Camera Orientation Estimation in Natural Scenes Using Semantic Cues

Jan Brejcha[1] Martin Čadík[2]
CPhoto@FIT, Faculty of Information Technology, Brno University of Technology
[1]ibrejcha@fit.vutbr.cz, [2]cadik@fit.vutbr.cz

Abstract

Camera orientation estimation in natural scenes has recently been approached by several methods, which rely mainly on matching a single modality – edges or horizon lines with 3D digital elevation models. In contrast to previous works, our new image to model matching scheme is based on a fusion of multiple modalities and is designed to be naturally extensible with different cues. In this paper, we use semantic segments and edges. To our knowledge, we are the first to consider using semantic segments jointly with edges for alignment with digital elevation model. We show that high-level features, such as semantic segments, complement the low-level edge information and together help to estimate the camera orientation more robustly compared to methods relying solely on edges or horizon lines. In a series of experiments, we show that segment boundaries tend to be imprecise and important information for matching is encoded in the segment area and a coarse shape. Intuitively, semantic segments encode low frequency information as opposed to edges, which encode high frequencies. Our experiments exhibit that semantic segments and edges are complementary, improving camera orientation estimation reliability when used together. We demonstrate that our method combining semantic and edge features is able to reach state-of-the-art performance on three datasets.

1. Introduction

Camera orientation estimation has recently been approached by a variety of works [25, 24, 7, 18, 6, 27, 26, 13]. With the knowledge of orientation and position of camera in the world, we can infer answers to questions such as: "Is it possible to move forward?", or "What are we looking at?" While state-of-the-art data-driven methods [17, 2] can answer such questions, they are focused mainly on urban areas. In contrast, this work focuses on camera orientation estimation in mountainous areas, which are important as well. Knowledge of camera orientation may be valuable for scene understanding and organizing large databases of photographs. Furthermore, camera orientation may augment other sensors in robots, UAV's or helicopters for automatic navigation. Several works on camera orientation estimation in mountainous areas were developed recently [7, 6, 26]. However, the problem still remains challenging for real world images as illustrated by our experiments.

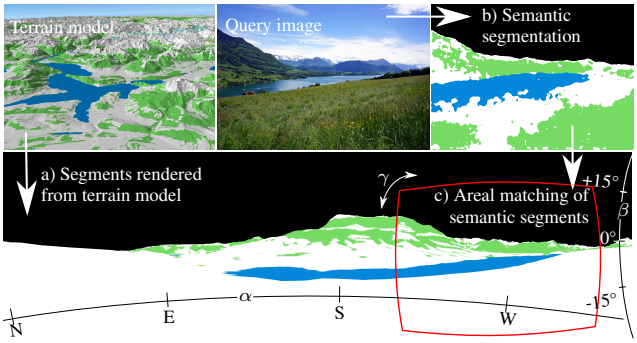

Figure 1. Overview of the proposed method. (a) Synthetic semantic segments are rendered using terrain model and geospatial database. (b) Query image is segmented via semantic segmentation method. (c) Semantic segments from query image are aligned with synthetic semantic segments and camera orientation (α, β, γ) is recovered.

Evolution of handheld devices brought the possibility to recover orientation from inertial sensors. However, such orientation is usually inaccurate and vulnerable to drift. Camera position, on the other hand, is often stored more accurately as a GPS coordinate. Similarly, the majority of images and videos on the internet contain no information about camera orientation. The knowledge of accurate camera orientation opens up interesting applications and facilitates difficult image recognition tasks. For example, images with known camera pose can be augmented with information from geospatial databases and used in augmented and virtual reality applications. Existing solutions to camera orientation estimation in mountainous scenes rely on matching a query image with a terrain model [7, 6, 26]. In general, these methods are based on aligning query image features (edge maps) with synthetic edges generated from the terrain model. While we also use a terrain model as a reference, we do not rely solely on the edge information. In contrast to previous works, we align areal features which complement edge information. Specifically, development

2475-7888/18/$31.00 ©2018 IEEE
DOI 10.1109/3DV.2018.00033

of semantic segmentation allows us to employ matching based on semantic segments. We map terrain features, such as forests, bodies of water, and glaciers from a geospatial (GIS) database to a digital elevation model (DEM) and render into a panorama image containing semantic segments (Fig. 1(a)). From the query image, we extract semantic segments (Fig. 1(b)) using recent semantic segmentation methods [20, 22, 11]. To estimate camera orientation, we match the query and the panorama (Fig. 1(c)). We estimate a correspondence between the query image and the synthetic panorama based on similarity of semantic segments of the same class. Intuitively, spatial relationships between different semantic classes disambiguate in-plane rotations. In order to exploit these spatial relationships, we introduce confidence fusion (**CF**), which prefers camera orientations with highest confidence agreement across all semantic classes. The benefit of the proposed technique is the possibility to naturally fuse confidence estimates of different modalities, such as different segment classes and edge maps.

Contributions We propose a novel method for aligning a single image to a digital terrain model. To our knowledge, we are the first to consider joint combination of semantic segments and edges to match an image with a rendered panorama of the terrain. We train semantic segmentation on a synthetically rendered dataset and show that synthetic data is needed to achieve reasonable accuracies when used for orientation estimation in mountainous environment. To enable matching of several semantic segment classes and an edge map with the rendered panorama, we propose a novel confidence fusion (**CF**) method which fuses individual beliefs together to achieve better accuracy. Our experiments show that the proposed method outperforms state-of-the-art on publicly available test sets – GeoPose3K [9], Venturi Mountain dataset [26], and CH1 dataset [29].

2. Related work

Explosion of publicly available photographic data in recent years allowed researchers to develop data-driven camera pose estimation methods, especially using Structure-from-Motion (SfM) techniques [16, 21, 31, 23, 17]. Unfortunately, such methods require an abundance of overlapping images to compute camera pose making them difficult to use in the natural environment, where the image coverage is still sparse. Another problem for natural scenes resides in the difficulty of finding stable key-points under changing appearance – illumination, weather, seasons, vegetation, *etc*.

Works dealing with natural scenes have shown that the horizon line is an important and relatively stable feature for camera orientation and position estimation [15, 32, 12, 29]. However, relying solely on the horizon line can be misleading, since there are many situations, when the horizon line is ill-defined, non-descriptive or completely invisible: (i) view from an elevated place to a flat landscape implies a flat horizon line, (ii) horizon line is contaminated with foreground objects, like trees, (iii) horizon line is not visible due to camera pitch (images without the sky).

Recent works dealing with the camera orientation estimation with fixed position for outdoor and mountainous scenes are based on alignment of a query image with a terrain model [8, 7, 28, 6, 26]. For alignment, edge maps were used by Baboud *et al*. [7] and Porzi *et al*. [26]. Produit *et al*. [28] used pixel patches located at corners of salient edges. Most closely to our work, Baatz *et al*. [6] used semantic segments for the image alignment. They extracted binary descriptors capturing the spatial relationships between different classes of segments. However, the descriptors encode local changes between neighboring segments, meaning that only segment boundaries are exploited by this technique. The boundaries are usually inaccurate for real world cases (see Fig. 1(b)), rendering the method unstable. To address this issue, we propose areal matching of semantic segments. The main idea is that segment areas should match well, unlike segment boundaries which are potentially wrong.

Several approaches for camera position and orientation estimation based on semantic segments were also developed for urban environments. Senlet *et al*. [30] and Castaldo *et al*. [10] used semantic segments for matching an input image with a GIS map to estimate a camera position, but their approach is unable to recover the camera orientation precisely. Ardeshir *et al*. [3], on the other hand, used estimated superpixels and semantic segments projected from a GIS database to infer a geo-semantic segmentation. Their iterative algorithm fine-tunes the camera position and orientation, and produces semantic segmentation based on projections from the GIS database. Their work assumes matching buildings with known height, which reduces description of segment areas to two points (leftmost and rightmost). This renders the method inappropriate for natural scenes, where the segment shapes are more complicated and cannot be simplified in this way. Armagan *et al*. [4] proposed an iterative approach to fine-tune camera position and orientation based on semantic segmentation with known camera position and orientation estimate. In contrast to their work, our approach is more general as it does not need *any* initial camera orientation estimate.

3. Orientation estimation using semantic cues

For a given query image, our aim is to estimate its camera orientation using a digital terrain model. The basic idea is to project the query image onto the sphere and align it with the spherical panorama rendered from the model. The correct alignment then defines the searched camera orientation. We assume that the position λ (latitude, longitude, elevation) and the horizontal field-of-view p_f of the query

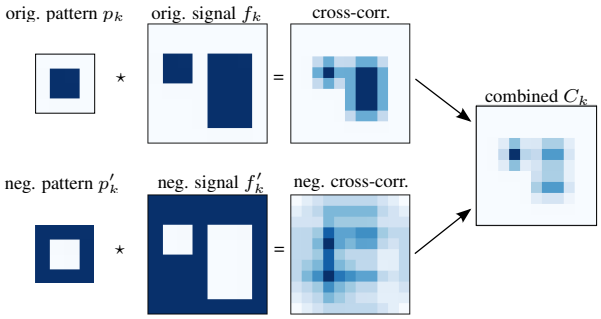

Figure 2. Illustration of the cross-correlation behavior for two functions $p_k > 0$ and $f_k > 0$, which are, without loss of generality, defined on $\mathbb{R}^2$ for this example. White color denotes $\epsilon \to 0^+$, darker color denotes a higher value. In the first line, the cross-correlation is maximized even for translations, where surroundings of the pattern are not in agreement with the signal. The inverted pattern and signal on the second line create a complementary cross-correlation map. When the two cross-correlations are combined, the maximum value is correctly in place where both the pattern and its surroundings overlap the largest areas.

image p are known. The goal is to find a rotation $g \in SO(3)$ of the camera frame with respect to the frame of the digital terrain f. The terrain model is rendered with synthetic semantic segments as a spherical $360° \times 180°$ panorama (see Fig. 1(a)), with λ as the unit sphere center. A projective query image containing estimated semantic segments is projected on the unit sphere as well. The query image is scaled to cover the part of the unit sphere corresponding to its field-of-view. The image is scaled by a factor $s_q = \frac{p_f}{2\pi w_q}$, where w_q is the width of the query image.

3.1. Cross-correlation as a measure of confidence

To estimate the camera orientation $g = (\alpha, \beta, \gamma)$, we compute a matching confidence $C(\alpha, \beta, \gamma)$ over all possible combinations of rotations $\alpha \in \langle 0°, 360° \rangle$, $\beta \in \langle 0°, 180° \rangle$, $\gamma \in \langle 0°, 360° \rangle$ (see Fig. 1(a) for respective rotations). We also define a confidence $C_k > 0$ for semantic segment class k and later fuse all confidences into the total confidence C. The combination of parameters maximizing the total confidence defines the camera orientation estimate $g = \arg\max_{\alpha,\beta,\gamma}(C(\alpha, \beta, \gamma))$.

We propose the confidence C_k to be a cross-correlation of the query and panorama on $SO(3)$, both containing semantic segments of class k. Similarly to Baboud *et al.* [7], we exploit the cross-correlation theorem for efficient computation of cross-correlation in the Fourier domain. Cross-correlation of two real valued functions f and p on $SO(3)$ is similar to ordinary 2D cross-correlation, but we are integrating over all positions on a sphere (S^2):

$$\forall g \in SO(3) : f \star p(g) = \int_{S^2} f(\omega) p(g^{-1}\omega) d\omega. \quad (1)$$

For each class k, we construct two spherical functions p_k (query segments) and f_k (synthetic segments) as follows. In order to obtain strictly positive confidence, we need the spherical functions to be strictly positive as well. We sample both query segments and synthetic segments of class k on a unit sphere, where 1 is assigned to pixels containing the segment of class k and $\epsilon \to 0^+$ to pixels that contain other segment classes, where ϵ is a small positive constant. However, calculating cross-correlation for a single segment class k using p_k and f_k may not be sufficient for correct alignment (see the top line in Fig. 2). In this case, the cross-correlation is maximized for all rotations, where $p_k(g) \leq f_k$. This way, segments from the query image tend to "hide" inside larger synthetic segments of the panorama image. In other words, there are large areas with the maximum cross-correlation value. To alleviate this problem, we divide the computation of class confidence C_k into two steps that are combined together, as illustrated in Fig. 2. The first step is the cross correlation $\forall g \in SO(3) : f_k \star p_k(g)$, given the class k. The second step is a complementary cross correlation with inverted spherical functions $f'_k = 1 + \epsilon - f_k$, $p'_k = 1 + \epsilon - p_k$. The combined cross-correlation, which equals to class confidence C_k across all rotations $g \in SO(3)$ is then calculated as:

$$\forall g \in SO(3) : C_k(g) = (f_k \star p_k(g))(f'_k \star p_k(g)'). \quad (2)$$

Intuitively, the first cross-correlation maximizes rotations where query segments overlap the synthetic segments, while the second cross-correlation maximizes rotations where the surroundings of query segments overlap the surroundings of the synthetic segments. By multiplying the two cross-correlation results, we robustly enforce rotations where overlap of both the segment area and its surroundings are maximized.

Please note, that the two-step cross-correlation is necessary and cannot be replaced by 1 and -1 encoding for the segment and the background, respectively. Consider the situation in Fig. 3, where our two-step correlation is com-

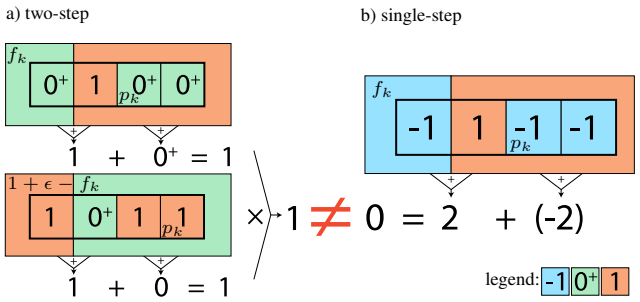

Figure 3. Two-pass cross-correlation is not equivalent to a single-pass using negative values. Our two-pass approach calculates number of correct pixels and disregards the wrong pixels if *both* foreground and background is matched. In contrast, the single-pass penalizes the wrong pixels, which leads to result incompatible with our definition of confidence.

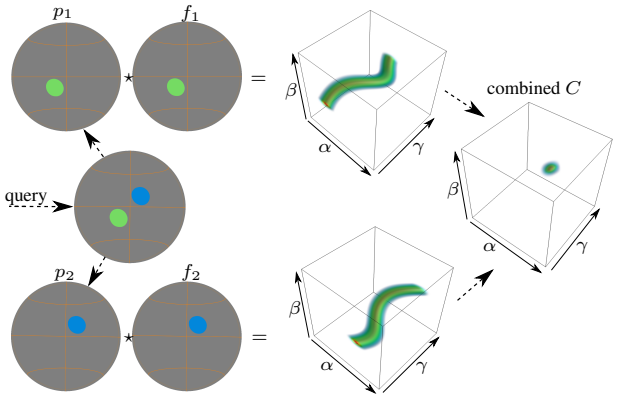

Figure 4. Synthetic experiment illustrating the confidence fusion. Cross-correlations are visualized as a heatmap over orientations (α, β, γ), which form a cube. The query image contains two circles, each circle represents one semantic segment (classes of the segments are different). In this case, cross-correlation of a single segment class does not disambiguate the roll angle (γ). On contrary, the fusion of confidence maps maximizes at a single orientation, as visualized in the rightmost cube.

pared to a single-step version. The leftmost pixel is matching background, the second pixel is matching foreground, and two pixels on the right do not match (background on foreground). Since two of four pixels match foreground or background, we expect the confidence to be greater than ϵ. Our two-step approach maximizes correct overlap of segments (and returns 1) while the single-step method is biased by non-matching regions (and returns 0).

3.2. Confidence fusion

So far we have considered confidence of a single segment class. A single segment class k is usually not sufficiently descriptive to constrain the correct rotation, since the semantic segment areas are often similar to each other for many rotations. Mutual spatial relationships between different segment classes help to disambiguate the correct rotation. While a single segment class does not disambiguate the roll angle (see Fig. 4), the combination of two segments, gives a single precise maximum, which is located at the desired rotation (see Fig. 4, combined C).

With the assumption that the segments are correctly detected in the query image and no segments are missing from the rendered panorama, the correct rotation would be determined by the highest confidence across *all* fused classes. To calculate it, we would simply calculate the product of confidences across all classes:

$$\forall g \in SO(3) : C(g) = \prod_k (C_k(g)). \tag{3}$$

However, the assumption of correct detection and complete model cannot be fully satisfied in real-world applications. In this case, the wrongly detected segment could cause drift from the correct solution. To be able to compensate mistakes in the detection or missing parts in the model, we propose to compute the Confidence Fusion (**CF** framework) as a weighted geometric mean:

$$\forall g \in SO(3) : C(g) = \prod_k (C_k(g))^{w_k}. \tag{4}$$

The importance of the segment class k can now be tuned by the weight $w_k \in \langle 0, 1 \rangle$: the weight should be small for wrongly detected segment classes and high for classes that are detected and rendered correctly.

The weights can be estimated in many ways. We tried to regress them directly based on the GeoPose3K training set, but this approach has not proved to be robust across different datasets. The robust estimation of the weights for fusion of multiple densities can be borrowed from Ajgl and Šimandl [1] (Theorem 2), where the authors derive the computation of weights in the sense of minimization of maximal Kullback-Leibler divergence between the fused confidence C and the class confidences C_k. The method needs to be used carefully in order to keep the computational complexity reasonably low and to allow suppression of wrongly detected class confidences (supplementary material, Sec. 1).

Moreover, expensive calculation of weights can be avoided completely. We observed, that class confidences C_k are potentially incorrect for segments covering small areas, as small segments may be often mis-detected, or occluded. We solved this problem by setting the weights empirically. If the area covered by the segment class k in the query or panorama image is lower than a threshold t[1], we simply turn off the segment class k by setting its value $w_k = 0$. For remaining segment classes, we set $w_k = 1$. This simple approach significantly outperforms non-weighted fusion (eq. 3) and provides comparable results to the approach of Ajgl and Šimandl [1] in our application.

Semantic segments and edge features. Our Confidence Fusion framework (**CF**, eq. 4) is in fact able to use any non-negative result based on spherical cross-correlation, it is not limited to semantic segments only. Majority of methods employs edge features for matching real image with a terrain model [7, 5, 29, 32, 12, 14]. Our goal is to show that it is highly beneficial to combine edge features with other cues, such as the semantic segments. To detect edges, we use edge detector trained to estimate silhouette edges similar to the rendered ones [9]. To calculate confidence based on edge features, we use cross correlation metric developed exclusively for edges, VCC-2011 [7], for which we replace negative values with $\epsilon \to 0^+$.

3.3. Semantic segmentation

For the task of matching a query image with rendered semantic segments, we need a segmentation method to es-

[1] We use $t = 0.1\%$ of the total image area (found experimentally).

	DeepLab-v2 VGG16		DeepLab-v2 VGG16 + CRF		FCN8s SiftFlow		FCN8s Pascal-Context		ALE		Naive baseline	
mACC	**0.63**		0.62		0.59		0.54		0.61		0.20	
mIU	**0.53**		0.52		0.46		0.38		0.46		0.07	
	IU	**ACC**	**IU**	**ACC**	**IU**	**ACC**	**IU**	**ACC**	**IU**	**ACC**	**IU**	**ACC**
mountain	**0.60**	0.78	**0.60**	**0.79**	0.56	0.77	0.44	0.60	0.49	0.60	0.00	0.00
sky	**0.89**	**0.93**	**0.89**	**0.93**	**0.89**	**0.93**	0.82	0.89	0.79	0.91	0.35	1.00
forest	**0.38**	0.53	0.37	0.52	0.34	0.48	0.32	**0.57**	0.33	0.56	0.00	0.00
water	**0.44**	**0.55**	**0.44**	0.54	0.31	0.51	0.17	0.43	0.30	0.47	0.00	0.00
glacier	0.36	0.37	0.31	0.32	0.21	0.24	0.14	0.19	**0.40**	**0.49**	0.00	0.00

Table 1. Results of semantic segmentation methods trained with GeoPose3K. Results are measured on GeoPose3K test set; accuracy (*ACC*) and intersection over union (*IU*) are measured per class independently, mean pixel accuracy over all classes is denoted by *mACC*, and mean intersection over union over all classes is denoted as *mIU*. Last column represents a naive segmentation into a single class (sky), which has the largest *prior* probability in the GeoPose3K dataset.

timate semantic segments which are visually similar to the rendered counterparts. To achieve this, we fine-tune several state-of-the-art semantic segmentation models. Please note, that the fine-tuning using the synthetic dataset is a crucial step in the whole **CF** framework and it is one of the contributions of this paper.

We consider two state-of-the-art convolutional neural network (CNN) architectures: FCN [22] and Deeplabv2-VGG-16 [11], and one non-CNN method which is used as a reference: Automatic Labeling Environment [20]. We start with SiftFlow and Pascal-Context models for FCN8s, and similarly for training DeepLab-v2, we use VGG-16 as an initial model. All models were fine-tuned on GeoPose3K dataset [9], which contains synthetic semantic labels for more than 3K images registered into the 3D terrain model. We split GeoPose3K into train (1927 images), validation (472 images), and test sets (516 images), so that these three sets are geographically disjoint (see supplementary material Fig. 1). This way we ensure, there are no similar images across the train, validation and test sets. Furthermore, we optimize the geographical distribution of images so that the sets contain similar amount of semantic segments per class (measured in pixels). The train/validation/test splits are available in the supplementary material.

The GeoPose3K dataset contains in total 14 classes for semantic segmentation, including sky. Unfortunately, many segment classes, such as sinkhole or bare-rock are available only for a limited subset of images. Segments of these classes often span a small area of the image which reduces their descriptivity. Motivated by this observation, we selected the following subset of semantic segment classes, which cover a sufficient number of images: *mountain*, *sky*, *forest*, *bodies of water*, and *glacier*. To fine-tune these classes using FCN8s and DeepLab-v2, we replaced the last classification neural network layer with a layer containing our own 5 classes.

4. Experiments

In this section we provide an in-depth evaluation of the proposed camera orientation estimation. For evaluation, we used three publicly available data sets – GeoPose3K test set (516 test photos), CH1 dataset (203 photos) [29], and Venturi Mountain Dataset [26] (12 videos). The original CH1 dataset [29] does not contain camera orientation ground truths. However, the GeoPose3K contains images from CH1 dataset and provides camera orientation ground truth [9]. As well as the CH1 dataset, we held out the GeoPose3K test set and Venturi dataset from semantic segmentation training, and used it only for testing. The presented evaluation is by far the largest analysis of methods dealing with camera orientation estimation in natural environment without any help of device sensors (compass, accelerometer, gyroscope). We compare our work directly with Baboud *et al*. [7], Porzi *et al*. [26], and our implementation of Saurer *et al*. [29]. Since Baatz *et al*. [6] and previous methods [25, 24, 28] use rather a limited number of private images for their evaluation, we were unable to establish direct comparison with these methods.

Evaluation metric. To be able to compare our approach with the recent work of Porzi *et al*. [26], we are evaluating the estimated camera orientation accuracy using the same *orientation estimation error* defined as:

$$e(\mathbf{R}_{gt}, \mathbf{R}_c) = \arccos\left(\frac{\mathrm{tr}\left[\mathbf{R}_{gt}^{\intercal}\mathbf{R}_c\right] - 1}{2}\right), \qquad (5)$$

where $\mathbf{R}_{gt}$ is the ground truth camera rotation matrix and $\mathbf{R}_c$ is the estimated rotation matrix. This metric calculates the magnitude of the smallest rotation between the ground truth and the estimated rotation. We calculate and plot a cumulative distribution of the orientation error, where certain fractions of images have the orientation error equal or lower than given threshold. A random baseline illustrates what is the probability of guessing an orientation. For better clarity, we also give a measure of Area Under Curve (AUC), where AUC = 1 is in theory the best possible result.

4.1. Evaluation of semantic segmentation methods

We select a semantic segmentation method for our orientation estimation framework using standard semantic segmentation metrics, namely *mean accuracy* and *mean Intersection over Union (mIU)*. These metrics, shown in Tab. 1,

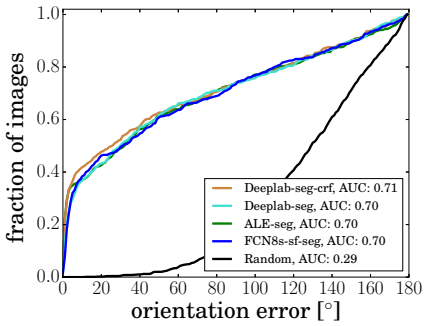

Figure 5. Performance of **CF** framework with different segmentation methods. The best – Deeplab with CRF (AUC: 0.71); other methods scored similarly (AUC: 0.70).

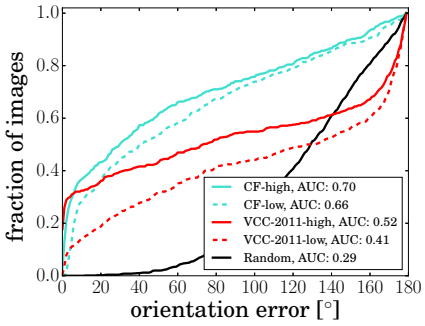

Figure 6. Comparison our **CF** framework using semantic segments with edge-based VCC-2011 [7] on high (solid curves) and low (dashed curves) resolutions.

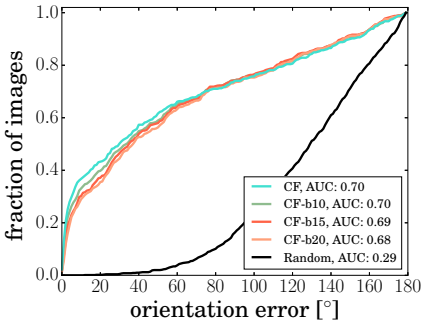

Figure 7. Original semantic segments were smoothed by gaussian blur using three different kernel radii 10 px (b10), 15 px (b15) and 20 px (b20).

illustrate, that both Deeplab with and without Conditional Random Fields (CRF) are the methods of choice. Since the metrics are based on the ratio of correctly classified pixels, we expect that the best method based on these metrics is also the best for our camera orientation estimation framework. This expectation was verified by testing our orientation estimation framework with the semantic segmentation methods listed in Tab. 1. The results of this experiment are shown in Fig. 5. The best result was achieved by Deeplab with CRF (Deeplab-seg-crf, AUC: 0.71), but other segmentation methods – Deeplab without CRF, FCN8s, and ALE scored almost the same (AUC: 0.70). According to visual inspection, CNNs are slightly more successful in ignoring objects not present in the digital terrain model (see supplementary material, Fig. 2(f) vs. Fig. 2(g)). We use DeepLab for all following experiments with our Confidence Fusion (**CF**) framework.

4.2. The impact of cross-correlation resolution

For calculating cross-correlation in $SO(3)$ using Fourier transform (FFT), we use publicly available SOFT package [19]. Precision of the cross-correlation as well as computation time and memory footprint are driven by two factors – the input resolution of the spherical functions and resolution of the cross-correlation output. Higher input resolution implies more precise sampling of input spherical functions. Resolution of the output drives sampling of the resulting cross correlation. Please, note that lower input and output resolutions do not restrict the search space to any particular orientation – full 3D rotation is searched no matter what resolutions are selected.

In general, it is expected that lower resolution (coarser sampling) of the functions would decrease the precision of the method. Intuitively, coarser sampling might negatively affect high-frequency functions more than low-frequency functions. Semantic segments encode low frequency information, while edge features encode mainly high frequencies. According to this observation, we expect that using lower input and output resolution affects the precision of cross-correlation of semantic segments much less than cross-correlation of edges. To verify this hypothesis, we run an experiment to compare the effect of input and output resolutions on the achieved accuracy (see Fig. 6). We consider two versions of input and output resolution. The first version is a low resolution, with input resolution of 1024 samples and the output resolution of 128 samples (see dashed curves in Fig. 6). Low resolution yields fast evaluation (about 1.5 second per cross-correlation) and the orientation estimation of a single query lasts at most 30 seconds (depending on the number of segment classes). However, the result confidence is stored in a cube of size $(128)^3$, yielding almost 3° per bin, which may increase the orientation error. The second version is a high resolution one, where the input resolution is set to 4096 and the output to 512 samples (see solid curves in Fig. 6). The experiment confirmed our expectation, that using lower resolution for cross-correlating semantic segments does not increase the orientation error dramatically (see cyan solid, vs. cyan dashed curve in Fig. 6). Using lower resolution the time and memory footprint is reduced extensively (from 45 seconds per cross-correlation to just 1.5 second, and from 12GB of memory to just 247MB on high and low resolution, respectively). Compared to semantic segments, the edges contain higher frequencies, which are more affected by subsampling. In the case of edge-based cross-correlation (VCC-2011 [7]), the high resolution variant brings a decent improvement in terms of accuracy over the low resolution (see Fig. 6 red solid vs. red dashed curve). This result is in agreement with our expectations as well. The relative indifference to subsampling is an advantage of using segments over the edges.

4.3. Importance of segment boundaries

To ensure that our approach factually does not boil down to matching boundary edges of semantic segments, we conducted an experiment in which we suppressed the impor-

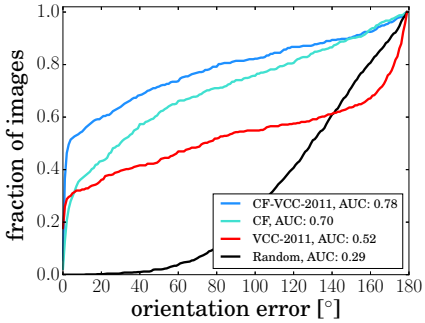

Figure 8. Comparison of the edge-based VCC-2011 [7], our **CF** framework using semantic segments, and combination of both approaches. We use our **CF** framework to fuse semantic segments and edges (**CF-VCC-2011**), which gives the best result.

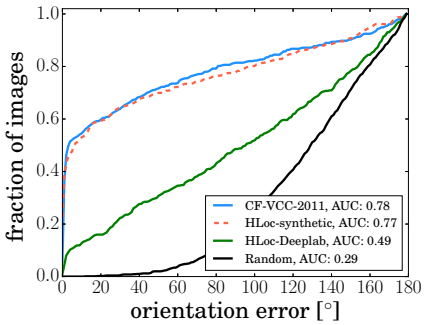

Figure 9. Our **CF** compared to **HLoc** [29] on GeoPose3K test set. **CF** (blue) using automatic segmentation and **HLoc** using synthetic sky segmentation (dashed) perform similarly; **HLoc** using automatic sky segmentation (green) performs worse.

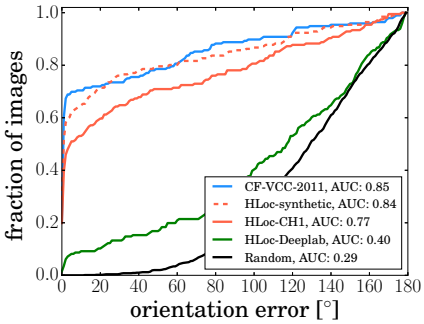

Figure 10. Results on CH1 dataset. Our **CF** framework using *automatic* segments + edges has superior accuracy compared to **HLoc** with original, *manually* refined horizon line from CH1 dataset (**HLoc-CH1**).

tance of segment boundaries by gaussian blur. We blurred the original query and synthetic segments with three different kernel radii – 10 px (0.43°), 15 px (0.65°) and 20 px (0.86°). This removes hard boundaries of semantic segments and reduces their impact. Since the boundaries of segment areas tend to be imprecise, we expect that suppressing their importance should not negatively affect the result. The achieved performance is shown in Fig. 7. The best performance was achieved using non-blurred and 10 px kernel radius (AUC: 0.70). For larger kernel radii the accuracy dropped only slightly, having AUC: 0.69 and 0.68 in case of 15 px and 20 px radius, respectively. We see that the blur does not affect the results significantly. This illustrates that potentially inaccurate segment boundaries are not very informative for camera orientation estimation using our **CF** method and the main information resides in segment areas and coarse shapes.

4.4. Are edges and semantic areas complementary?

The previous experiment suggests that segment areas encode the main information unlike the segment boundaries. Intuitively, segment areas correspond to low-frequency information, while edge features encode high-frequencies. This property should allow *combining* both types of features to increase orientation estimation accuracy. We calculate two confidences: one using VCC-2011, and the second one using semantic segments. The final result is obtained by fusion of both confidences with our **CF** framework. Since VCC-2011 penalizes query and silhouette edge crossings, the result of VCC-2011 may contain negative values. To be able to use the VCC-2011 result in our **CF** framework, we clamp negative values with $\epsilon \to 0^+$ before fusion.

Our expectation that the combination of edges and segments improves the orientation estimation result was confirmed in the following experiment. We used the GeoPose3K test set to measure the orientation error of VCC-2011 [7] (Fig. 8 – red curve, AUC: 0.52). Cyan curve in Fig. 8 denotes the result obtained by our **CF** framework using semantic segments only – AUC: 0.70. We can see that our method using semantic segments yields better performance, than edge-based VCC-2011. The combination of both (VCC-2011 and segments) using our **CF** framework, scored the best performance – (Fig. 8 – blue curve, AUC: 0.78). The difference between using edges and semantic segments is 18%. Furthermore, the combined result brings improvement of 26% over the VCC-2011. Similar results were recorded also on the Venturi Mountain dataset [26], and the CH1 dataset [29], but were omitted for the sake of readability from the main paper (see Fig. 3 and Tab. 1 in the supplementary material). We conclude, that according to this experiment, the semantic and edge features are complementary. Combining both approaches improves the camera orientation performance significantly.

4.5. Comparison with state-of-the-art

This section presents series of experiments showing, that our **CF** framework produces more accurate results than existing state-of-the-art methods. With personal advice of the authors, we have reimplemented a horizon line-based localization method (abbreviated as *HLoc*[2]) by Saurer *et al*. [29] into the same DEM rendering pipeline as **CF**, and evaluated its ability to find correct camera orientation with known camera position. We used the best dir&loc [29] scheme to calculate a heading estimate of a given query and a panorama horizon line, followed by Iterative Closest Points (ICP) to obtain the full 3D camera rotation.

We report the results on GeoPose3K test set (Fig. 9), CH1 dataset [5] (Fig. 10), and Venturi Mountain dataset [26] (Tab. 2). First, we provide an upper bound

[2]The source code and experiment data are available at: http://cphoto.fit.vutbr.cz/semantic-orientation.

Resolution	Method	Avg. mean	Avg. stddev	F1	F2	F3	F4	F5	F6	J1	J2	J3	J4	J5	J6
low	**CF-VCC-2011-m3D** (ours)	**5.93**	**21.82**	1.82	**3.50**	30.26	**4.15**	**13.92**	4.02	**3.51**	1.20	1.31	**1.20**	**5.93**	**2.41**
	VCC-2011-m3D	21.06	44.20	**1.00**	6.01	**21.27**	116.87	41.30	**1.69**	132.92	**0.71**	**0.55**	**1.20**	41.04	4.46
	CF-VCC-2011 (ours)	34.19	41.75	6.67	5.00	100.11	132.06	51.42	39.95	7.41	6.75	23.87	8.85	55.39	19.01
high	**CF-VCC-2011-m3D** (ours)	**1.92**	**10.62**	2.57	3.68	**1.06**	1.57	**2.68**	**0.61**	4.54	1.26	**0.50**	**1.18**	**5.24**	**0.47**
	VCC-2011-m3D	2.88	14.72	1.49	8.94	1.27	6.25	4.42	1.18	5.17	1.08	**0.50**	**1.18**	6.29	0.66
	CF-VCC-2011 (ours)	12.42	32.44	**0.93**	**0.67**	85.68	**1.09**	21.18	2.45	**1.85**	**0.93**	8.32	1.42	41.65	0.75
-	HLoc-synthetic	28.0	50.54	52.73	1.84	11.54	36.08	4.17	10.21	115.54	86.08	6.01	4.11	3.84	40.85
-	HLoc-Deeplab	98.76	61.24	133.69	47.52	85.66	128.47	54.48	120.4	115.23	134.89	28.61	100.1	57.67	155.35
-	$\mathrm{RFN}_h - \mathrm{HOR}$ [26]	1.23	1.24	-	-	-	-	-	-	-	-	-	-	-	-
-	SENSORS [26]	9.43	4.16	-	-	-	-	-	-	-	-	-	-	-	-

Table 2. Mean orientation error (in degrees) of the proposed method and its variants on Venturi Mountain dataset (video sequences F1 – F6, and J1 – J6). The last two rows refer to the reference results obtained with the help of device inertial sensors by Porzi *et al*. [26].

of our *HLoc* implementation. We measure results with horizon line rendered from the DEM with perspective projection (*HLoc-synthetic*). Second, we measure *HLoc* performance on queries with automatically segmented sky class using Deeplab (*HLoc-Deeplab*). Third, on the CH1 dataset, we use queries from the original publication [29], segmented with the help of the user (*HLoc-CH1*). According to our experiments, *HLoc* is fairly sensitive to the quality of the segmentation – *HLoc-Deeplab* provides poor results compared to the *HLoc-synthetic*, and *HLoc-CH1*. Performance of our **CF** framework is similar to *HLoc-synthetic* and is higher by a large margin compared to *HLoc-CH1*. Please note that our **CF** framework uses only **automatically** detected segments and edges. Compared to the *HLoc-Deeplab*, it scored significantly better on all three datasets. We conclude, that the *HLoc* method depends on fine-grained horizon line segmentation and it is not suitable for a fully automatic processing. Our *CF* framework is much more robust to imprecisions in feature detection, and achieves significantly better results compared to *HLoc* for automatically detected segment classes.

We further compare our method with Robust silhouette map matching metric (m3D-2011) by Baboud *et al*. [7]. This non-linear metric penalizes crossings of query edges with synthetic depth discontinuities. The metric is fairly accurate, however, it needs a reasonably small subset of candidate rotations since its computation time is enormous (hours per query on the whole $SO(3)$). One can look at this metric as a geometric verification step; once the subset of candidate rotations is known, we can use this metric to verify and re-rank the best candidates. Tab. 2 illustrates, that our **CF-VCC-2011** framework (segments + edges) already outperforms more complex Robust silhouette map matching metric (VCC-2011-m3D) on several Venturi sequences (F1, F2, F4, J1, and J2) on high resolution. On the other hand, the mean error of our method is considerably higher than VCC-2011-m3D for the sequences F3, F5, and J5. These sequences are sparsely populated with synthetic semantic segment descriptions, which is attributed to the inaccuracy of the GIS database (OpenStreetMap). Additionally, in these sequences the horizon line is often straight, which rapidly reduces its descriptivity.

However, when we use **CF-VCC-2011** as an initial estimate (search space reduction) for m3D-2011 [7], we improve the state-of-the-art result of m3D-2011, since it searches through the smaller number of outlier candidates. Considerable improvement has been achieved especially at the low resolution. The combination of m3D-2011 and our **CF** method (**CF-VCC-2011-m3D**) achieves mean error of 5.93°, which is smaller by more than 70% compared to the original VCC-2011-m3D method (see Tab. 2, **CF-VCC-2011-m3D** vs. VCC-2011-m3D). This is an important result, since the proposed method is fast on low resolution (seconds per query). The most accurate result (1.92°) was achived by **CF-VCC-2011-m3D** at high resolution. The improvement over the original method **VCC-2011-m3D** (2.88°) is 33%.

5. Conclusion

We proposed a novel method for camera orientation estimation in natural scenes, which is based on semantic segmentation cues. To extract semantic segments from the query image, we utilized three state-of-the-art semantic segmentation methods and evaluated their suitability for the orientation estimation task. We used an extensive synthetic dataset GeoPose3K to train the methods for extraction of natural segments like forested areas, bodies of water, sky segments or glaciers.

Our experiments indicate that boundaries of semantic segments are less informative than their areas which are therefore complementing the information stored in edge maps. Using the proposed confidence-based fusion framework we measured that semantic segments are more informative than edges. However, as the edges add complementary information to the estimation process, the *combination* of semantic segments and edges achieves the state-of-the-art result in camera orientation estimation on natural scenes.

Acknowledgements This work was supported by V3C – "Visual Computing Competence Center" by Technology Agency of the Czech Republic, project no. TE01020415; 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". We thank Jakub Pelikán for semantic segmentation.

References

[1] J. Ajgl and M. Simandl. Design of a robust fusion of probability densities. In *Proceedings of the American Control Conference*, volume 2015-July, pages 4204–4209, 2015.

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

[3] S. Ardeshir, K. M. Collins-Sibley, and M. Shah. Geo-semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 2792–2799, 2015.

[4] A. Armagan, M. Hirzer, P. M. Roth, and V. Lepetit. Learning to align semantic segmentation and 2.5d maps for geolocalization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

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

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

[7] L. Baboud, M. Čadík, E. Eisemann, and 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*, pages 41–48, Washington, D.C., USA, 2011. IEEE Computer Society Press.

[8] R. Behringer. Improving registration precision through visual horizon silhouette matching. In *Proceedings of the International Workshop on Augmented Reality : Placing Artificial Objects in Real Scenes*, IWAR '98, pages 225–232, Natick, MA, USA, 1999. A. K. Peters, Ltd.

[9] J. Brejcha and M. Čadík. GeoPose3K: Mountain landscape dataset for camera pose estimation in outdoor environments. *Image and Vision Computing*, 66:1 – 14, 2017.

[10] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, and S. Savarese. Semantic Cross-View Matching. In *2015 IEEE International Conference on Computer Vision Workshop*, pages 1044–1052, New York, NY, USA, 2015. IEEE.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv preprint*, pages 1–14, 2016.

[12] Y. Chen, G. Qian, K. Gunda, H. Gupta, and K. Shafique. Camera geolocation from mountain images. In *2015 18th International Conference on Information Fusion*, pages 1587–1596, New York, NY, USA, 2015. IEEE.

[13] H. Chu, A. Gallagher, and T. Chen. GPS Refinement and Camera Orientation Estimation from a Single Image and a 2D Map. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 171–178, jun 2014.

[14] R. Fedorov, P. Fraternali, and M. Tagliasacchi. Mountain peak identification in visual content based on coarse Digital Elevation Models. In *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*, pages 7–11, 2014.

[15] R. I. Hammoud, S. A. Kuzdeba, B. Berard, V. Tom, R. Ivey, R. Bostwick, J. Handuber, L. Vinciguerra, N. Shnidman, and 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*, pages 320–327, Washington, D.C., USA, 2013. IEEE Computer Society Press.

[16] A. Irschara, C. Zach, J. M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 2599–2606, Washington, D.C., USA, 2009. IEEE Computer Society Press.

[17] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pages 2938–2946, New York, NY, USA, 2015. IEEE.

[18] J. Košecká and W. Zhang. Video compass. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision — ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part IV*, pages 476–490, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[19] P. J. Kostelec and D. N. Rockmore. FFTs on the rotation group. *Journal of Fourier Analysis and Applications*, 14(2):145–179, 2008.

[20] L. Ladicky, C. Russell, P. Kohli, and 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, pages 239–253, Berlin, Heidelberg, 2010. Springer-Verlag.

[21] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*, pages 15–29, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[22] J. Long, E. Shelhamer, and 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*, pages 3431–3440, 2015.

[23] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-dof localization on mobile devices. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014: 13th European Conference, Zurich,*

Switzerland, September 6-12, 2014, Proceedings, Part II, pages 268–283, Cham, 2014. Springer International Publishing.

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

[25] P. C. Naval, M. Mukunoki, M. Minoh, and 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*, pages 9–16, 1997.

[26] L. Porzi, S. R. Bulò, O. Lanz, P. Valigi, and E. Ricci. An automatic image-to-DEM alignment approach for annotating mountains pictures on a smartphone. *Machine Vision and Applications*, pages 1–15, 2016.

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

[28] T. Produit, D. Tuia, F. Golay, and C. Strecha. Pose estimation of landscape images using DEM and orthophotos. In *2012 International Conference on Computer Vision in Remote Sensing (CVRS)*, pages 209–214, New York, NY, USA, 2012. IEEE.

[29] O. Saurer, G. Baatz, K. Köser, L. Ladický, and M. Pollefeys. Image based geo-localization in the alps. *International Journal of Computer Vision*, 116(3):213–225, 2016.

[30] T. Senlet, T. El-Gaaly, and A. Elgammal. Hierarchical semantic hashing: Visual localization from buildings on maps. In *Proceedings - International Conference on Pattern Recognition*, pages 2990–2995, New York, NY, USA, 2014. IEEE.

[31] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3D models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 532–539, Washington, D.C., USA, 2014. IEEE Computer Society Press.

[32] E. Tzeng, A. Zhai, M. Clements, R. Townshend, and A. Zakhor. User-Driven Geolocation of Untagged Desert Imagery Using Digital Elevation Models. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 237–244, New York, NY, USA, 2013. IEEE.