








# LandscapeAR: Large Scale Outdoor Augmented Reality by Matching Photographs with Terrain Models Using Learned Descriptors

Jan Brejcha<sup>1,2</sup> , Michal Lukáč<sup>2</sup> , Yannick Hold-Geoffroy<sup>2</sup> ,  
Oliver Wang<sup>2</sup> , and Martin Čadík<sup>1</sup> 

<sup>1</sup> Faculty of Information Technology, CPhoto@FIT, Brno University of Technology,  
Božetěchova 2, 61200 Brno, Czech Republic  
[ibrejcha@fit.vutbr.cz](mailto:ibrejcha@fit.vutbr.cz)

<http://cphoto.fit.vutbr.cz>

<sup>2</sup> Adobe Inc., 345 Park Ave, San Jose, CA 95110-2704, USA

**Abstract.** We introduce a solution to *large scale* Augmented Reality for outdoor scenes by registering camera images to textured Digital Elevation Models (DEMs). To accommodate the inherent differences in appearance between real images and DEMs, we train a cross-domain feature descriptor using Structure From Motion (SFM) guided reconstructions to acquire training data. Our method runs efficiently on a mobile device and outperforms existing learned and hand-designed feature descriptors for this task.

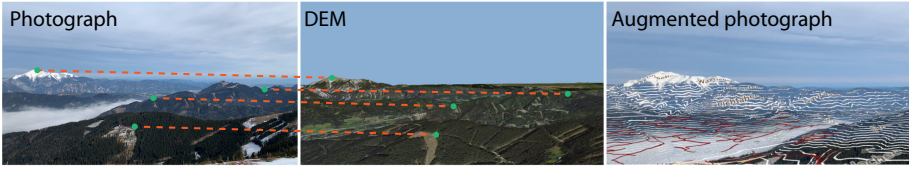
## 1 Introduction

Augmented reality systems rely on some approximate knowledge of physical geometry to facilitate the interaction of virtual objects with the physical scene, and tracking of the camera pose in order to render the virtual content correctly. In practice, a suitable scene is tracked with the help of active depth sensors, stereo cameras, or multiview geometry from monocular video (e.g. SLAM). All of these approaches are limited in their *operational range*, due to constraints related to light falloff for active illumination, and stereo baselines and camera parallax for multiview methods.

In this work, we propose a solution for outdoor *landscape-scale* augmented reality applications by registering the user's camera feed to large scale textured Digital Elevation Models (DEMs). As there is significant appearance variation between the DEM and the camera feed, we train a data driven cross-domain feature descriptor that allows us to perform efficient and accurate feature matching.

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-58526-6\\_18](https://doi.org/10.1007/978-3-030-58526-6_18)) contains supplementary material, which is available to authorized users.



**Fig. 1.** Our method matches a query photograph to a rendered digital elevation model (DEM). For clarity, we visualize only four matches (dashed orange). The matches produced by our system can then be used for localization, which is a key component for augmented reality applications. In the right image (zoomed-in for clarity), we render contour lines (white), gravel roads (red), and trails (black) using the estimated camera pose. (Color figure online)

Using this approach, we are able to localize photos based on long-distance cues, allowing us to display large scale augmented reality overlays such as altitude contour lines, map features (roads and trails), or 3D created content, such as educational geographic-focused features. We can also augment long-distance scene content in images with DEM derived features, such as semantic segmentation labels, depth values, and normals.

Since modern mobile devices as well as many cameras come with built-in GPS, compass and accelerometer, we could attempt to compute alignment from this data. Unfortunately, all of these sensors are subject to various sources of imprecision; e.g., the compass suffers from magnetic variation (irregularities of the terrestrial magnetic field) as well as deviation (unpredictable irregularities caused by deposits of ferrous minerals, or even by random small metal objects around the sensor itself). This means that while the computed alignment is usually close enough for rough localization, the accumulated error over geographical distances results in visible mismatches in places such as the horizon line.

The key insight of our approach is that we can take advantage of a robust and readily available source of data, with near-global coverage, that is DEM models, in order to compute camera location using reliable, 3D feature matching based methods. However, registering photographs to DEMs is challenging, as both domains are substantially different. For example, even high-quality DEMs tend to have resolution too rough to capture local high-frequency features like mountain peaks, leading to horizon mismatches. In addition, photographs have (often) unknown camera intrinsics such as focal length, exhibit seasonal and weather variations, foreground occluders like trees or people, and objects not present in the DEM itself, like buildings.

Our method works by learning a data-driven cross-domain feature embedding. We first use Structure From Motion (SFM) to reconstruct a robust 3D model from internet photographs, aligning it to a known terrain model. We then render views at similar poses as photographs, which lets us extract cross-domain patches in correspondence, which we use as supervision for training. At test time, no 3D reconstruction is needed, and features from the query image can be matched directly to renderings of the DEM.

Registration to DEMs only makes sense for images that observe a significant amount of content *farther* away than ca 100 m. For this reason, we focus on mountainous regions, where distant terrain is often visible. While buildings would also provide a reasonable source for registration, in this work we do not test on buildings, as building geometry is diverse, and 3D data and textures for urban areas are not freely available.

Our method is efficient and runs on a mobile device. As a demonstration, we developed a mobile application that performs large-scale visual localization to landscape features locally on a recent iPhone, and show that our approach can be used to refine localization when embedded device sensors are inaccurate.

In summary, we present the following contributions:

- A novel data-driven cross-domain embedding technique suitable for computing similarity between patches from photographs and a textured terrain model.
- A novel approach to Structure-from-Motion using terrain reference to align internet photographs with the terrain model (using D2Net detector & descriptor). Using our technique, a dataset of 16k images has been built and was used for training our method; it is by far the largest dataset of single image precise camera poses in mountainous regions. The dataset and source is available on our project website<sup>1</sup>.
- A novel weakly supervised training scheme for positive/negative patch generation from the SfM reconstruction aligned with a DEM.
- We show that our novel embedding can be used for matching photographs to the terrain model to estimate respective camera position and orientation.
- We implement our system on the iPhone, showing that mobile large scale localization is possible on-device.

## 2 Related Work

### 2.1 Visual Localization

Localizing cameras in a 3D world is a fundamental component of computer vision and is used in a wide variety of applications. Classic solutions involve computing absolute pose between camera images and a known set of 3D points, typically solving the Perspective- $n$ -Point [14] algorithm, or computing relative pose between two cameras observing the same scene, which can be computed solving the 5-point problem [28]. These approaches are founded in 3D projective geometry and can yield very accurate results when dealing with reliable correspondence measurements.

Recently, deep learning has been proposed as a solution to directly try to predict the camera location from scene observations using a forward pass through a CNN [21, 38]. However, recent analysis has shown that these methods operate by image retrieval, computing the pose based on similarity to known images, and

---

<sup>1</sup> <http://cphoto.fit.vutbr.cz/LandscapeAR/>.

still do not exceed those from classic approaches to this problem [31]. Additionally, such approaches require the whole scene geometry to be represented within the network weights, and can only work on scenes that were seen during training. Our method leverages 3D geometric assumptions external to the model, making it more generalizable and accurate.

Existing approaches to outdoor camera orientation assessment [4, 8, 27], on the other hand, require a precise camera position. Accordingly, these works are insufficient in our scenario where the camera location is often inaccurate.

## 2.2 Local Descriptors

A key part of camera localization is correspondence finding. Most classical solutions to this problem involve using descriptors computed from local windows around feature points. These descriptors can be either hand-designed, e.g., SIFT [25], SURF [6], ORB [30], or learned end-to-end [12, 15, 26, 35, 39]. While our method is also a local descriptor, it is designed to deal with additional appearance and geometry differences, which is not the case for these methods.

Of these, HardNet++ [26] and D2Net [12] have been trained on outdoor images (HardNet on Brown dataset and HPatches, D2Net on Megadepth which contains 3D reconstructed models in the European Alps and Yosemite). Since it is possible that a powerful enough single-domain method might be able to bridge the domain gap (as demonstrated for D2Net and sketches), and these two methods are compatible with our use-case, we chose them as baselines to compare with our method.

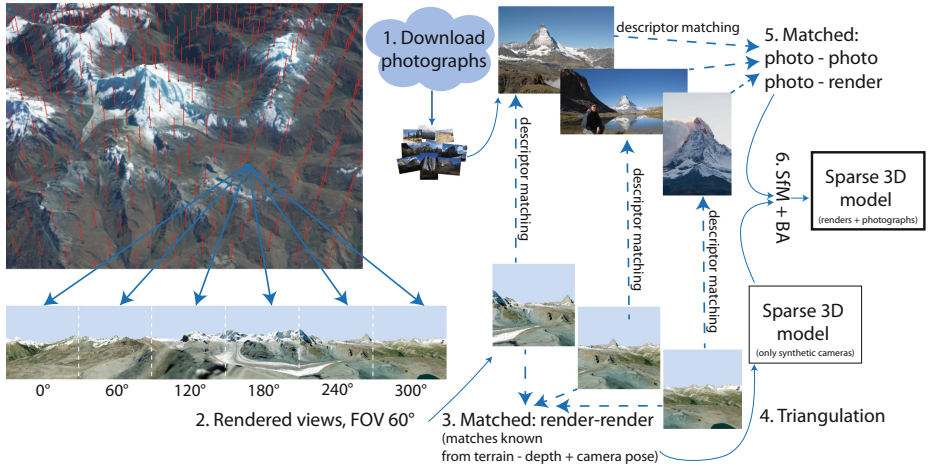
## 2.3 Cross-Domain Matching

A large body of research work has been devoted to alignment of multi-sensor images [19, 20, 36] and to modality-invariant descriptors [11, 18, 23, 32, 33]. These efforts often focus on optical image alignment with e.g., its infra-red counterpart. However, our scenario is much more challenging, because we are matching an image with a *rendered* DEM where the change in appearance is considerable.

With the advent of deep-learning, several CNN-based works on matching multimodal patches emerged and outperformed previous multimodal descriptors [1, 2, 5, 13, 16]. However, cross-spectral approaches [1, 2, 5, 13] need to account only for rapid visual appearance change, compared to our scenario, which needs to cover also the differences in scene geometry, caused by limited DEM resolution. On the other hand, RGB to depth matching approaches, such as Georgakis et al. [16] lack the texture information and need to focus only on geometry, which is not our case.

## 3 Method

Our goal is to estimate the camera pose of a query image with respect to the synthetic globe, which can be cast as a standard Perspective- $n$ -Point problem [14]



**Fig. 2. Structure-from-motion with a terrain reference for automatic cross-domain dataset generation.** In the area of interest, camera positions are sampled on a regular grid (red markers). At each position, 6 views covering the full panorama are rendered. A sparse 3D model is created from the synthetic data using known camera poses and scene geometry. Each photograph is localized to the synthetic sparse 3D model. Image credit, photographs left to right: John Bohlmeier (<https://flic.kr/p/gm3zRQ>), Tony Tsang (<https://flic.kr/p/gWmPbU>), distantranges (<https://flic.kr/p/gJCPui>). (Color figure online)

given accurate correspondences. The main challenge is therefore, to establish correspondences between keypoints in the query photograph and a rendered synthetic frame. We bridge this appearance gap by training an embedding function which projects local neighborhoods of keypoints from either domain into a unified descriptor space.

### 3.1 Dataset Generation

The central difficulty of training a robust cross-domain embedding function is obtaining accurately aligned pairs of photographs and DEM renders. Manually annotating camera poses is tedious and prone to errors, and capturing diverse enough data with accurate pose information is challenging. Instead, we use internet photo collections, which are highly diverse, but contain unreliable location annotations. For each training photograph, we therefore need to retrieve precise camera pose  $P = K[R|t]$ , which defines the camera translation  $t$ , rotation  $R$ , and intrinsic parameters  $K$  with respect to the reference frame of the virtual globe.

In previous work [9, 37], Structure-from-Motion (SfM) techniques have been used in a two-step process to align the photographs into the terrain. These methods reconstruct a sparse 3D model from photographs and then align it to the terrain model using point cloud alignment methods, such as Iterative Closest

Points. However, significant appearance variation and relatively low density of outdoor photographs makes photo-to-photo matching difficult, leading to reconstruction which is highly unstable, imprecise, and prone to drift. In many areas, coverage density is too low for the method to work at all.

Instead, we propose a registration step where photographs are aligned via a DEM-guided SfM step, in which the known camera parameters and geometry of the DEM domain help overcome ambiguous matches and lack of data in the photo domain. As input, we download photographs within a given rectangle of  $10 \times 10$  km from an online service (Fig. 2-1), such as Flickr.com. For the same area, we also render panoramic images sampled 1 km apart on a regular grid (Fig. 2-2). For each sampled position, we render 6 images with  $60^\circ$  field-of-view each rotated  $60^\circ$  around the vertical axis, where for each rendered image, we store a depth map, full camera pose and detected keypoints and descriptors using a baseline feature descriptor D2Net [12]. For rendered images, we calculate matches directly from the terrain geometry using the stored camera poses and depth maps – no descriptor matching between rendered images is needed (Fig. 2-3). We obtain an initial sparse 3D model directly from the synthetic data (Fig. 2-4).

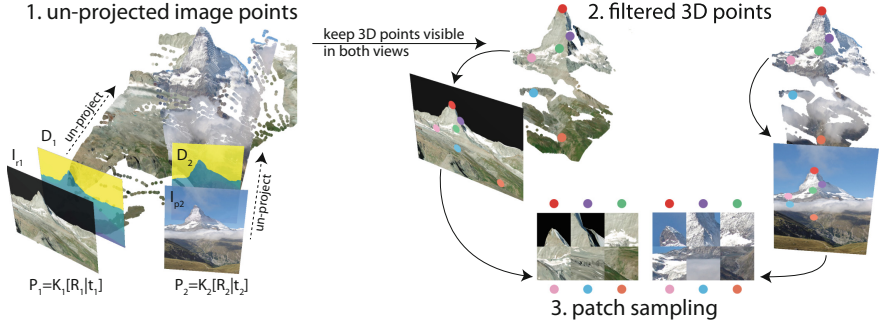
In the next step, we extract keypoints and descriptors from the input photographs using D2Net. The input photographs are matched to every other photograph *and* to rendered images using descriptor matching (Fig. 2-5), and localized to the terrain model using Structure-from-Motion (Fig. 2-6). Global bundle adjustment is used to refine camera parameters belonging to photographs and 3D points, while the rendered cameras have fixed all parameters, since they are known precisely.

Importantly, while existing single-domain feature descriptors are not robust to the photo-DEM domain gap, we can overcome this limitation by sheer volume of synthetic data. Most of the matches will be within the same domain (e.g., photo to photo), and only a small handful need to successfully match to DEM images for the entire photo domain model to be accurately registered. This procedure relies on having a collection of photos from diverse views and extensive processing, therefore doing so at inference time would be prohibitive. However, we can use this technique to build a dataset for training, after which our learned descriptor can be used to efficiently register a *single* photograph.

Finally, we check the location for each reconstructed photograph from the terrain model and prune photographs that are located below, or more than 100 m above the terrain since they are unlikely to be localized precisely. This approach proved to be much more robust and drift-free, and was able to georegister photographs in every area we tested. To illustrate this, we reconstructed 6 areas across the European Alps region, and 1 area in South American Andes. In total, we localized 16,611 photographs using this approach.

### 3.2 Weakly Supervised Cross-domain Patch Sampling

While the rendered image is assumed to contain a similar view as the photograph, it is not exact. Therefore, our embedding function should be robust to



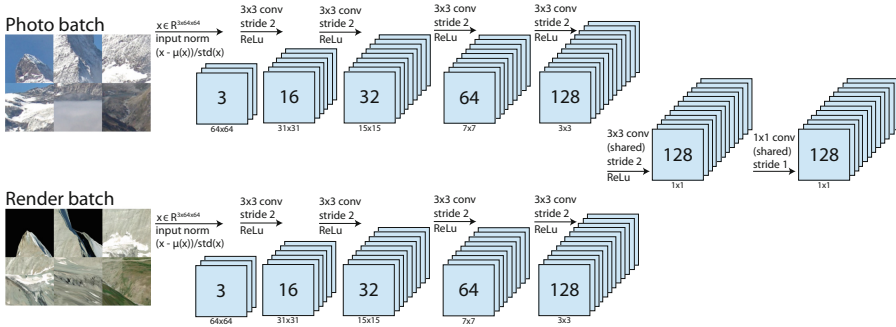
**Fig. 3.** 1. For a pair of images  $I_{r1}$  (render),  $I_{p2}$  (photograph), 2D image points are un-projected into 3D using the rendered depth maps  $D_1$ ,  $D_2$ , and the ground truth camera poses  $P_1$ ,  $P_2$ , respectively. 2. Only points visible from both views are kept. 3. A randomly selected subset of 3D points is used to form patch centers, and corresponding patches are extracted. Image credit: John Bohlmeyer (<https://flic.kr/p/gm3xwP>).

slight geometric deformations caused by viewpoint change, weather and seasonal changes, and different illumination. Note that these phenomena do not occur only in the photograph, but also in the ortho-photo textures. Previous work on wide baseline stereo matching, patch verification and instance retrieval illustrate that these properties could be learned directly from data [3, 12, 26, 29]. For efficient training process, an automatic selection of corresponding (positive) and negative examples is crucial. In contrast with other methods, which rely on the reconstructed 3D points [12, 26] dependent on a keypoint detector, we instead propose a weakly supervised patch sampling method completely independent of a preexisting keypoint detector to avoid any bias that might incur. This is an important and desirable property for our cross-domain approach, since (I) the accuracy of existing keypoint detectors in the cross domain matching task is unknown, (II) our embedding function may be used with any keypoint detector in the future without the need for re-training.

Each photograph in our dataset contains ground truth camera pose  $P = K[R|t]$  transforming the synthetic world coordinates into the camera space. For each photograph  $I_{p1}$ , we render a synthetic image  $I_{r1}$  and a depth map  $D_1$ , see Fig. 3. We pick all pairs of cameras which have at least 30 corresponding 3D points in the SfM reconstruction described in Sect. 3.1. For each pair, the camera pose and depth map are used to un-project all image pixels into a dense 3D model (Fig. 3-1). Next, for each domain, we keep only the 3D points visible in both views (Fig. 3-2). Finally, we uniformly sample  $N$  random correspondences (Fig. 3-3), each defining the center of a local image patch.

### 3.3 Architecture

In order to account for the appearance gap between our domains, we employ a branched network with one branch for each of the input domains followed



**Fig. 4.** Architecture of our two branch network with partially shared weights for cross-domain descriptor extraction. Photo and render branches contain four  $3 \times 3$  2D convolutions with stride 2; weights are not shared between branches. The last two convolutions form a trunk of the network with shared weights to embed both domains into a single space. Output is 128-d descriptor. Either one or the other branch is used, each branch is specific for its own domain. Image credit: John Bohlmeyer (<https://flic.kr/p/gm3xwP>).

by a shared trunk. A description of the architecture is shown in Fig. 4. The proposed architecture is fully convolutional and has a receptive field of 63 px. To get a single descriptor, we use an input patch of size  $64 \times 64$  px. We use neither pooling nor batch normalization layers. Similarly to HardNet [26], we normalize each input patch by subtracting its mean and dividing by its standard deviation. Thanks to the structure of our task formulation and the simplicity of the chosen architecture, our network is quite compact and contains only 261,536 trainable parameters, compared to VGG-16 [34] used by D2Net [12] which contains more than 7.6 million of trainable parameters. The small size allows our architecture to be easily deployed to a mobile device like the iPhone, enabling a wider scale of applications.

### 3.4 Training

We use a standard triplet loss function adjusted to our cross-domain scenario:

$$L(a^h, p^r, n^r) = \sum_i \max(\|f^h(a_i^h) - f^r(p_i^r)\|_2 - \|f^h(a_i^h) - f^r(n_i^r)\|_2 + \alpha, 0), \quad (1)$$

where  $a$ ,  $p$ ,  $n$  denotes a mini-batch of anchor, positive, and negative patches, respectively, superscript denotes photograph ( $h$ ), or render ( $r$ ),  $f^h$  and  $f^r$  denotes our embedding functions for *photograph* and *render* branches respectively, and  $\alpha$  denotes the margin.

Previous work on descriptor learning using the triplet loss function [26] illustrated the importance of sampling strategy for selecting negative examples. In this solution, for each patch in a mini-batch, we know its 3D coordinate in an



euclidean world space  $x(p_j) \in \mathbf{R}^3$ . Given a mini-batch of anchor and positive descriptors  $f^h(a_i^h), f^r(p_i^r), i \in [0, N]$  where  $N$  is a batch size, we first select subset of *possible* negatives  $n^r$  from all positive samples within a current batch, which are farther than  $m$  meters from the anchor:  $n^r = \{p_j^r | (\|x(p_j^r) - x(a_i^h)\|_2) > m\}$ . In HardNet [26], for each positive only a hardest negative from the subset of possible negatives should be selected. However, we found that this strategy led the embedding function to collapse into a singular point. Therefore, we propose an adaptive variant of hard negative sampling inspired by a prior off-line mining strategy [17], modified to operate on-line.

We introduce a curriculum to increase the difficulty of the randomly sampled negatives during training. In classic hard negative mining, for each anchor descriptor  $a_i$  we randomly choose descriptor  $p_j$  as a negative example  $n_j$ , if and only if the triplet loss criterion is violated:

$$\|a_i - p_j\|_2 < \|a_i - p_i\|_2 + \alpha, \quad (2)$$

where we denote  $a_i = f^h(a_i^h)$  as an anchor descriptor calculated from a photo patch using the photo encoder, and similarly for  $p_j = f^r(p_j^r)$ , and  $p_i = f^r(p_i^r)$ . We build on this, and for each anchor descriptor  $a_i$ , randomly choose a descriptor  $p_j$  as a negative example  $n_j$  iff:

$$\|a_i - p_j\|_2 < d^+ - (d^+ - (n_{\min} + \epsilon)) \cdot \lambda, \quad (3)$$

where  $\lambda$  is a parameter in  $[0, 1]$  defining the difficulty of the negative mining,  $\epsilon \rightarrow 0^+$  is a small positive constant,  $d^+$  is the distance between anchor and positive plus margin:  $d^+ = \|a_i - p_i\|_2 + \alpha$ , and  $n_{\min}$  is the distance between the anchor and the hardest negative:  $n_{\min} = \min_{p_j} \|a_i - p_j\|_2$ . Intuitively, when  $\lambda = 0$ , Eq. 3 is reduced to random hard negative sampling defined in Eq. 2, and when  $\lambda = 1$ , the Eq. 3 is forced to select  $p_j$  as a negative only if it is equal to the hardest negative  $n_{\min}$ , reducing the sampling method to HardNet [26]. Thus,  $\lambda$  allows us to select harder negatives throughout the training. For details, please see the supplementary material.

So far, we defined our loss function to be a cross-domain triplet loss, having an anchor as a *photograph*, and the positive and negative patches as *renders*. However, this loss function optimizes only the distance between the *photograph* and *render* descriptors. As a result, we use a variant with auxiliary loss functions optimizing also the distances between *photo-photo* and *render-render* descriptors:

$$L_{\text{aux}} = L(a^h, p^r, n^r) + L(a^h, p^h, n^h) + L(a^r, p^r, n^r). \quad (4)$$

As we illustrate by our experiments, this variant performs the best in the cross-domain matching scenario.

### 3.5 Pose Estimation

We illustrate the performance of our descriptor on a camera pose estimation task from a single query image. For each query image, we render a fan of 12 images from the initial position estimate (using GPS in our application and using ground

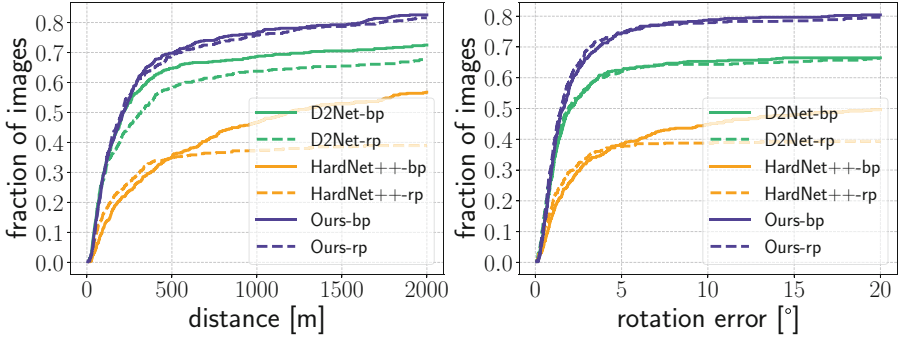
truth position in our experiments) with  $\text{FOV} = 60^\circ$  rotated by  $30^\circ$  around the vertical axis, similarly to Fig. 2-2. The input photograph is scaled by a factor  $s$  proportional to its FOV  $f$ :  $s = (f \cdot M) / (\pi \cdot I_w)$ , where  $M$  is the maximum resolution corresponding to  $\text{FOV} = 180^\circ$  and  $I_w$  is the width of the image. We use the SIFT keypoint detector (although any detector could be used), take a  $64 \times 64$  px patch around each keypoint, and calculate a descriptor using our method.

We start by finding the top candidates from the rendered fan using a simple voting strategy: for each rendered image we calculate the number of mutual nearest neighbor matches with the input photograph. We use the top-3 candidates, since the photograph is unlikely to span more than three consecutive renders, covering a FOV of  $120^\circ$ . For each top candidate, we un-project the 2D points from the rendered image to 3D using rendered camera parameters and a depth map; then we compute full camera pose of the photograph with respect to the 3D coordinates using OpenCV implementation of EPnP [24] algorithm with RANSAC. From the three output camera poses, we select the *best pose* which minimizes the reprojection error while having reasonable number of inliers; if any candidate poses have more than  $N = 60$  inliers, we select the one with the lowest reprojection error. If none are found, we lower the threshold  $N$  and check for the *best pose* in a new iteration. If there is no candidate pose with at least  $N = 20$  inliers, we end the algorithm as unsuccessful. Finally, we reproject all the matches – not only inliers – into the camera plane using the *best pose*, and select those that are within frame. We repeat the matching proces and EPnP to obtain the *refined pose*.

## 4 Experiments

We present majority of the results as cumulative error plots, where we count the fraction of images localized below some distance or rotation error threshold. An ideal system is located at the top-left corner, where all the images are localized with zero distance and rotation errors. Throughout the experiments section, we denote our architecture and its variants trained on our training dataset as **Ours-\***. In addition, we report results for a larger single-branch architecture based on VGG-16 fine-tuned on our data (denoted as **VGG-16-D2-FT**). Similarly as D2Net, we cut the VGG-16 at conv 4-3, load the D2Net weights, and add two more convolutional layers to subsample the result descriptor to 128 dimensions. The newly added layers as well as the conv 4-3 were fine-tuned using our training method and data.

Our methods are compared with state-of-the-art deep local descriptors or matchers: HardNet++ [26], D2Net [12] and NCNet [29], which we use with original weights. Initially, we tried to train the HardNet and D2Net methods on our training dataset using their original training algorithms, but the results did not exhibit any improvements. We did not try to train the NCNet, since this method outputs directly matches and consumes a lot of computational resources, which is undesirable with our target applications capable of running on a mobile device.



**Fig. 5.** Comparison between the *best pose* (bp) and the *refined pose* (rp) using different descriptors on GeoPose3K using *cross-domain* matches between the query photograph and synthetically rendered panorama. **Left:** translation error, **right:** rotation error.

#### 4.1 Test Datasets

For evaluation of our method in a cross-domain scenario, we use the publicly available dataset GeoPose3K [7] spanning an area of the European Alps. We used the standard publicly available test split of 516 images [8]. We note that we were very careful while constructing our training dataset *not* to overlap with the test area of the GeoPose3K dataset. To illustrate that our method generalizes over the borders of the European Alps, on which it was trained, we also introduce three more test sets: *Nepal* (244 images), *Andes Huascarán* (126 images), and *Yosemite* (644 images). The *Nepal* and *Yosemite* datasets were constructed using SfM reconstruction using SIFT keypoints aligned to the terrain model with the iterative closest points algorithm as described by Brejcha et al. [9]. The *Huascarán* dataset has been constructed using our novel approach, as described in Sec. 3.1. Please note that this particular dataset may therefore be biased towards D2Net [12] matchable points, while *Nepal* and *Yosemite* datasets might be biased towards SIFT matchable points. Unlike the training images, camera poses in the test sets were manually inspected and outliers were removed.

#### 4.2 Ablation Studies

**Best Pose and Refined Pose.** We study the behavior of our cross-domain pose estimation approach on the GeoPose3K dataset, on which we evaluate the *best pose* (solid) and the *refined pose* (dashed) for three different embedding algorithms as illustrated in Fig. 5. In the left plot, we can see that the *refined pose* improves over the *best pose* for both HardNet++ and our method for well registered images (up to distance error around 300 m), whereas it decreases result quality with D2Net. We hypothesize that this is because in the pose refinement step, the descriptor needs to disambiguate between more distractors compared

**Table 1.** Comparison of different training strategies of our network on the pose estimation task on GeoPose3K dataset using *cross-domain* matches between the query photograph and the rendered panorama. The higher number the better. Adaptive semihard (ASH) performs better than random semihard (RSH).

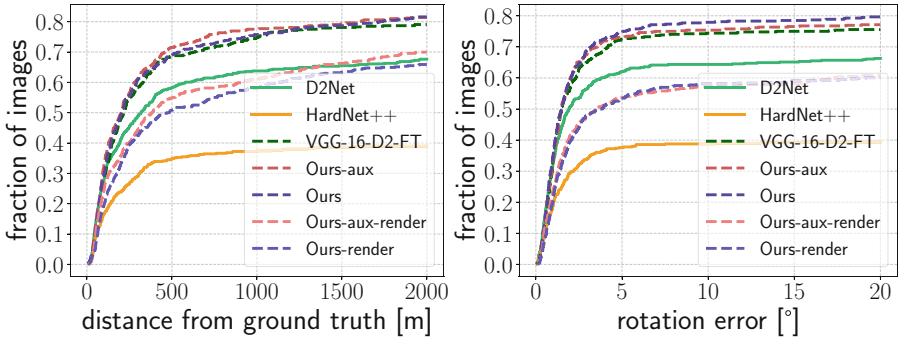
Method	Position error [m]					Rotation error [°]				
	100	300	500	700	900	1	3	5	7	9
	Cumulative fraction of photographs									
Ours-RSH	0.29	0.53	0.61	0.65	0.67	0.34	0.56	0.60	0.63	0.64
Ours-ASH	<b>0.30</b>	<b>0.54</b>	<b>0.63</b>	<b>0.67</b>	<b>0.70</b>	<b>0.39</b>	<b>0.60</b>	<b>0.65</b>	<b>0.68</b>	<b>0.69</b>

to the case of the best pose, where a single photograph is matched with a single rendered image, and D2Net seems to be more sensitive to these distractors than other approaches. Furthermore, the right plot of the Fig. 5 shows that the rotation error is improved on the refined pose for all three methods up to the threshold of  $5^\circ$ . Since points from multiple rendered views are already matched, the subsequent matching step covers a wider FOV, and thus a more reliable rotation can be found. For the following experiments, we use the *refined pose*, which seems to estimate camera poses with slightly better accuracy in the low-error regime.

**Random Semi-hard and Adaptive Semi-hard Negative Mining.** We analyze the difference between the baseline random semi-hard negative mining and adaptive semi-hard negative mining in Table 1. The experiment illustrates that adaptive semi-hard negative mining improves the random semi-hard negative mining baseline in both position and orientation errors, so we use it in all experiments.

**Auxiliary Loss.** Our network trained with the auxiliary loss function performs the best in the cross-domain scenario evaluated on the GeoPose3K dataset (Fig. 6, see Ours-aux). On this task, it outperforms the cross-domain variant of our network trained with the basic loss function (Ours). We also report the result of our network using a single encoder for both domains (Ours-render) which is consistently worse than the cross-domain variant. Furthermore, we see here that our network significantly outperforms both D2Net and HardNet++ in this task.

**Stability with Respect to DEM Sampling Density.** One question is how close does our DEM render have to be to the true photo location, for us to still find a correct pose estimate. To evaluate this, for each query photograph (with known ground truth location), we render a synthetic reference panorama offset from the photo location by a random amount (the “baseline”), sampled from a gaussian distribution with parameters  $\mathcal{N}(0 \text{ m}, 1000 \text{ m})$ . We then estimate

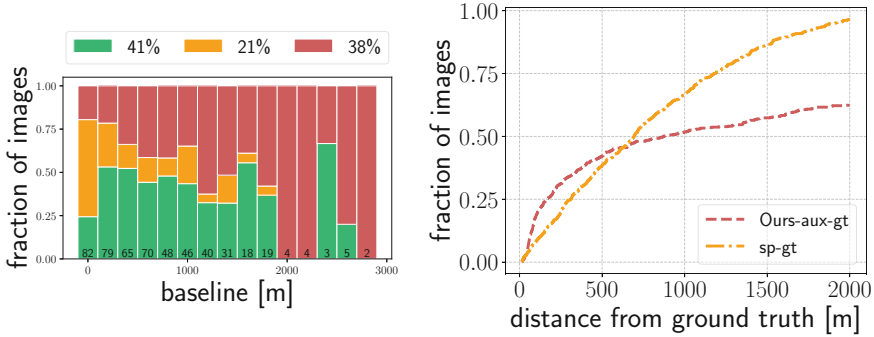


**Fig. 6.** Comparison of variants of our network with HardNet++ and D2Net for pose estimation task on GeoPose3K using *cross-domain* matches between query photograph and synthetically rendered panorama. **Left:** translation error, **right:** rotation error.

the pose of the query photograph by registering it with the render, and compare the predicted location to the known ground truth location. In Fig. 7-left we show the percentage of cases where the distance from ground truth to the predicted location was predicted to be less than the baseline. This gives us a measure for example, of how incorrect the GPS signal from a photo could be such that our approach improves localization. With low baselines, we see that the geometry mismatch to the DEM dominates and the position is difficult to improve on. With baselines over 200 m, we are able to register the photo, and then performance slowly degrades with increased baselines as matching becomes more difficult. Figure 7-right shows that the cross-over point where the position no longer improves over reference is around 700 m.

### 4.3 Comparison with State-of-the-Art

We compare our two-branch method and single-branch method based on VGG-16 with three state-of-the-art descriptors and matchers: HardNet [26], D2Net [12], and NCNet [29] in four different locations across the Earth. According to the results in Fig. 8, our two-branch method trained with auxiliary loss function (Ours-aux) exhibits the best performance on *GeoPose3K*, *Nepal*, and *Yosemite* datasets. The only dataset where our two-branch architecture is on-par with D2Net is *Andes Huascanan* (where the ground truth was created by D2Net matching), and where the single-branch VGG-16 architecture trained using our method and data performs the best. This is most probably due to differences in the ortho-photo texture used to render synthetic images. As the larger, pre-trained VGG-16 backbone has most likely learned more general filters than our two-branch network, which was trained solely on our dataset.

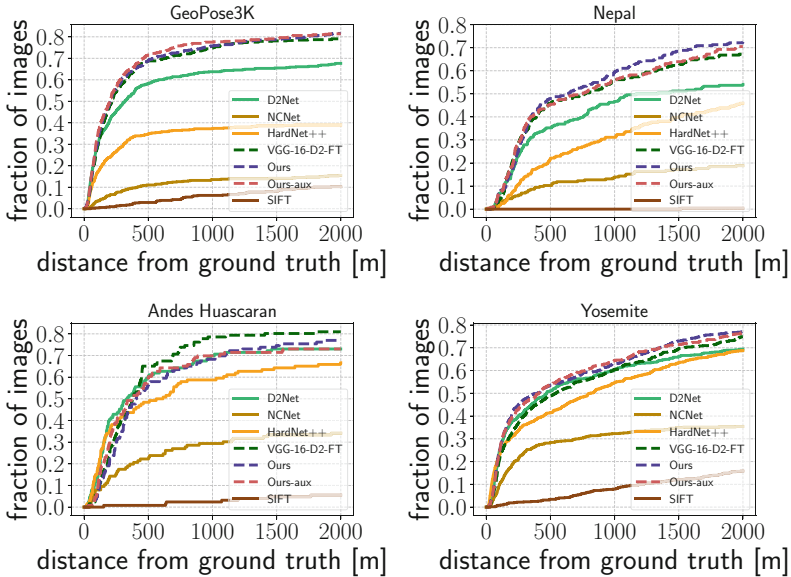


**Fig. 7.** Evaluation of robustness to baseline. **Left:** Fraction of improved (green), worsened (yellow), and failed (red) positions when matching query photo to a synthetic panorama as a function of baseline. The baseline is the distance between the ground truth position and a *reference position* generated by adding a gaussian noise  $\mathcal{N}(0\text{m}, 1000\text{m})$  to the ground truth position. Position is considered improved when the estimated distance to ground truth is less than the baseline. The numbers at the bottom of each bar give the total number of images within each bar. **Right:** Cumulative fraction of query photos with an estimated position less than a given distance from ground truth (Ours-aux in pink) versus the cumulative fraction of *reference positions* within a given distance of ground truth (sp-gt in yellow). Pink line above yellow line means our method improves over the sampled *reference position* at that baseline. (Color figure online)

## 5 Applications

**Mobile Application.** To demonstrate the practicality of our method, we implemented it in an iPhone application. The application takes a camera stream, an initial rotation and position derived from on-board device sensors, and renders synthetic views from the local DEM and ortho-photo textures. It then computes SIFT keypoints on both a still image from the camera stream and the synthetically rendered image and uses our trained CNN to extract local features on the detected keypoints. These features are matched across domains and are then unprojected from the rendered image using the camera parameters and the depth map. Finally, matches between the 2D still keypoints and 3D rendered keypoints are used to estimate the camera pose using  $PnP$  method with RANSAC. This estimated camera pose is used to update the camera position and rotation to improve the alignment of the input camera stream with the terrain model (see Fig. 9).

**Automatic Photo Augmentation.** Furthermore, we demonstrate another use-case of our camera pose estimation approach by augmenting pictures from the internet for which the prior orientation is unknown and GPS position imprecise, see Fig. 9. Please note that many further applications of our method are possible, e.g., image annotation [4, 22], dehazing, relighting [22], or refocusing and depth-of-field simulation [10].



**Fig. 8.** Comparison of our method with state-of-the-art descriptors in four different locations across the Earth. Our method (dashed red and blue) outperforms HardNet [26] on all datasets and D2Net [12] on GeoPose3K, Nepal and Yosemite. Our method seems to be on par with D2Net on Andes Huascanan dataset which has significantly less precise textures (from ESA RapidEye satellite) in comparison to other datasets. (Color figure online)



**Fig. 9.** An iPhone application (in the left) is used to capture the photograph (in the middle) for which precise camera pose is estimated using our method. The estimated camera pose (in the right) is used to augment the query photograph with contour lines (white) and rivers (blue). (Color figure online)

## 6 Conclusion and Future Work

We have presented a method for photo-to-terrain alignment for use in augmented reality applications. By training a network on a cross-domain feature embedding, we were able to bridge the domain gap between rendered and real images. This embedding allows for accurate alignment of a photo, or camera view, to the terrain for applications in mobile AR and photo augmentation.

Our approach compares favorably to the state-of-art in alignment accuracy, and is much smaller and more performant, facilitating mobile applications. We see this method as especially applicable when virtual information is to be visually aligned with real terrain, e.g., for educational purposes in scenarios where sensor data is not sufficiently accurate for the purpose. Going forward, we expect that our method could be made more performant and robust by developing a dedicated keypoint detector capable of judging which real and synthetic points are more likely to map across the domain gap.

**Acknowledgement.** This work was supported by project no. LTAIZ19004 Deep-Learning Approach to Topographical Image Analysis; by the Ministry of Education, Youth and Sports of the Czech Republic within the activity INTER-EXCELENCE (LT), subactivity INTER-ACTION (LTA), ID: SMSM2019LTAIZ. Computational resources were partly supplied by the project e-Infrastruktura CZ (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures. Satellite Imagery: Data provided by the European Space Agency.

## References

1. Aguilera, C.A., Aguilera, F.J., Sappa, A.D., Toledo, R.: Learning cross-spectral similarity measures with deep convolutional neural networks. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 267–275 (2016). <https://doi.org/10.1109/CVPRW.2016.40>
2. Aguilera, C.A., Sappa, A.D., Aguilera, C., Toledo, R.: Cross-spectral local descriptors via quadruplet network. *Sensors (Switzerland)* **17**(4), 1–14 (2017). <https://doi.org/10.3390/s17040873>
3. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. *Arxiv* (2015). <http://arxiv.org/abs/1511.07247>
4. Baboud, L., Čadík, M., Eisemann, E., Seidel, H.P.: Automatic photo-to-terrain alignment for the annotation of mountain pictures. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, pp. 41–48. IEEE Computer Society, Washington (2011). <https://doi.org/10.1109/CVPR.2011.5995727>
5. Baruch, E.B., Keller, Y.: Multimodal matching using a hybrid convolutional neural network. *CoRR* abs/1810.12941 (2018). <http://arxiv.org/abs/1810.12941>
6. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). [https://doi.org/10.1007/11744023\\_32](https://doi.org/10.1007/11744023_32)
7. Brejcha, J., Čadík, M.: GeoPose3K: mountain landscape dataset for camera pose estimation in outdoor environments. *Image Vis. Comput.* **66**, 1–14 (2017). <https://doi.org/10.1016/j.imavis.2017.05.009>
8. Brejcha, J., Čadík, M.: Camera orientation estimation in natural scenes using semantic cues. In: 2018 International Conference on 3D Vision (3DV), pp. 208–217, September 2018. <https://doi.org/10.1109/3DV.2018.00033>



9. Brejcha, J., Lukáč, M., Chen, Z., DiVerdi, S., Čadík, M.: Immersive trip reports. In: Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, UIST 2018, pp. 389–401. Association for Computing Machinery, New York (2018). <https://doi.org/10.1145/3242587.3242653>
10. Čadík, M., Sýkora, D., Lee, S.: Automated outdoor depth-map generation and alignment. *Elsevier Comput. Graph.* **74**, 109–118 (2018)
11. Chen, J., Tian, J.: Real-time multi-modal rigid registration based on a novel symmetric-SIFT descriptor. *Prog. Nat. Sci.* **19**(5), 643–651 (2009). <https://doi.org/10.1016/j.pnsc.2008.06.029>
12. Dusmanu, M., et al.: D2-Net: a trainable CNN for joint detection and description of local features. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. <http://arxiv.org/abs/1905.03561>
13. En, S., Lechervy, A., Jurie, F.: TS-NET: Combining modality specific and common features for multimodal patch matching. In: Proceedings - International Conference on Image Processing, ICIP, pp. 3024–3028 (2018). <https://doi.org/10.1109/ICIP.2018.8451804>
14. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
15. Georgakis, G., Karanam, S., Wu, Z., Ernst, J., Kosecka, J.: End-to-end learning of keypoint detector and descriptor for pose invariant 3D matching, February 2018. <http://arxiv.org/abs/1802.07869>
16. Georgakis, G., Karanam, S., Wu, Z., Kosecka, J.: Learning local RGB-to-CAD correspondences for object pose estimation. In: The IEEE International Conference on Computer Vision (ICCV), October 2019
17. Harwood, B., Vijay Kumar, B.G., Carneiro, G., Reid, I., Drummond, T.: Smart mining for deep metric learning. In: Proceedings of the IEEE International Conference on Computer Vision (2017). <https://doi.org/10.1109/ICCV.2017.307>
18. Hasan, M., Pickering, M.R., Jia, X.: Modified sift for multi-modal remote sensing image registration. In: 2012 IEEE International Geoscience and Remote Sensing Symposium, pp. 2348–2351, July 2012. <https://doi.org/10.1109/IGARSS.2012.6351023>
19. Irani, M., Anandan, P.: Robust multi-sensor image alignment. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), pp. 959–966, January 1998. <https://doi.org/10.1109/ICCV.1998.710832>
20. Keller, Y., Averbuch, A.: Multisensor image registration via implicit similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(5), 794–801 (2006). <https://doi.org/10.1109/TPAMI.2006.100>
21. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
22. Kopf, J., et al.: Deep photo: model-based photograph enhancement and viewing. In: Transactions on Graphics (Proceedings of SIGGRAPH Asia), vol. 27, no. 6, article no. 116 (2008)
23. Kwon, Y.P., Kim, H., Konjevod, G., McMains, S.: Dude (duality descriptor): a robust descriptor for disparate images using line segment duality. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 310–314, September 2016. <https://doi.org/10.1109/ICIP.2016.7532369>
24. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: an accurate O(n) solution to the PnP problem. *Int. J. Comput. Vision* (2009). <https://doi.org/10.1007/s11263-008-0152-6>

25. Lowe, D.G., et al.: Object recognition from local scale-invariant features. In: ICCV, vol. 99, pp. 1150–1157 (1999)
26. Mishchuk, A., Mishkin, D., Radenović, F., Matas, J.: Working hard to know your neighbor's margins: local descriptor learning loss. In: Advances in Neural Information Processing Systems, NIPS 2017, vol. 2017-Decem, pp. 4827–4838. Curran Associates Inc., Red Hook (2017)
27. Nagy, B.: A new method of improving the azimuth in mountainous terrain by skyline matching. PFG – J. Photogrammetry Remote Sens. Geoinform. Sci. **88**(2), 121–131 (2020). <https://doi.org/10.1007/s41064-020-00093-1>
28. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 0756–777 (2004)
29. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. In: Advances in Neural Information Processing Systems, vol. 2018-Decem, pp. 1651–1662 (2018)
30. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the IEEE International Conference on Computer Vision (2011). <https://doi.org/10.1109/ICCV.2011.6126544>
31. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of CNN-based absolute camera pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3302–3312 (2019)
32. Kim, S., Min, D., Ham, B., Ryu, S., Do, M.N., Sohn, K.: DASC: dense adaptive self-correlation descriptor for multi-modal and multi-spectral correspondence. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2103–2112, June 2015. <https://doi.org/10.1109/CVPR.2015.7298822>
33. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2007. <https://doi.org/10.1109/CVPR.2007.383198>
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2015)
35. Tian, Y., Fan, B., Wu, F.: L2-Net: deep learning of discriminative patch descriptor in euclidean space. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6128–6136, July 2017. <https://doi.org/10.1109/CVPR.2017.649>
36. Viola, P., Wells, W.M.: Alignment by maximization of mutual information. Int. J. Comput. Vision **24**(2), 137–154 (1997). <https://doi.org/10.1023/A:1007958904918>
37. Wang, C.P., Wilson, K., Snavely, N.: Accurate georegistration of point clouds using geographic data. In: 2013 International Conference on 3DTV-Conference, pp. 33–40 (2013). <https://doi.org/10.1109/3DV.2013.13>
38. Weyand, T., Kostrikov, I., Philbin, J.: PlaNet - photo geolocation with convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 37–55. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46484-8\\_3](https://doi.org/10.1007/978-3-319-46484-8_3)
39. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: learned invariant feature transform. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 467–483. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46466-4\\_28](https://doi.org/10.1007/978-3-319-46466-4_28)