GeoPose3K: Mountain Landscape Dataset for Camera Pose Estimation in Outdoor Environments

Jan Brejcha[a,*], Martin Čadík[a]

[a] *Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2 612 66 Brno, Czech Republic*

Abstract

We introduce a new dataset called *GeoPose3K*[1] which contains over three thousand precise camera poses of mountain landscape images. In addition to camera location and orientation, we provide data for the training and evaluation of computer vision methods and applications in the context of outdoor scenes; synthetic depth maps, normal maps, illumination simulation and semantic labels. In order to illustrate properties of the dataset, we compare results achieved by state-of-the-art visual geo-localization method on *GeoPose3K* with results achieved on an existing dataset for visual geo-localization. So as to foster research of computer vision algorithms for outdoor environments, several novel future use-cases of our new *GeoPose3K* dataset are proposed.

Keywords: camera pose estimation, visual geo-localization, camera orientation estimation, image-to-model registration, digital terrain models, semantic segmentation

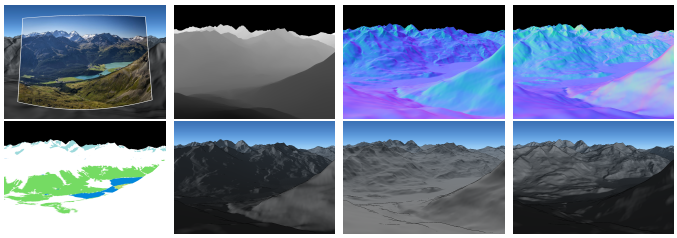


Figure 1: *GeoPose3K* dataset: for each mountain landscape photograph, the dataset contains (in reading order) its GPS coordinate and camera orientation, distance from the camera in meters, normals w.r.t. camera, normals w.r.t. cardinal direction, semantic labels and approximate illumination during the day (here shown at 5am, 12pm and 8pm).

1. Introduction

Camera location and orientation are key attributes of every photograph. Many non-conventional applications can be developed, with a known location and the orientation of a camera. These include computational photography and image enhancement techniques [1, 2], realistic texture synthesis [3], photographs visualization [4, 5], scene understanding [6], and image retrieval based on view direction [7].

However, *geo-location* (geographical coordinates) and camera poses are unknown for many photographs. While many image capturing devices track geo-location via Global Positioning System (GPS), it may be imprecise in various environments (*e.g.* mountains), forgotten or unknown for older photographs. Camera orientation can also be estimated with onboard sensors, but in practice it is unknown for a majority of photographs. Visual *geo-localization* aims to assess the camera location solely from image content. This problem is very challenging, particularly in mountainous scenes, and has been the subject of several studies in the last two decades [8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. Such methods are often built from blocks that need to be trained in a supervised manner. These are mostly detectors of various features like horizon lines, silhouette edges, keypoints, semantic segments (forests, glaciers, bodies of water, rocks, etc.), depth, normals and the direction of sun rays. Moreover, the evaluation and comparison of current methods is the key for further research. Success of all these tasks depends on convenient datasets containing enough training and evaluation samples. Unfortunately, the number of public datasets for visual *geo-localization* and *pose estimation* outdoors is very limited. We are aware of only two datasets for visual *geo-localization* (CH1 [13], CH2 [17]) and a single dataset for camera orientation estimation [18] in mountains.

*Corresponding author
Email address: `ibrejcha@fit.vutbr.cz` (Jan Brejcha)

Contributions. We present a new dataset *GeoPose3K* which addresses three main issues with existing datasets for camera pose estimation outdoors: (I) a small amount of images with known ground truth position, (II) an absence of full camera orientation and (III) an absence of metadata for the training and evaluation of feature detectors and further applications outdoors. The dataset *GeoPose3K* consists of more than three thousand photographs collected mainly from the photo sharing site Flickr.com. All photographs originate in the Alps region, which is the highest mountain range in Europe. For each image all camera pose parameters (GPS position, FOV (field-of-view), full orientation) are provided. The camera pose parameters were assessed with an image-to-model matching technique and manually verified. In order to enable the training and development of future approaches for outdoor environments, we provide various synthetic data per image: depth map, normal map, simulation of illumination during the day, and semantic labels. One sample image from our dataset with corresponding synthetic data is shown in Fig. 1.

A semi-automatic method for dataset acquisition is introduced. We have improved the camera orientation approach of Baboud *et al.* [19] by incorporating weights into the original *Alignment Metric* and by training a specialized mountain silhouette detector. Using this method we have built a procedure for refining noisy estimations of a camera position and FOV. In order to illustrate properties of *GeoPose3K*, baseline measurements of state-of-the-art horizon-based localization method is introduced [17].

2. Related Work

We are not aware of any other dataset that provides precise camera poses for arbitrary outdoor images from a photo sharing site. Furthermore, we are not aware of any similar mountain dataset that provides data for training and evaluating feasible features for outdoor – like depth, normals, illumination changes during the day and semantic segments. However, some datasets for visual geo-localization have been published recently.

Geo-localization datasets for urban environments. Visual geo-localization, place recognition and camera pose estimation is a widely studied topic closely connected with data from social media [20]. The topic gained a lot of interest in the context of urban environments, where image-based methods proved to be a successful solution [21, 22, 23, 24, 25, 26, 27, 28, 29, 30].

While it is relatively easy to obtain a lot of user-tagged images from services like Flickr [31, 32], the quality of annotations (or GPS tags) from these sources is uncertain. This issue is alleviated by using street level imagery obtained by a mapping vehicle [33], or Google Street View images [27, 34].

Geo-localization datasets for natural environments. The geo-localization approaches for mountainous terrain are mainly based on local features extracted from a horizon line [15, 16, 17], or multimodal measurement fusion [14]. The camera orientation estimation problem was approached by matching edges in the query image and synthetic panorama of Baboud *et al.* [19] and later using the smartphone sensors of Porzi *et al.* [18].

Porzi *et al.* [18] also published the Venturi Mountain Dataset[2] with annotated camera poses for video frames. It contains 3,117 video frames from 12 video sequences. For this reason the Venturi dataset contains a lot of similar images: while it is suitable as a benchmark for camera orientation estimation, it is not a suitable benchmark for geo-localization problems. An image-based dataset from the Alps region called Alps100K was collected by Čadík *et al.* [35]. It was downloaded from Flickr by querying hill names and filtering out evident outliers using CNN. The photos in this dataset contain a GPS position, elevation and FOV. However, the dataset does not provide *camera pose* parameters and the provided ground truth *geo-locations* were not verified and thus might be noisy. Datasets for visual geo-localization called CH1 and CH2 were provided by Saurer *et al.* [17]. Both datasets contain in total a thousand images with known ground truth GPS location and FOV, but *camera orientation* is not provided. Segmentation of the sky and foreground terrain is provided for 203 images in the CH1 dataset.

RGB-D datasets. Since we provide additional synthetic metadata as depth and semantic labels, etc. (see Fig. 1), we also briefly overview works introducing existing datasets containing similar data. Thanks to the ease of RGB-D images acquisition using devices such as Microsoft Kinect, many indoor RGB-D datasets exist [36]. Acquisition of outdoor RGB-D datasets is more challenging, because the depth range and resolution of depth sensors is limited. Saxena *et al.* [37, 38] used a laser depth scanner with a maximum depth of 81 m and resolution 55×305; Kitti dataset [39] contains 3D point clouds collected by a LIDAR sensor. However, such approaches are unusable for mountainous environments, where the depth of the scene varies from several meters to hundreds of kilometers. An option suitable for mountainous environments would be to calculate depth from two stereo images, but the disparity needed to obtain viable results would be prohibitive for practical scenarios. We solve these problems by rendering the corresponding depth for each image from a DEM (Digital Elevation Model).

Semantic segmentation datasets. Several standard datasets for semantic segmentation exist [40, 41, 42, 43]. The methods and datasets for semantic segmentation are usually generic – they contain a number of classes that are supposed to cover various kinds of content. While existing

[2] `https://venturi.fbk.eu/results/public-datasets/mountain-dataset/`

datasets, such as Pascal-Context dataset [42] contain relevant classes for mountainous areas – *mountain*, *rock*, *tree*, *grass*, *water*, *road*, *snow* or *sky*, it does not provide mountain specific classes – *forest*, *glacier*, *cliff* or *moor*. With this motivation, we include synthetic semantic labels into *GeoPose3K* dataset. We overlay the DEM with the 13 most relevant OpenStreetMap natural features[3] and for each image in the dataset render a corresponding synthetic view containing the semantic labels (see Fig. 3).

An approach to camera orientation estimation in outdoor scenes based on semantic segments was published by Baatz *et al.* [44]. However, they used only four classes – residential area, bodies of water, sky, and "everything else". Also, their dataset seems to only contain several images and is not publicly available.

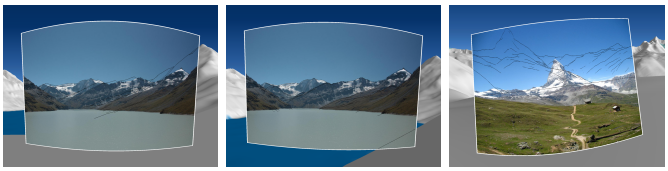


Figure 2: Example of a *viable*, but contaminated camera pose (left), refined camera pose (middle) and *incorrect* camera pose (right). Synthetic mountain silhouettes are overlaid with the aligned image. Image credit: left and middle image – Flickr.com user *Michael Holtrop*: `https://www.flickr.com/photos/bartje_assen/2851555201/`, right image – Flickr.com user *Bossi*: `https://www.flickr.com/photos/thisisbossi/2973222425/`

3. GeoPose3K Dataset

We introduce *GeoPose3K*, a dataset of images with known parameters – field-of-view f, camera position: latitude a, longitude o, elevation e, and camera orientation: yaw α, pitch β and roll γ. The dataset consists of two main parts. The first part contains 339 images captured and annotated manually (which is over 10% of the whole dataset). For each image the GPS position was recorded by its authors using a GPS sensor. Camera orientation was found by selecting correct correspondences of the image and the DEM in a way similar to Kopf *et al.* [2]. However, such manual collection and annotation is a lengthy and tedious task. Therefore, the second part of the dataset (2,772 images) was calculated using a semi-automatic algorithm. The whole dataset consists mainly of images from an online photo service. In addition, we also assessed orientations for the CH1 dataset images which originally contained only the camera position and field-of-view.

3.1. Camera Pose Assessment

We used photographs with a known FOV and GPS position $P = (f, a, o, e)$ from the Alps100K dataset by Čadík *et al.* [35]. These photographs were originally acquired from Flickr online sharing service, so we assume the parameters P to be known, but noisy. Our goals were: (I) for each image I recover a correct elevation and estimate camera orientation $O = (\alpha, \beta, \gamma)$ so that a complete camera pose $C(I) = (\mathrm{P,O})$ could be assembled; (II) classify each recovered camera pose as *viable* or *incorrect*; and (III) refine parameters of each *viable* camera pose C. We assume the camera pose of a given image to be *viable*, if a human user observes obvious correspondence between the query image and the synthetic image rendered from DEM with given camera pose parameters (see Fig. 2). It should be noted that this does not necessarily mean the camera pose is 100% correct; for this reason we pass *viable* images to the refinement process. An *incorrect* camera pose means there is no obvious correspondence to the DEM; images with an *incorrect* camera pose are discarded. Images that passed the refinement process were included into the dataset with the best camera pose we found.

Method Outline. Since we know the camera position (a, o) for each image, we recover the elevation simply by querying the DEM at the position. For camera orientation estimation an existing approach by Baboud *et al.* was used [19]. We propose an improvement of their *Alignment Metric* for camera orientation estimation, and show that it performs better than the baseline. We have used the improved *Weighted Alignment Metric* to automatically estimate the camera pose of 30K photos from the Alps100K dataset. We manually verified the estimated camera pose of each photo. In case the found camera pose was *viable*, we added the photo into a list of candidates.

For each photo in the list which consisted of more than 3K candidates we sampled several hypotheses of the FOV and their position around the original FOV f and position (a, o). The position is sampled in order to mitigate the positional error introduced by an imprecise GPS tag. The FOV is sampled in order to eliminate possible inaccuracies of the recorded focal length or camera sensor size. The FOV might be incorrect due to several factors. First, the image might be cropped (many images in the Alps100K dataset contained artistic borders, which were cropped automatically). Secondly, the FOV might be wrongly calculated due to an incomplete list of cameras and their sensor sizes. Finally, the values stored in EXIF might be manipulated by third party software before sharing it online. One should note that moving the camera while keeping the original FOV fixed is not equivalent to adjusting the FOV with a fixed position. According to Hartley and Zisserman [45], 3D scenes containing objects near the camera are perspectively distorted, hence moving the camera towards/backwards a nearby mountain will change the perspective distortion, while a decrease/increase of FOV (zoom in/out) does not affect it.

For each sampled hypothesis we ran the camera pose

[3] `http://wiki.openstreetmap.org/wiki/Key:natural`

estimation method again and manually chose the camera pose that visually matched the DEM best. Only if the resulting camera pose matched the DEM precisely (we tolerate error up to several pixels), it was then added into the dataset.

3.1.1. Alignment Metric

For each possible camera orientation (α, β, γ), the original method by Baboud *et al.* [19] calculates the image-to-DEM matching score using edges from a query image and silhouettes extracted from the panoramic rendering of a digital terrain model. This score can be used to select an optimal camera orientation for a specific camera position; however, it does not reflect the confidence of the found camera pose as its absolute value varies. The found camera pose has to be visually inspected by a human user in order to recognize a *viable* or *incorrect* result. A detailed description of the *Original Alignment Metric* and our improved *Weighted Alignment Metric* can be found in Appendix A.

3.1.2. Candidate Refinement

Images, for which the *Weighted Alignment Metric* was able to recover a *viable* camera pose, were added into a list of candidates. The camera pose could be contaminated due to a combination of many factors: an imprecise GPS tag, imprecise FOV, imprecise DEM, and the distortion of a query image. Since there was a high number of images with *viable*, but contaminated camera pose after the camera pose estimation process, we further refined contaminated camera poses.

We hypothesised a position with 8 samples regularly placed around the original position (a, o). Four samples were placed in the corners of a smaller square with a side of 500 m, and four samples were placed in the corners of a bigger square with a side of 1000 m. The original position (a, o) is located in the center of both nested squares. For each new position we have also sampled FOV f of the camera. The minimum value of FOV was $f - 0.1f$, the maximum $f + 0.1f$ and there were in total four steps sampled linearly between the minimal and maximal value. We ran our *Weighted Alignment Metric* on each sampled position. In this way we obtained thirty-two new camera poses for each candidate. Finally, these camera poses were verified by the user. In case the best camera pose of the candidate was precise enough (the alignment error was not bigger than several pixels, see middle image in Fig. 2), the new refined camera pose was added into the dataset.

The process of candidate refinement was very demanding on computational resources and time. The alignment of all sampled positions and FOV's took three weeks on seven computers equipped with Intel Core i3-4360 CPU and NVidia GTX 980 GPU. In addition, it took one man-month to manually assess the estimated camera poses.

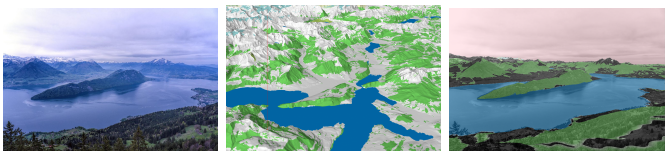


Figure 3: Example of OpenStreetMap semantic segments provided per dataset image. Left: original photograph. Middle: terrain metadata from OpenStreetMap [46] rendered on the digital elevation model. Right: original image overlaid with terrain metadata from OpenStreetMap. Color coding: **sky**, **water**, **forest**, **glacier**, **rock**, **other**.

3.2. Synthetic Data Acquisition

Each image I in the dataset is provided with a camera pose $C(I) = (f, a, o, e, \alpha, \beta, \gamma)$. This camera pose allowed us to support the dataset with additional synthetic data rendered from DEM.

Depth. A depth map was acquired by pixel-wise raycasting and the measuring of the distance from the camera to the first intersection with scene geometry. The accuracy of the depth map is dependent on the accuracy of the DEM; our DEM consists of samples spaced by 24 meters and the DEM was obtained from the viewfinderpanoramas website[4].

Normals. We produced two types of normals. Normals w.r.t. the camera were calculated relatively to the camera position. Normals w.r.t. cardinal direction were calculated with regard to a world coordinate system. Original normals of the surface $n \in \mathbb{R}^3$, where $n_x, n_y, n_z \in \langle -1, 1 \rangle$ are encoded into the RGB image $n'_{rgb} = 0.5 + 0.5 \cdot n$, where $n'_r, n'_g, n'_b \in \langle 0, 1 \rangle$.

Illumination. Illumination approximation was simulated hour by hour from 4 am till 9 pm on 21st June, when the days are the longest during a year. The illumination simulation was calculated using a local illumination model, so that it does not contain casted shadows.

Semantic Segments. We make use of publicly available metadata from OpenStreetMap [46]; however, other sources (e.g. NASA Visible Earth[5] or USGS Land Cover[6]) are feasible too. More specifically, we render 13 natural and physical land features from OpenStreetMap natural feature set[7]: bare rock, cliff, fell, forest = wood, glacier, grassland, moor, scree, shingle, sinkhole and water. Each feature is mapped on one color layer in a geo-referenced texture. The texture is subsequently mapped on the surface of our 3D terrain model (see Fig. 3, middle). Assuming the image (e.g. Fig. 3, left) is correctly aligned with

[4] `http://viewfinderpanoramas.org`
[5] `http://visibleearth.nasa.gov/view.php?id=61004`
[6] `http://www.usgs.gov`
[7] `http://wiki.openstreetmap.org/wiki/Key:natural`

the model, we can project the texture onto the image using the virtual camera while correcly accounting for the visibility thanks to the 3D terrain model. This results in the final pixel-wise semantic labels (Fig. 3, right).

3.3. Dataset Properties

The *GeoPose3K* dataset consists of two main parts, collected *manually* and *semi-automatically*. The first part contains 339 images, which were collected and annotated manually. Since this was a tedious task, collecting more dataset samples in this way was unfeasible. The second part of the dataset contains 2,772 images for which the camera parameters were optimized using *Weighted Alignment Metric* combined with hypotheses sampling and manual selection of the best candidate, as was described in Sec. 3.1.

3.3.1. Potential Bias for Evaluation of Camera Orientation Estimation Methods

The *semi-automatically* collected part of the dataset was gathered with the help of a method by Baboud *et al.* [19]. For this reason, methods based on edge features might be potentially privileged over algorithms based on different principles. This bias must be taken into account when using GeoPose3K for the evaluation of orientation estimation methods. However, for evaluating problems based on different features, the usage of the dataset is valid. We illustrate this property in a benchmark evaluating a state-of-the-art horizon-based geo-localization method by Saurer *et al.* [17] in Sec. 4. In our evaluation we show that the dataset difficulty on the localization task is on par with datasets collected solely manually (CH1, CH2).

For evaluating methods similar to the method by Baboud *et al.* [19], a *manually* collected part of the dataset (339 images) shall be used. Since the selection of the images in this part was not affected by any algorithm, there is no limitation on which methods can be evaluated using this part of the dataset.

3.3.2. Statistics

The majority of images in the GeoPose3K dataset originate from Alps100K dataset. The GeoPose3K dataset consists of images, which contain an accurate GPS tag, and capture a reasonable portion of the mountain scene so that they might be registered with the terrain model. This definition restricts the set of images in which we can generalize our conclusions on. However, the GeoPose3K dataset has adequate coverage by users – two thousand users have a single photograph in the dataset, around a hundred and fifty users have two photographs, and only a single user has twelve photographs in the dataset, which is the largest number of photographs created by a single user. The majority of images in the dataset were taken between 2007 and 2014. Both the user and year distributions exhibit some degree of similarity to the Yahoo Flickr Creative Commons 100 Million dataset (YFCC100M) [32]. We employed a two-sample Kolmogorov-Smirnov hypothesis test with a null hypothesis: the GeoPose3K and YFCC100M distributions do not differ significantly. For the user distribution, the two-sample Kolmogorov-Smirnov test failed to reject the null hypothesis ($D(6487, 12) = 0.3042, p = 0.18$). For yearly distribution, the same test clearly rejected the null hypothesis ($D(35, 35) = 1, p < 0.001$). From this we can conclude that year's distributions differ significantly, but we were not able to prove the same for user distribution. This fact illustrates that the GeoPose3K and YFCC100M share some degree of similarity, but GeoPose3K is a more specific subset than YFCC100M.

In spite of the above-defined restrictions, the GeoPose3K dataset enables us to speculate about the following important questions. What cameras are the most common in the mountain environment, and which of them are the best candidates for visual geo-localization? Which focal lengths are the most successful for matching mountain images? Are people really taking photographs with a zero roll angle? Which orientation at what time is the most favorite? In order to answer these questions and to illustrate the properties of the *GeoPose3K* dataset, we measured several statistics.

Geographical Distribution. The distribution of *geo-locations* of photographs in the *GeoPose3K* dataset is visualized in Fig. 4 (left). The dataset was built from Alps100K images and hence, the photos are distributed over the whole Alps region. However, the photos are not distributed uniformly – the areas of tourist interest like Switzerland and northern Austria contain more images.

Cameras. A full list of cameras is due to its excessive size attached in Appendix, in Table B.3. The most frequent camera in the dataset is Canon DIGITAL IXUS 860 IS (around 8% of dataset images). Interestingly, the first and third most frequent cameras in our dataset are not equipped with a built-in GPS sensor; according to this fact at least 10% of images in the dataset obtained their original GPS coordinates by a third-party logger, or were geo-tagged manually.

Focal Lengths. A histogram of focal lengths recalculated to 35mm equivalent is visualized in Fig. 4 (middle). Shorter focal lengths are common in mountain landscape images (Alps100K) and the measured distribution of *GeoPose3K* shows this bias as well.

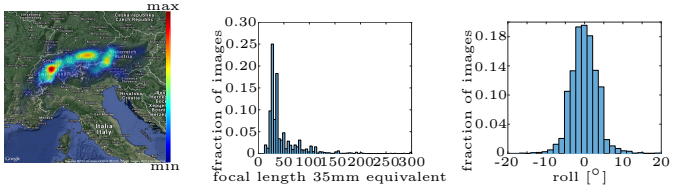


Figure 4: Dataset statistics. Left: distribution of photo locations in *GeoPose3K* dataset. Middle: histogram of focal lengths in the dataset. Right: average camera roll.

Average roll. People usually aim to level their photos with the horizon line. Accordingly, the histogram of roll angles in Fig. 4 is centered around zero. However, keeping the camera level may be difficult in the mountains and some shots are clearly rotated. This justifies that the geo-localization methods need to be optimized for a camera roll as well.

Time-Orientation Correlation. According to Fig. 5 (right) photos in the *GeoPose3K* dataset were taken most frequently around 2pm with a heading of 90°. In general, photos were photographed mainly between 10am and 4pm, and the favorite headings ranges are between 0°-120° and 170°-300°.

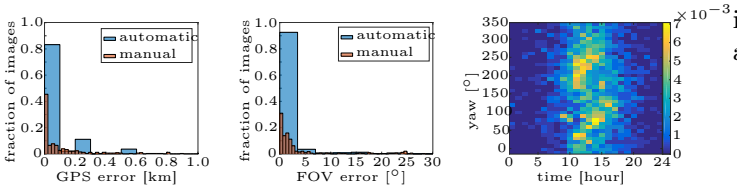


Figure 5: Dataset statistics. Left: histogram of GPS error distribution. Center: histogram of FOV error distribution. Positions and FOV's are refined using our semi-automatic method which are drawn in blue, while the positions refined manually are in brown. Right: time/orientation correlation visualization.

GPS and FOV error. We measured GPS and FOV errors based on the refinement (Sec. 3.1.2) we employed. We measured geo-distance between the original and refined GPS position for each image, and plotted a histogram of these errors (see Fig. 5 on the left). We measured a similar histogram of FOV errors (Fig. 5 in the middle), based on the difference between original and refined FOV. According to our measurements, there are discovered discrepancies in GPS values. The images sometimes exhibit noisy GPS tags, probably due to manual geo-tags, bad reception of a GPS sensor, or the fact that cameras have their GPS refresh rate set to a long time interval. On the contrary, the FOV error histogram peak is near zero, reflecting the fact that the original fileds-of-view of photos in the dataset were nearly correct. Imperfections in FOV up to 1°-2° discovered by the manual annotation are most probably caused by tiny inaccuracies of the digital elevation model or by a very small GPS error. Therefore, the FOV error within a small margin of 1°-2° is assumed to be correct.

Edge accumulation. The *GeoPose3K* dataset allows us to analyze properties and discover the importance of the query edges. We did the following calculations on a uniformly sampled grid of 9×9 samples with a 0.001° resolution in both N-S and W-E directions. From each grid location we have rendered the synthetic silhouettes and matched them to the query image edges. We have incremented 1 to all pixels in the accumulator containing a synthetic silhouette, which contributed positively to the matching score. We ran such an evaluation for every position in the grid and summed up the accumulators in order to obtain one accumulated image for a query. Example of such an accumulated image can be found in the Appendix in Fig. B.10.

In order to analyze the importance of edges in all images in the dataset, we have created the average accumulated edge map (Fig. 6, left). The most populated one is the central area of an image with a slight bias towards the bottom of the frame. Similarly, we have created an average accumulated cylindric panorama image, where each image has been accumulated with respect to its original camera orientation (Fig. 6, right). As one would expect, the area around the horizon is the most populated; however, the silhouettes off ±10°seem to be of similar importance. The image further exhibits a good *GeoPose3K* dataset coverage of camera orientations.

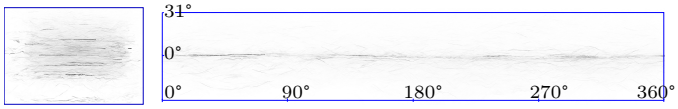


Figure 6: Edge accumulation. Left: Normalized average from all accumulated images. Right: Accumulated panorama image.

4. Experiments

We use the *GeoPose3K* dataset to extensively evaluate the state-of-the-art method by Saurer *et al.* [17] for horizon-based visual geo-localization in mountains. By this evaluation we bring deeper insight into the dataset properties; and according to our measurements, the difficulty of the new *GeoPose3K* dataset for visual geo-localization is similar to the difficulty of the original CH1 [13] and CH2 [17] datasets. However, *GeoPose3K* allows us to evaluate the baseline using more photos covering a larger area than the original datasets. For this evaluation, the vastest area ever issued – our largest experiment deals with area more than twice the size of the original area reported by Saurer *et al.* [17]. In addition, *GeoPose3K* also allows us to evaluate the camera heading accuracy. The evaluated method [17] is capable of camera heading estimation; however, it had never been quantitatively evaluated before, because there was no suitable dataset containing *camera orientations*. Furthermore, we have evaluated geo-localization performance using three fully automatic horizon line extraction methods to illustrate that the automatic horizon line extraction is still a challenging problem.

4.1. Setup

We have reimplemented the state-of-the-art method by Saurer *et al.* [17] and measured its performance on *GeoPose3K*. The method utilizes a database of densely sampled horizon lines from a DEM to retrieve locations given a query horizon line. We have extracted a database of

synthetic horizon lines that covers 86 000 km^2 (red area in Fig. 7(a) – *GP1*), which is more than twice the size the area used in the original paper [17] (40 000 km^2). We have sampled the area of interest in both N-S and W-E directions with a resolution of 0.001°. Samples in N-S and W-E directions are 111 m and 78 m far from each other, respectively.

The original method is based on the bag-of-words retrieval adapted to horizon line contours. The approach allows us to retrieve the approximate heading and position of the camera using a voting scheme. We have used a voting for location and direction with 2.5° and 10° descriptors and 3° directional bin size, which seemed to be the best choice according to the results presented in the original paper. The evaluation method proposed by Baatz *et al.* is used [13]; and the distance between a candidate and a ground truth location is measured, assuming the location is correct if the distance is smaller than 1 km. The cumulative percentage of correctly localized images given top-k candidates is then plotted.

4.2. Automatic Horizon Line Detection

The baseline localization method requires a horizon line as a query input. In order to measure performance of the method on *GeoPose3K* dataset, we have experimented with several algorithms for automatic detection of horizon lines.

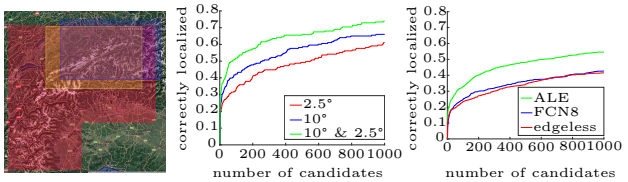
Automatic Labeling Environment (ALE) [47]. ALE is an energy minimization-based semantic segmentation framework adopted for sky extraction by Saurer et al. [17]. Specifically, the energy is predicted by a pixel-wise classifier trained on contextual and superpixel feature representations. Multiple bag-of-words representations over the random set of 200 rectangles, and superpixels are used for the contextual and superpixel parts, respectively. The segmentation is obtained by minimizing the energy using dynamic programming (DP). We have implemented the algorithm [17] into the Automatic Labeling Environment (ALE), with the personal advice of the authors [47]. As with the original paper [17] we set the number of bag-of-words clusters to 512 and trained ALE using the CH1 dataset [17].

An Edge-Less Approach to Horizon Line Detection [48]. This approach also uses machine learning and dynamic programming to extract the horizon line from an image. Specifically, each pixel is assigned a classification score expressing the likelihood of the pixel belonging to the horizon line. As suggested by the authors, we have used the SVM classifier trained by their training set [48]. Assuming that the horizon line extends from left to right (not top to bottom), the horizon line is finally extracted using DP, maximizing the sum of classification scores.

Fully Convolutional Networks (FCN) [49]. FCN achieve state-of-the-art results in semantic segmentation. For the given input image, the fully convolutional network produces a correspondingly-sized semantic segmentation image. We have experimented with several semantic segmentation models (FCN-Xs) and selected the FCN-8s (three stream, 8 pixel prediction stride), which gave us the best results, for further evaluation. We have used a model trained for the 21-class (including background) PASCAL VOC segmentation task and finetuned for sky-foreground segmentation using the CH1 dataset [13].

4.3. Localization Performance

We have evaluated the localization performance using several scenarios. In order to illustrate the performance of our implementation of the baseline method, we measured the performance on the original *CH1* dataset using the original *CH1* horizon lines (*CH1* area, *CH1* data, 7(b)). Since horizon lines were not provided with the *CH2* dataset, we measured the performance of the *CH2* dataset using query horizon lines obtained by automatic segmentation of query images using three different methods (*CH2* area, *CH2* data, 8(a)). In order to study the difficulty of *CH1*, *CH2* and *GeoPose3K* datasets, we have also evaluated the method on the *CH1* and *CH2* areas using *GeoPose3K* data (*CH1* area, *GeoPose3K* data, 7(c); *CH2* area, *GeoPose3K* data, 8(b), respectively). Furthermore, we have studied the total performance of the method using the *GeoPose3K* images inside the largest *GP1* area, 8(c).



(a) localization area (b) *CH1* area, *CH1* data (c) *CH1* area, *GeoPose3K*

Figure 7: (a) trained localization areas: ***CH1* dataset area**, **CH2 dataset area**, **our *GP1* area** (largest); results of horizon-based localization on (b) *CH1* dataset: red – 2.5° features, blue – 10° features, green – combination of both 2.5° and 10° features; (c) *GeoPose3K* data in the *CH1* dataset area (yellow rectangle) using three automatic segmentation techniques – ALE (green), FCN (blue), Edge-Less (red).

CH1 area, CH1 data (Fig. 7(b)). We evaluated the performance of our implementation on the original *CH1* dataset [17]; and used a database of horizons inside the yellow rectangle (Fig. 7(a)) and 203 query images from *CH1* dataset. Like the authors of the baseline method [17], we visualized performance for 2.5° features, 10° features and a combination of both. The obtained performance is a bit worse

than in the original publication. We see the main reason is in the data we use – the original method uses non-free DEM from the Federal Office of Topography swisstopo[8], which contains one sample per 2 m^2. We use publicly available DEM from viewfinderpanoramas [9], which contains one sample per 576 m^2.

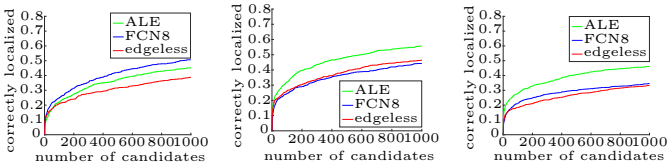
CH1 area, GeoPose3K data (Fig. 7(c)). In this experiment we used 865 query images from *GeoPose3K* located inside the *CH1* area – yellow box (Fig. 7(a)). The fraction of correctly localized images is lower than in Fig. 7(b). This might be caused by the lower accuracy of the horizon extraction algorithms (ALE, FCN-8s, Edge-Less) compared to the ones used in the original *CH1* dataset (ALE guided by user). This assumption is supported by the fact that the performance of the experiment in the *CH2* area (Fig. 8(a) and 8(b)) is similar for both *CH2* and *GeoPose3K* data.

CH2 area, CH2 data (Fig. 8(a)). Here we have used 949 query images from *CH2* dataset located inside the *CH2* area – blue box in Fig. 7(a). Performance of this experiment can be directly compared to the *CH2* area – *GeoPose3K* experiment (Fig. 8(b)), since query horizon lines for both sets were extracted by the same (automatic) techniques. According to the results, the method performed little bit worse on *CH2* dataset.

CH2 area, GeoPose3K data (Fig. 8(b)). For this experiment we used 791 images from *GeoPose3K* dataset, which were located inside the *CH2* area – blue box in Fig. 7(a). The results are in agreement with other experiments; performance of the segmentation techniques is consistent with other experiments. ALE seems to be the best method for horizon line segmentation, FCN and the EdgeLess approach scored similarly.

GP1 area, GeoPose3K data (Fig. 8(c)). The *GeoPose3K* dataset covers almost the whole Alps (Fig. 4). However, training such a large area for the horizon-based localization was not feasible due to hardware limitations. For this reason, we trained the *GP1* area (red area on (Fig. 7(a)), which is the largest area used for horizon-based localization so far (86 000 km^2). In this area we evaluated the method using a subset of 1151 images from *GeoPose3K*, which fit into the *GP1* area. The results of this experiment are in Fig. 8(c). The performance is only slightly worse than in previous experiments (*CH1, CH2* areas, *GeoPose3K* data). This is expected, since the geo-localization area is more than twice the size of the *CH1* and *CH2* areas. From this result it seems that the geo-localization performance of the horizon line-based localization method [17] decreases only marginally with the increasing size of the geo-localization area.

[8] `https://www.swisstopo.admin.ch`
[9] `http://www.viewfinderpanoramas.org`



(a) *CH2* area, *CH2* data (b) *CH2* area, *GeoPose3K* (c) *GP1* area, *GeoPose3K*

Figure 8: results of horizon-based localization using three automatic segmentation algorithms – ALE (green), FCN (blue), Edge-Less (red); (a) *CH2* dataset using automatic segmentation; (b) *CH2* dataset area, using *GeoPose3K* images, (c) largest *GP1* area using *GeoPose3K* images.

4.4. Orientation Performance

Since *GeoPose3K* also contains *camera orientation* for each image, we evaluated the error of estimated heading (Fig. 9). To our knowledge, this is the first such evaluation of heading estimated by the method of Saurer *et al.* [17]. For each correct candidate of all queries we measured the difference between the heading of the ground truth image and the estimated heading. From Fig. 9 we can see that the orientation error peaks around 0° and errors larger than several degrees are negligible. This result supports our expectations: for a correct location, the algorithm is able to find a correct heading estimate up to a small error of several degrees. A deeper analysis of the heading estimation error can be found in Table 1. We measured the heading error on all three geolocalization areas – CH1, CH2 and GP1, using GeoPose3K data. Mean and standard deviation is calculated from the absolute value of the difference between the estimated orientation and the ground truth. Quantiles are calculated from the difference between the estimated orientation and the ground truth. The statistics correspond with Fig. 9. In the CH1 area, the Edge-less segmentation method achieved the best result; however, this is not consistent across other areas. It is likely that CH2 and GP1 area contains more difficult images, since the standard deviation is worse on CH2 than on CH1. In the largest GP1 area, the FCN8-s segmentation method has the lowest heading error according to the reported mean and standard deviation. On average, the lowest mean error in heading accuracy was achieved by ALE, which also has other average statistics that are slightly deviated from the lowest observed values; however, the difference is not significantly different from other methods.

4.5. Experiments Summary

In this section, we provided experimental results of the state-of-the-art horizon-based visual localization technique by Saurer *et al.* [17]. We evaluated both localization and heading estimation performance. For evaluation, we used original CH1 and CH2 datasets and compared the achieved performance with our GeoPose3K dataset. We

CH1 area, GeoPose3K, 865 images					
method	**mean**	**std**	**median**	**q = 0.95**	**q = 0.05**
Edge-less	**7.89**	**22.39**	**-0.76**	**6.62**	-8.78
FCN8-s	9.92	26.73	-1.66	15.74	**-8.77**
ALE	16.10	36.36	-0.80	80.36	-25.54
CH2 area, GeoPose3K, 791 images					
Edge-less	36.28	51.88	**0.39**	124.48	-120.18
FCN8-s	25.68	43.62	-1.80	107.84	-122.17
ALE	**11.76**	**32.04**	-0.79	**29.23**	**-9.08**
GP1 area, GeoPose3K, 1151 images					
Edge-less	14.76	34.89	**-0.18**	105.22	**-7.35**
FCN8-s	**13.00**	**33.97**	-1.24	106.21	-7.53
ALE	18.26	36.10	-0.61	**98.61**	-20.38
Average					
Edge-less	19.64	36.39	**-0.18**	78.77	-45.44
FCN8-s	16.20	**34.77**	-1.57	76.60	-46.16
ALE	**15.37**	34.83	-0.73	**69.40**	**-18.33**

Table 1: Statistics of the camera orientation error in degrees for a localization experiment on GeoPose3K data using three automatic segmentation techniques. Symbol $q = 0.95$ denotes quantile at 0.95.

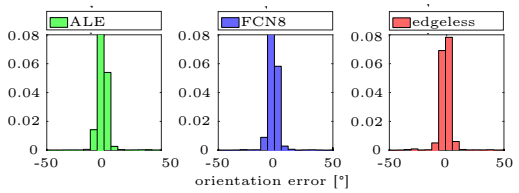


Figure 9: Normalized histograms of camera orientation error (in degrees) for localization experiment on *CH1* area and *GeoPose3K* dataset using three automatic segmentation techniques – ALE (green), FCN (blue), Edge-Less (red).

also conducted the largest horizon-based localization performance experiment ever, with the use of the GeoPose3K dataset and a GP1 area of 86 000 km^2. We identified a large performance gap between automatically estimated horizon lines and manually corrected ones provided with the CH1 dataset. Usually, the method was able to localize around 15% of top-1 candiates with a localization error below 1 km using our dataset. The performance was two times better with the original CH1 dataset: the method localized around 30% of top-1 candidates with a localization error below 1 km. The best method for automatic horizon line extraction is, according to our experiments, Automatic Labeling Environment (ALE) [47] (which scored best in 3 out of 4 experiments), and the second best is the Fully Connected Networks (FCN-8s) approach (which scored best in 1 out of 4 experiments).

For the first time, we also evaluated heading estimation performance of the horizon-based localization method by Saurer *et al.* [17]. Through our experiment we illustrated that candidates located up to a distance of 1 km from the ground truth showed a heading error of around a few degrees, and larger discrepancies from the ground truth heading are rare. In other words, a correctly localized image also implies a correctly estimated heading. However, such an estimated heading is only an approximate estimation, since the usual mean error varies between 0.12° and 11.88° across various scenarios.

5. Future Applications

GeoPose3K dataset is a rich source of information for solving geo-localization, camera orientation and additional computer vision and image processing problems. Besides geographic location and orientation of the camera, it contains additional synthetic data which can serve to train, evaluate and compare existing and future algorithms. We propose future research that can be built using this dataset.

Depth Estimation from Single Image. Depth estimated from a single image is an important cue for image processing algorithms, like single image haze removal [50, 51]. It is an ill-posed problem, since there is no unique mapping from a single RGB image to RGB-D. Prior and contextual information must be taken into account in order to obtain feasible estimates. The prior is usually chosen arbitrarily, such as a dark channel [50]. However, the prior, or the whole end-to-end estimation process can be trained given our synthetic depth and normal data.

Sun Position from Illumination. Sun position is a viable feature for location recognition [52]. Previous work estimates the sun's position given a set of temporal images. Using our synthetic illumination data, single image estimation of the sun's position might be trained in an end-to-end manner.

Semantic Segmentation. Semantic segments proved to be usable for camera orientation assessment [44]. Using semantic labels from *GeoPose3K* makes it possible, not only to quantitatively evaluate existing methods, but also to train semantic segmentation algorithms for outdoor images.

6. Conclusion

We have presented here the *GeoPose3K* dataset for camera pose estimation. We showed, that the dataset is unique and valuable for the training and evaluation of methods in the context of visual *geo-localization* and *camera pose* estimation. We demonstrated an approach to semi-automatic dataset acquisition using an improved camera orientation estimation algorithm. We performed an in-depth analysis of dataset properties and provided the largest baseline evaluation on a geo-localization task using a state-of-the-art visual geo-localization algorithm. Our experiments demonstrated that the *GeoPose3K* is usable for camera orientation and geo-localization evaluation and the difficulty is on par with original *CH1* [13] and *CH2* [17] datasets. In addition, we proposed several unconventional future applications which the dataset enables us to develop.

Acknowledgements

This work was supported by The Ministry of Education, Youth and Sports from the "National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602" and by the IT4Innovations infrastructure which is supported from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center - LM2015070".

This work was supported by SoMoPro II grant (financial contribution from the EU 7 FP People Programme Marie Curie Actions, REA 291782, and from the South Moravian Region).

This work was supported by the TACR Competence Centres project V3C – Visual Computing Competence Center (no. TE01020415).

We would also like to thank Martin Simonovsky for his contribution – Weighted Edge Detector (See Appendix A, *Weighted Edge Detector*).

The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

Appendix A. Camera Orientation Assessment

Original Alignment Metric. We reformulate the problem of matching as introduced by Baboud *et al.* [19] who proposed a matching score per edge es (A.1):

$$es(e,D) = \sum_{j\in e}^{|e|} |c(e_j,D)| \left(\frac{1+c(e_j,D)}{2} d(e_j)^p + \frac{1-c(e_j,D)}{2} m \right), \quad \text{(A.1)}$$

and the final matching score $s(Q,D) = \sum_{e\in Q} es(e,D)$, where Q denotes the set of edges extracted from a query image, and D denotes the set of synthetic silhouettes in which the query image is matched. The term $d(e_j)$ measures the length of the edge segment e_j, p and m are constant parameters. The parameter p defines nonlinear weighting of edges based on their length, and the negative parameter m defines the cost of edge crossings.

The term $c(e_j, D)$ measures the spatial configuration of a query edge segment e_j with respect to a synthetic silhouette segment e_i. In case edge segment e_j is parallel with silhouette segment e_i, the term is equal to 1, and in case the edge segments are crossing each other, the term $c(e_j, D)$ is equal to -1, and to 0 in other cases. Two edge segments are parallel, so that all points of the query edge segment e_j are in the $\mathcal{E}$ neighborhood of the synthetic silhouette segment e_i. In summary, the score $s(Q,D)$ sums up the lengths of edges that are parallel with some synthetic silhouettes, and penalizes edges crossing the synthetic silhouettes.

Weighted Alignment Metric. In the original alignment metric (A.1), all edges were assigned the same importance regardless their visual appearance, even though their appearance can correlate with their importance for matching.

Scenario	Thresholded	Weighted
Compass	2.75%	2.75%
CannyDM	6.20%	0.01%
Silhouette	7.25%	9.75%

Table A.2: Image Registration Experiments: The table shows a fraction of successfully registered images from 400 randomly selected images from Alps100K [35] test set.

In order to improve the matching performance of the original method, we propose to weight image edges based on their strength. We implemented edge strength as a weight of the edge segment $w(e_j) \in \langle 0, 1\rangle$. Weight w is simply multiplied with terms $d(e_j)$ and m in (A.1) respectively, so we get:

$$es(e,D) = \sum_{j\in e}^{|e|} |c(e_j,D)| \left(\frac{1+c(e_j,D)}{2} w(e_j) d(e_j)^p + \frac{1-c(e_j,D)}{2} w(e_j) m \right). \quad \text{(A.2)}$$

Weighted Edge Detector. In order to detect edges from query images with meaningful weights, we adopt the edge detection framework by Dollár and Zitnick [53]. Their approach predicts a 16×16 edge map from a larger 32×32 image patch. Individual predictions are averaged to produce a soft edge map for the whole input image. The learning problem is solved using structured random forests. In order to use standard node splitting criteria, the structured space of labels $\mathcal{Y}$ is mapped to a discrete set of labels $\mathcal{C}$ by a two-stage mapping via an intermediate space $\mathcal{Z}$ at each node. The authors assume segmentation maps being available for training. Instead, we use our synthetic depth maps. So as to be able to use depth maps as labels, we redefine the intermediate mapping $\Pi : \mathcal{Y} \to \mathcal{Z}$ to produce a vector that encodes depth difference $y(j_1) - y(j_2)$ for every unique pair of indices $j_1 \neq j_2$ within a label patch $y \in \mathcal{Y}$. In practice, we sample $m = 256$ dimensions of $\mathcal{Z}$, resulting in a node-specific reduced mapping Π_Φ, which is then further discretized as in the original paper.

Appendix A.1. Performance of Alignment Metric

Both the *original* and the new *Weighted Alignment Metric* (Sec. Appendix A) assume edge maps on their input. We have experimented with several edge map acquisition methods, including novel depth-based approaches (described below) to find the best possible settings. The performance was evaluated by manually counting correctly registered results. As a test set, we randomly selected 400 images from the Alps100K test set [35]. The number of selected images is based on the fact that the alignment metric is demanding on computational time and validation demands a lot of manpower; and testing several variants of edge maps would be too expensive with a bigger test set. The results are summarized in Table A.2 and show that the weighted variant of the silhouette detector is by far the best.

Thresholded edge maps *Compass* edge detector [54] has been used in the baseline metric [19]. Similarly to the authors, we thresholded the edge map ($\tau = 0.7$) to keep

only significant edges. *Canny detector applied on the depth map estimate* is an alternative approach based on the estimated image depth map. The depth map is constructed using a dark channel prior technique directly from an input image [50]. The edges are then obtained from this depth map using a thresholded Canny edge detector, representing depth discontinuities. The resulting edge map often exhibits more distinctive edges further from the camera, as compared to the edges detected directly from the original query image. *Thresholded silhouette detector* is a version of our novel data-driven edge detection method (see below), with a low threshold $\tau = 0.1$.

Weighted edge maps Besides standard thresholded versions, *Weighted Alignment Metric* enables us to incorporate weighted edge maps. We have involved weighted versions of both methods described above, where the raw edge strength was linearly rescaled into edge weights $w \in \langle 0, 1 \rangle$. Additionally, we propose a novel data-driven method for detection of weighted silhouettes from depth maps (*Weighted edge detector*). Our measurements show that the weighted variant of our matching metric produces significantly better results, and the new silhouette detector is the preferred edge map construction method.

Appendix B. GeoPose3K Properties

The following figures were taken from the main text for better clarity in the paper. Fig. B.10 was referenced from Sec. 3.3.2, *Edge accumulation*, and Tab. B.3 was referenced from Sec. 3.3.2, *Cameras*.

Appendix B.1. Edge accumulation

Figure B.10: Accumulation of matched edges for a single query image. We split the area around the GPS location of the query image uniformly into a grid of 9×9 cells (resolution of 0.001° in both N-S and W-E directions). From each cell we render synthetic silhouettes (**left**) and match them to edges in the query image. In the accumulated image, we increase the value of all pixels containing a synthetic silhouette, which contributed positively to the matching score. We run such an evaluation for every cell in the grid and sum up the accumulators to obtain one accumulated image for a query (**right**).

Camera model	#	Camera model	#
Canon DIGITAL IXUS 860 IS	253	Canon EOS 650D	2
Canon EOS 6D	112	Canon EOS DIGITAL REBEL XTi	2
COOLPIX L5	109	Canon EOS-1D Mark II N	2
Canon PowerShot G9	90	Canon EOS-1D Mark III	2
iPhone 5	89	Canon PowerShot A530	2
NIKON D300	76	Canon PowerShot S3 IS	2
iPhone 4	72	Canon PowerShot S60	2
NIKON D80	63	Canon PowerShot SX120 IS	2
NIKON D7000	60	Canon PowerShot SX200 IS	2
DMC-TZ5	59	Canon PowerShot SX260 HS	2
NIKON D90	59	DMC-FS10	2
Canon EOS 400D DIGITAL	54	DMC-FT1	2
Canon PowerShot D10	52	DMC-FZ28	2
EX-S600	46	DMC-FZ62	2
Canon DIGITAL IXUS 870 IS	45	DMC-GF1	2
NIKON D700	45	DMC-GF2	2
Canon EOS 450D	42	DMC-TS2	2
Canon EOS 7D	39	DMC-TZ4	2
SLT-A55V	37	DMC-TZ41	2
Canon DIGITAL IXUS 970 IS	36	DSLR-A700	2
SLT-A77V	34	DiMAGE A1	2
Canon EOS 60D	30	DiMAGE A2	2
Canon PowerShot G11	30	DiMAGE Z1	2
NIKON D800	30	Digimax V5 / Kenox V5	2
NIKON D3X	29	E5900	2
Canon EOS 350D DIGITAL	28	E7900	2
Canon PowerShot S95	27	EX-Z120	2
Canon EOS 5D Mark II	26	EX-Z5	2
DMC-TZ10	26	EX-Z700	2
NIKON D40	26	FinePix S2000HD	2
Canon EOS 40D	25	KODAK DX4530 ZOOM DIGITAL CAMERA	2
NIKON D5000	25	KODAK EASYSHARE V1273 DIGITAL CAMERA	2
NIKON D60	25	KODAK Z612 ZOOM DIGITAL CAMERA	2
DSC-RX100	23	NEX-5	2
Canon EOS 500D	22	NIKON D3200	2
DSLR-A290	22	NIKON D3S	2
Canon DIGITAL IXUS 40	21	PENTAX DL	2
Canon EOS 5D Mark III	21	PENTAX K-7	2
Canon PowerShot S100	20	PENTAX Optio VS20	2
iPhone 4S	20	TG-1	2
Canon PowerShot A710 IS	19	iPad	2
E-P3	18	iPhone	2
M9 Digital Camera	18	iPhone 3GS	2
NEX-7	18	Digimax U-CA 5, Kenox U-CA 5 / Kenox U-CA 50	1
Canon EOS 50D	17	KENOX S860 / Samsung S860	1
Canon EOS REBEL T3	17	C-5000Z	1
Canon PowerShot G10	17	C40Z,D40Z	1
PENTAX K100D	17	C720UZ	1
Canon DIGITAL IXUS 800 IS	15	COOLPIX AW110	1
Canon EOS 1000D	15	COOLPIX L1	1
Canon EOS 550D	15	COOLPIX L27	1
Canon PowerShot A640	15	COOLPIX P1	1
COOLPIX AW100	14	COOLPIX P300	1
KODAK EASYSHARE Z950 DIGITAL CAMERA	13	COOLPIX P510	1
NIKON D200	13	COOLPIX P7000	1
NIKON D800E	13	COOLPIX P7700	1
DMC-FZ38	12	COOLPIX P80	1
DiMAGE 7i	12	COOLPIX S10	1
SLT-A65V	12	COOLPIX S230	1
Canon EOS 20D	11	COOLPIX S500	1
DiMAGE A200	11	COOLPIX S610	1
DiMAGE Z5	11	Canon DIGITAL IXUS 330	1
DMC-LX3	10	Canon DIGITAL IXUS 500	1
DSLR-A550	10	Canon DIGITAL IXUS 700	1
E4500	10	Canon DIGITAL IXUS 750	1
NEX-3	10	Canon DIGITAL IXUS 90 IS	1
NIKON D3100	10	Canon EOS REBEL T2i	1
Canon EOS 30D	9	Canon EOS-1Ds Mark II	1
DMC-TZ35	9	Canon IXY DIGITAL 25 IS	1
DSLR-A350	9	Canon PowerShot A3200 IS	1
NIKON D5100	9	Canon PowerShot A4000 IS	1
NIKON D70	9	Canon PowerShot A470	1
PENTAX K200D	9	Canon PowerShot A590 IS	1
COOLPIX P520	8	Canon PowerShot A610	1
DSLR-A900	8	Canon PowerShot A700	1
PENTAX K-3	8	Canon PowerShot A80	1
PENTAX K10D	8	Canon PowerShot A95	1
PENTAX Optio 33L	8	Canon PowerShot G1 X	1
S1	8	Canon PowerShot G5	1
Canon EOS 300D DIGITAL	7	Canon PowerShot G7	1
Canon PowerShot G12	7	Canon PowerShot Pro1	1
DMC-FZ18	7	Canon PowerShot S30	1
DMC-TZ20	7	Canon PowerShot S50	1
DYNAX 7D	7	Canon PowerShot S80	1
NEX-6	7	Canon PowerShot SD1000	1
X-E1	7	Canon PowerShot SD700 IS	1
COOLPIX S9100	6	Canon PowerShot SX100 IS	1
Canon DIGITAL IXUS 55	6	Canon PowerShot SX220 HS	1
Canon EOS 600D	6	D40	1
Canon PowerShot A650 IS	6	D700	1
FinePix S5600	6	DMC-FX37	1
NIKON D4	6	DMC-FX40	1
NIKON D7100	6	DMC-FX8	1
Digimax S1000 / Kenox S1000	5	DMC-FZ30	1
COOLPIX L22	5	DMC-FZ5	1
COOLPIX P5000	5	DMC-FZ50	1
COOLPIX P6000	5	DMC-FZ7	1
COOLPIX S620	5	DMC-G2	1
Canon EOS 5D	5	DMC-LS75	1
Canon EOS DIGITAL REBEL XSi	5	DMC-LX5	1

Canon PowerShot A720 IS	5	DMC-TZ15	1
DMC-GH2	5	DMC-TZ3	1
DMC-TZ18	5	DMC-TZ7	1
DSLR-A500	5	DSC-W120	1
E-M5	5	DSLR-A230	1
PENTAX K-5	5	DSLR-A580	1
PENTAX Optio W20	5	DiMAGE G500	1
VSCOcam	5	DiMAGE X1	1
iPhone 5s	5	E-510	1
Canon DIGITAL IXUS 850 IS	4	E3100	1
Canon EOS REBEL T1i	4	E3500	1
Canon PowerShot S2 IS	4	E4600	1
DMC-FT3	4	E5200	1
DMC-FX01	4	EOS 40D	1
DSLR-A200	4	EX-FH20	1
FinePix F31fd	4	EX-H20G	1
FinePix2800ZOOM	4	EX-Z110	1
KODAK EASYSHARE C613 ZOOM DIGITAL CAMERA	4	EX-Z4	1
N97	4	EX-Z40	1
NIKON D300S	4	EX-Z55	1
NIKON D50	4	EX-Z60	1
NIKON D70s	4	EX-Z750	1
PENTAX DS	4	FinePix A800	1
SAMSUNG WB550, WB560 / VLUU WB550 / SAMSUNG HZ15W	4	FinePix F30	1
Digimax S830 / Kenox S830	3	FinePix F450	1
COOLPIX P5100	3	FinePix J150W	1
COOLPIX S4	3	FinePix S200EXR	1
Canon DIGITAL IXUS 65	3	FinePix S5000	1
Canon DIGITAL IXUS 950 IS	3	FinePix S6500fd	1
Canon EOS 1100D	3	FinePix S7000	1
Canon PowerShot A620	3	HP PhotoSmart C945 (V01.46)	1
Canon PowerShot A70	3	HP PhotoSmart R707 (V01.00)	1
Canon PowerShot S5 IS	3	KODAK CX7530 ZOOM DIGITAL CAMERA	1
Canon PowerShot SX230 HS	3	KODAK DX7440 ZOOM DIGITAL CAMERA	1
DC P500	3	KODAK EASYSHARE C195 Digital Camera	1
DMC-FX35	3	KODAK EASYSHARE ZD710 ZOOM DIGITAL CAMERA	1
DMC-FZ200	3	KODAK V610 DUAL LENS DIGITAL CAMERA	1
DMC-G3	3	Konica Digital Camera KD-400Z	1
DMC-TZ30	3	LEICA X1	1
DMC-TZ31	3	NEX-3N	1
DMC-TZ6	3	NIKON D100	1
DSLR-A100	3	NIKON D1X	1
DSLR-A300	3	NIKON D2Xs	1
E-PL3	3	NIKON D3000	1
EX-Z75	3	NV20, VLUU NV20	1
FinePix JZ500	3	PENTAX DL2	1
Hasselblad H3D	3	PENTAX K-r	1
KODAK Z740 ZOOM DIGITAL CAMERA	3	PENTAX K100D Super	1
NIKON D3	3	PENTAX Optio S4	1
NIKON D40X	3	PENTAX Optio S7	1
NIKON D600	3	PENTAX Optio WPi	1
PENTAX K-x	3	QV-R52	1
VR330,D730	3	SAMSUNG ES15 / VLUU ES15 / SAMSUNG SL30	1
WB2000	3	SAMSUNG ES55,ES57 / VLUU ES55 / SAMSUNG SL102	1
C750UZ	2	SAMSUNG ES74,ES75,ES78 / VLUU ES75,ES78	1
COOLPIX S9500	2	SAMSUNG WB850F/WB855F	1
Canon DIGITAL IXUS 50	2	SLT-A35	1
Canon DIGITAL IXUS 70	2	SLT-A57	1
Canon DIGITAL IXUS 80 IS	2	SLT-A99V	1
Canon DIGITAL IXUS 85 IS	2	SP560UZ	1
Canon EOS 100D	2	X-Pro1	1

Table B.3: Number of images per camera model in the *GeoPose3K* dataset.

References

[1] S. Bae, A. Agarwala, F. Durand, Computational rephotography, ACM Transactions on Graphics 29 (3) (2010) 1–15.

[2] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, D. Lischinski, Deep photo: model-based photograph enhancement and viewing, ACM Transactions on Graphics 27 (5) (2008) 1–10.

[3] J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, D. Sýkora, StyLit: Illumination-guided example-based stylization of 3D renderings, ACM Transactions on Graphics 35 (4).

[4] N. Snavely, S. M. Seitz, R. Szeliski, Photo tourism: Exploring Photo Collections in 3D, ACM Transactions on Graphics 25 (3) (2006) 835–846.

[5] C.-c. Hsieh, W.-h. Cheng, C.-h. Chang, Y.-y. Chuang, J.-l. Wu, Photo navigator, in: MULTIMEDIA '08: Proceeding of the 16th ACM international conference on Multimedia, ACM, New York, NY, USA, 2008, pp. 419–428.

[6] B. Epshtein, E. Ofek, Y. Wexler, P. Zhang, Hierarchical photo organization using geo-relevance, in: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems - GIS '07, New York, NY, USA, 2007, pp. 1–7.

[7] Z. Luo, H. Li, J. Tang, R. Hong, T.-S. Chua, ViewFocus: Explore Places of Interests on Google Maps Using Photos with View Direction Filtering, in: Proceedings of the 17th ACM International Conference on Multimedia, MM '09, ACM, New York, NY, USA, 2009, pp. 963–964.

[8] R. Talluri, J. Aggarwal, Position estimation for an autonomous mobile robot in an outdoor environment, IEEE Transactions on Robotics and Automation 8 (5) (1992) 573–584.

[9] R. Talluri, J. K. Aggarwal, Image Map Correspondence for Mobile Robot Self-Location Using Computer Graphics, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (6) (1993) 597–601.

[10] F. Stein, G. Medioni, Map-based localization using the panoramic horizon, in: IEEE Transactions on Robotics and Automation, Vol. 11, IEEE, New York, NY, USA, 1995, pp. 892–896.

[11] C. Prospero, Naval, M. Mukunoki, M. Minoh, K. Ikeda, Estimating Camera Position and Orientation from Geographical Map and Mountain Image, in: 38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers, 1997, pp. 9–16.

[12] P. C. Naval, Camera Pose Estimation by Alignment from a Single Mountain Image, International Symposium on Intelligent Robotic Systems (1998) 157–163.

[13] G. Baatz, O. Saurer, K. Köser, M. Pollefeys, Large scale visual geo-localization of images in mountainous terrain, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 517–530.

[14] R. I. Hammoud, S. A. KuzdebA, B. Berard, V. Tom, R. Ivey, R. Bostwick, J. Handuber, L. Vinciguerra, N. Shnidman, B. Smiley, Overhead-based image and video geo-localization framework, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Press, Washington, D.C., USA, 2013, pp. 320–327.

[15] E. Tzeng, A. Zhai, M. Clements, R. Townshend, A. Zakhor, User-Driven Geolocation of Untagged Desert Imagery Using Digital Elevation Models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 237–244.

[16] Y. Chen, G. Qian, K. Gunda, H. Gupta, K. Shafique, Camera geolocation from mountain images, in: 2015 18th International Conference on Information Fusion, IEEE, New York, NY, USA, 2015, pp. 1587–1596.

[17] O. Saurer, G. Baatz, K. Köser, L. Ladický, M. Pollefeys, Image Based Geo-localization in the Alps, in: International Journal of Computer Vision, Springer US, 2015, pp. 1–13.

[18] L. Porzi, S. R. Buló, P. Valigi, O. Lanz, E. Ricci, Learning Contours for Automatic Annotations of Mountains Pictures on a Smartphone, in: Proceedings of the International Conference on Distributed Smart Cameras, ACM, New York, NY, USA, 2014, pp. 13:1–13:6.

[19] L. Baboud, M. Čadík, E. Eisemann, H.-P. Seidel, Automatic Photo-to-terrain Alignment for the Annotation of Mountain Pictures, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Washington, D.C., USA, 2011, pp. 41–48.

[20] R. Ji, Y. Gao, W. Liu, X. Xie, Q. Tian, X. Li, When Location Meets Social Multimedia: A Survey on Vision-Based Recognition and Mining for Geo-Social Multimedia Analytics, ACM Transactions on Intelligent Systems and Technology 6 (1) (2015) 1:1—-1:18.

[21] D. Robertsone, R. Cipolla, An image-based system for urban navigation, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2004, pp. 84.1–84.10.

[22] W. Zhang, J. Košecká, Image Based Localization in Urban Environments, in: Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), IEEE, 2006, pp. 33–40.

[23] G. Schindler, M. Brown, R. Szeliski, City-scale location recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Washington, D.C., USA, 2007, pp. 1–7.

[24] E. Johns, G. Z. Yang, From images to scenes: Compressing an image cluster into a single scene model for place recognition, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, New York, NY, USA, 2011, pp. 874–881.

[25] A. R. Zamir, M. Shah, Accurate image localization based on google maps street view, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 255–268.

[26] A. R. Zamir, M. Shah, Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (8) (2014) 1546–1558.

[27] A. R. Zamir, S. Ardeshir, M. Shah, GPS-tag refinement using random walks with an adaptive damping factor, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Washington, D.C., USA, 2014, pp. 4280–4287.

[28] D. Mishkin, M. Perdoch, J. Matas, Place Recognition with WxBS Retrieval, in: CVPR 2015 Workshop on Visual Place Recognition in Changing Environments, 2015.

[29] A. Kendall, R. Cipolla, Modelling Uncertainty in Deep Learning for Camera Relocalization, in: Proceedings of the International Conference on Robotics and Automation (ICRA), IEEE, New York, NY, USA, 2016, pp. 4762–4769.

[30] T. Weyand, I. Kostrikov, J. Philbin, Planet - photo geolocation with convolutional neural networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII, Springer International Publishing, Cham, 2016, pp. 37–55.

[31] J. Hays, A. A. Efros, IM2GPS: Estimating geographic information from a single image, in: Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, IEEE, New York, NY, USA, 2008, pp. 1–8.

[32] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, L.-J. Li, YFCC100M: The New Data in Multimedia Research, in: Communications of the ACM, Vol. 59, ACM, New York, NY, USA, 2016, pp. 64–73.

[33] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, R. Grzeszczuk, City-scale landmark identification on

mobile devices, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Washington, D.C., USA, 2011, pp. 737–744.

[34] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Washington, D.C., USA, 2016, pp. 5297–5307.

[35] M. Čadík, J. Vašíček, M. Hradiš, F. Radenović, O. Chum, Camera elevation estimation from a single mountain landscape photograph, in: M. W. J. Xianghua Xie, G. K. L. Tam (Eds.), Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, 2015, pp. 30.1–30.12.

[36] M. Firman, RGBD Datasets: Past, Present and Future, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2016, pp. 661–673.

[37] A. Saxena, S. H. Chung, A. Y. Ng, Learning Depth from Single Monocular Images, Advances in Neural Information Processing Systems 18 (2006) 1161–1168.

[38] A. Saxena, M. Sun, A. Y. Ng, Make3D: Learning 3D scene structure from a single still image, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (5) (2009) 824–840.

[39] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset, The International Journal of Robotics Research 32 (11) (2013) 1231–1237.

[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.

[41] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD images, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 7576 LNCS, 2012, pp. 746–760.

[42] R. Mottaghi, X. Chen, X. Liu, N. G. Cho, S. W. Lee, S. Fidler, R. Urtasun, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 891–898.

[43] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V, Springer International Publishing, Cham, 2014, pp. 740–755.

[44] G. Baatz, O. Saurer, K. Köser, M. Pollefeys, Leveraging Topographic Maps for Image to Terrain Alignment, in: 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, IEEE, New York, NY, USA, 2012, pp. 487–492.

[45] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, Cambridge, UK, 2004.

[46] M. M. Haklay, P. Weber, Openstreetmap: User-generated street maps, Vol. 7, IEEE Educational Activities Department, Piscataway, NJ, USA, 2008, pp. 12–18.

[47] L. Ladicky, C. Russell, P. Kohli, P. H. S. Torr, Graph cut based inference with co-occurrence statistics, in: Proceedings of the 11th European Conference on Computer Vision: Part V, ECCV'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 239–253.

[48] T. Ahmad, G. Bebis, M. Nicolescu, A. Nefian, T. Fong, An Edge-Less Approach to Horizon Line Detection, in: 14th IEEE International Conference on Machine Learning and Applications (ICMLA'15), Miami, Florida, USA, December 9-11, 2015., IEEE, 2015.

[49] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, 2015, pp. 3431–3440.

[50] K. He, J. Sun, X. Tang, Single image haze removal using dark channel prior, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (12) (2011) 2341–2353.

[51] R. Fattal, Dehazing using color-lines, in: ACM Transactions on Graphics, Vol. 34, ACM, New York, NY, USA, 2014, pp. 13:1–13:14.

[52] J.-F. Lalonde, S. G. Narasimhan, A. A. Efros, What do the sun and the sky tell us about the camera?, Vol. 88, Springer US, 2010, pp. 24–51.

[53] P. Dollár, C. Zitnick, Fast Edge Detection Using Structured Forests, in: Proceedings of the 2013 IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, 2014, pp. 1841–1848.

[54] M. Ruzon, C. Tomasi, Color edge detection with the compass operator, in: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, IEEE Computer Society Press, Washington, D.C., USA, 1999, pp. 160–166.