testing query images, we use 1,000 panoramas randomly selected from 8,999 panoramas of the Google Pittsburgh Research Data Set [*1] (**Fig. 4** (a)). From each panorama of 3,328×1,664 pixels, we cropped the bottom 572 pixels in order to remove the blending artifacts present in the street view images. The ground truth is derived from the GPS positions of the query images similarly to Ref. [27].

Shibuya dataset. We collected a new Shibuya dataset by downloading 24,701 Google Street View panoramas of the Shibuya area in Tokyo (Fig. 4 (d)). Each panorama has a resolution of 3,328 × 1,664 pixels and is cropped in the same manner as the Pittsburgh dataset.

As testing query images, we took wide angle view images using cameras on five different smartphones: iphone4, iphone5, iphone5s, LG-Android, and NEC-Android. We captured a total of 366 panoramic query images using three different applications running on the device [29], [30], [31]. All panoramas have 360° horizontal FoV. To investigate the benefit of using a large FoV images, we also took several perspective images at the same location of each panorama image. This results in total of 947 perspective images and we randomly choose one perspective image for each panorama location resulting in a test set of 366 perspective views. We resized the perspective images to have the maximum of the height/width 640 pixels to obtain roughly the same resolution as the Google Street View panoramas, assuming that FoV is 60°. This dataset is available on request.

Implementation details. We build a visual vocabulary of 200,000 visual words by approximate k-means clustering [20], [22] for BoVW and 256 centroids by k-means for VLAD. For each dataset, the vocabulary is built from features detected in a subset of randomly selected database panorama images. We use the SIFT descriptors with the upright image gravity vector followed by the RootSIFT normalization [2]. If the

[*1] Provided and copyrighted by Google.



(a) Pittsburgh dataset (b) BoVW representations on Pittsburgh (c) VLAD representations on Pittsburgh

(d) Shibuya dataset (e) BoVW representations on Shibuya (f) VLAD representations on Shibuya

Fig. 4 Comparison with baselines. Locations of query (blue dots) and database (gray dots) images (a) and (d). The fraction of correctly localized queries (Recall, y-axis) vs. the number of top N retrieved database images (x-axis) for the proposed methods ("tile") compared to their corresponding baselines ("pano") for BoVW (b), (e) and VLAD (c), (f) representations.

query panorama followed a consistent projection model, adopting spherical SIFT [8] would have been also a reasonable choice. We keep the standard (upright) SIFT because our query panoramas are generated by different types of projections.

Comparison with baseline BoVW methods. We compare the proposed method with several baselines: the standard tf-idf weighting (pano tf-idf) [22], burstiness weights (pano brst-idf) [10], and a binarization (pano bin-idf) [11]. For the proposed method, the tiled BoVW with an ordering constraint, we set the vertical image tile width $\tau = 20°$ and the circular sliding interval $\alpha = 20°$, and applied the same weighting scheme as the baselines (tile tf-idf, tile brst-idf, and tile bin-idf).

For each method, we measure the percentage of correctly localized queries (Recall) similarly to Refs. [4], [24]. 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.

Figure 4 (b) shows the results of different methods on the Pittsburgh dataset for $m = 25$ meters while varying the value of N. Figure 4 (e) shows the results on the Shibuya dataset for $m = 50$ meters while varying the value of N. The distance threshold m is relaxed for the Shibuya dataset because of the typical GPS distance between the query and the corresponding database image is larger compared to the Pittsburgh dataset where both query and database images are captured in the middle of the street. All the tiling representations consistently outperform their corresponding baselines. Notice that the improvement at the top 1 recall is more than 20%.

Comparison with baseline VLAD methods. Figure 4 (c) and Fig. 4 (f) show the same experiments but for the standard VLAD descriptor including PCA compression (VLAD PCA128, VLAD PCA512) [9]. We examined even lower PCA compression rates but did not see significant improvements in accuracy. Our tiled VLAD with an ordering constraint (tile VLAD) gives significant improvements compared to the baselines (pano VLAD).

Table 1 Memory footprint of the Pittsburgh database (MB).

tile width (degree)	360	60	20	10
tiled BoVW	466	477	481	484
tiled VLAD PCA 128	5.4	33	98	195
tiled VLAD PCA 512	22	130	390	780

Scalability. Table 1 shows the size of database represented by the tiled BoVW and VLAD. For the BoVW, we count the memory complexity by 8 bytes (4 bytes for index and 4 bytes for weight) per visual word entry. For the VLAD, we count 4 bytes (single precision) per dimension. The tiled VLAD PCA 128 at the tile width 20° is compact enough to be loadable on the recent mobile devices and yet gives good performance as shown in Fig. 4.

Impact of FoV. On the Shibuya dataset, we compare the performance with 366 perspective query images in order to understand the benefits of large FoV. The recall obtained by the perspective queries using the bin-idf and VLAD PCA 512 representation is shown (black dashed curves "prsp") in Fig. 4 (e) and (f), respectively. We can clearly see that place recognition with narrow FoV is extremely difficult on this real challenging dataset.

Qualitative examples. Figures 5 and **6** show example matches. Each figure shows the query image (top), the best matching database image (middle) correctly matched by the proposed method (tile bin-idf), and the best matching image (bottom) incorrectly matched by the baseline method (pano bin-idf). The colored horizontal bars indicate the similarity score between the matched tile descriptors for the query and the best matching

Fig. 5 Matching example of the query image captured by PhotoSynth on the Shibuya dataset. (Top) the query image. (Middle) the best matching database image "correct" by the proposed method (tiled bin-idf). (Bottom) the best matching image "incorrect" by the baseline method (pano bin-idf). See text for detailed explanation of the colored bars.

Fig. 6 Matching example of the query image captured by Dermander on the Shibuya dataset. See the caption of Fig. 5 for details.

database image using the circular ordering constraint. The colors correspond to the Matlab "jet" color map, i.e., red indicates high similarity and blue indicates low similarity. The similarity scores are normalized independently for each pair of images such that the maximal similarity is equal to one for visibility.

5. Conclusion

We have demonstrated localizing query panoramas with a large field of view given a large database of street-level geotagged imagery. We have shown that panoramic image representation by vertical image tiles together with a tile ordering constraint significantly improves place recognition performance on a real challenging dataset. As our planet is covered by panoramic images [28] efficient visual matching to this imagery is of significant practical importance.

Acknowledgments Supported by JSPS KAKENHI Grant Number 24700161, the EIT ICT Labs, ANR project Semapolis and Google, and FP7-SPACE-2012-312377 PRoViDE.

References

[1] Agarwal, S., Snavely, N., Simon, I., Seitz, S. and Szeliski, R.: Building rome in a day, *ICCV*, pp.72–79 (2009).
[2] Arandjelović, R. and Zisserman, A.: Three things everyone should know to improve object retrieval, *CVPR*, pp.2911–2918 (2012).
[3] Arandjelović, R. and Zisserman, A.: All about VLAD, *CVPR* (2013).
[4] Chen, D., Baatz, G., et al.: City-scale landmark identification on mobile devices, *CVPR*, pp.737–744 (2011).
[5] Csurka, G., Bray, C., Dance, C. and Fan, L.: Visual categorization with bags of keypoints, *WS-SLCV, ECCV* (2004).
[6] Cummins, M. and Newman, P.: Highly scalable appearance-only SLAM - FAB-MAP 2.0, *Proc. Robotics: Science and Systems*, Seattle, USA (June 2009).
[7] Gronat, P., Obozinski, G., Sivic, J. and Pajdla, T.: Learning and calibrating per-location classifiers for visual place recognition, *CVPR* (2013).
[8] Hansen, P., Corke, P., Boles, W. and Daniilidis, K.: Scale-invariant features on the sphere, *ICCV* (2007).
[9] Jégou, H. and Chum, O.: Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening, *ECCV*, Vol.2, pp.774–787 (2012).
[10] Jégou, H., Douze, M. and Schmid, C.: On the burstiness of visual elements, *CVPR*, pp.1169–1176 (2009).
[11] Jégou, H., Douze, M. and Schmid, C.: Packing bag-of-features, *ICCV*, pp.2357–2364 (2009).
[12] Jégou, H., Douze, M. and Schmid, C.: Product quantization for nearest neighbor search, *PAMI*, Vol.33, No.1, pp.117–128 (2011).
[13] Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P. and Schmid, C.: Aggregating local image descriptors into compact codes, *PAMI*, Vol.34, No.9, pp.1704–1716 (2012).
[14] Jogan, M. and Leonardis, A.: Robust localization using an omnidirectional appearance-based subspace model of environment, *Robotics and Autonomous Systems*, pp.51–72 (2003).
[15] Klingner, B., Martin, D. and Roseborough, J.: Street view motion-from-structure-from-motion, *ICCV* (2013).
[16] Knopp, J., Sivic, J. and Pajdla, T.: Avoiding confusing features in place recognition, *ECCV*, Vol.1, pp.748–761 (2010).
[17] Lazebnik, S., Schmid, C. and Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, *CVPR*, Vol.2, pp.2169–2178 (2006).
[18] Li, Y., Snavely, N., Huttenlocher, D. and Fua, P.: Worldwide pose estimation using 3D point clouds, *ECCV*, pp.15–29, Berlin, Heidelberg, Springer-Verlag (2012).
[19] Lowe, D.: Distinctive image features from scale-invariant keypoints, *IJCV*, Vol.60, No.2, pp.91–110 (2004).
[20] Muja, M. and Lowe, D.: Fast approximate nearest neighbors with automatic algorithm configuration, *VISAPP*, Vol.1, pp.331–340 (2009).
[21] Pajdla, T. and Hlavac, V.: Zero phase representation of panoramic images for image based localization, *CAIP*, pp.550–557 (1999).
[22] Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching, *CVPR* (2007).
[23] Sattler, T., Leibe, B. and Kobbelt, L.: Improving image-based localization by active correspondence search, *ECCV*, Vol.1, pp.752–765 (2012).
[24] Sattler, T., Weyand, T., Leibe, B. and Kobbelt, L.: Image retrieval for image-based localization revisited, *BMVC* (2012).
[25] Schindler, G., Brown, M. and Szeliski, R.: City-scale location recognition, *CVPR* (2007).
[26] Sivic, J. and Zisserman, A.: Video Google: A text retrieval approach to object matching in videos, *ICCV*, pp.1470–1477 (2003).
[27] Torii, A., Sivic, J., Pajdla, T. and Okutomi, M.: Visual place recognition with repetitive structures, *CVPR*, pp.883–890 (2013).
[28] Google Maps: available from ⟨http://maps.google.com/help/maps/streetview/⟩.
[29] Photosynth: available from ⟨http://photosynth.net/mobile.aspx⟩.
[30] Dermandar: available from ⟨http://www.dermandar.com/⟩.
[31] Photaf: available from ⟨http://www.photaf.com/⟩.
[32] Vedaldi, A. and Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), available from ⟨http://www.vlfeat.org/⟩.
[33] Zamir, A. and Shah, M.: Accurate image localization based on google maps street view, *ECCV*, Vol.4, pp.255–268 (2010).

(Communicated by *Keiji Yanai*)