Semantic Segmentation for 3D Localization in Urban Environments

Anil Armagan, Martin Hirzer and Vincent Lepetit
Institute of Computer Graphics and Vision
Graz University of Technology, Austria
{armagan,hirzer,lepetit}@icg.tugraz.at

***Abstract*—We show how to use simple 2.5D maps of buildings and recent advances in image segmentation and machine learning to geo-localize an input image of an urban scene: We first extract the façades of the buildings and their edges from the image, and then look for the orientation and location that align a 3D rendering of the map with these segments. We discuss how to use a 3D tracking system to acquire the data required for training the segmentation method, the segmentation itself, and how we use the segmentations to evaluate the quality of the alignment.**

I. Introduction

Accurate geo-localization of images is a very active area in Computer Vision, as it can potentially be used for autonomous driving and Augmented Reality. GPS and compass information are often not accurate enough for such applications, thus, image-based localization techniques have been developed in order to improve the camera pose estimate. However, standard registration methods based on image matching quickly become impractical, as many images need to be captured and registered in advance. Even collections such as GoogleStreetView are rather sparsely sampled and exhibit only specific illumination and season conditions, making image matching challenging.

We recently proposed a method that uses only an untextured 2.5D map as reference information [1]. 2.5D maps hold the 2D information about the environment, more precisely the buildings' outlines and their heights. In practice, we download such models from OpenStreetMap[1]. However, [1] relies heavily on the extraction of straight line segments, in particular to find the reprojections of the corners of the buildings. This step is specifically fragile, as buildings' edges do not necessarily appear as line segments in images, and additionally, also spurious segments can be extracted without corresponding to the corner of a building.

Instead, as shown in Fig. 1, we rely here on recent advances in semantic segmentation [4] and [11] to extract the visible façades and their edges. Since the segmentation method requires a large amount of training data, we use a 3D tracking algorithm to easily label many images with the required information. In order to find the correct pose, we sample random poses around the initial pose given by the sensors, and keep the one that maximizes the log-likelihood of the rendering of a 2.5D map of the surroundings. This sampling can be done efficiently as rendering is very fast on the GPU.

[1]http://osmbuildings.org

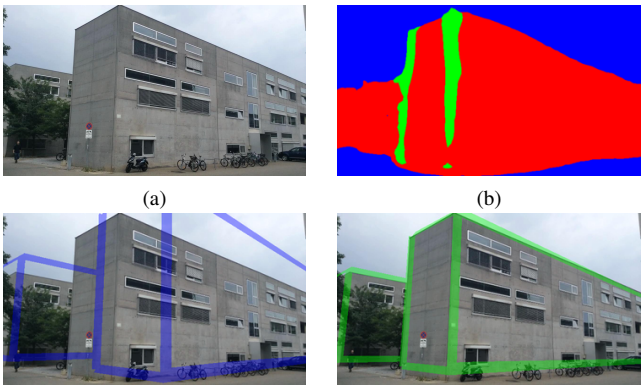

Fig. 1. Overview of our approach: Given an input image (a), we segment the façades and their edges (b). We sample poses around the pose provided by the sensors (c), and keep the one that aligns the 2.5D map and the segmentation (d).

In the remainder of this paper, we first discuss related work. We then describe the semantic segmentation step, and how we efficiently acquired labeled data in order to train the segmentation method. Next we present the segmentation-based 3D localization step, and finally, we demonstrate the whole approach on a challenging dataset.

II. Related Work

Camera localization using untextured models and maps has recently drawn a lot of attention in the CV community. However, when using single, narrow field-of-view images for localization, the results have not been satisfactory for AR purposes yet. Below, we discuss related work for both the localization and the segmentation task.

A. Image-Based Localization using Pre-Registered Images

Given one or more input images and optionally a sensor prior location from GPS, orientation from compass and magnetometer, image-based localization retrieves similar pre-registered images from a database to compute the pose of the input image. For example, [15] demonstrated image-based localization using 20 km of urban street-side imagery, organized in a vocabulary tree to handle the massive amount of data. [20] and [18] used images from GoogleStreetView, which, however, is only sparsely sampled and not available at all in certain regions and countries. Very recently, [10] used a CNN to predict a 6 DoF camera pose directly from an image.

978-1-5090-5808-2/17/$31.00

However, this approach is limited to the area that was used for training.

The major disadvantage of image-based localization approaches remains that they do not scale well: Many images need to be captured for each new location, and, even with sufficiently dense sampling, it is still very challenging to match images under changing conditions due to illumination, season, construction activity and many other sources of change.

B. Image-Based Localization from Untextured Models.

[2] and [5] rely on the skyline to align an untextured model with an input image. This assumes that the skyline or the horizon are visible. [3] registers semantically labeled images with a 3D terrain model, but can estimate only the camera orientation. [8] registers panoramic images with 2D maps using a building façade orientation descriptor. Such large field-of-view information significantly improves the success rate of localization. Mobile devices have a narrow field of view, and a descriptor such as the one used by [8] is not discriminant enough in such situations. [6] also considers panoramic images and aims at detecting vertical building outlines and façade normals resulting in 2D fragments which are then matched with a 2D map. [7] computes a descriptor from vertical building outlines, which is then matched with a 2D map, but the authors had to partially use manual annotations of the input images. [13] registers an image with respect to a 2.5D model by matching 3D and 2D lines and points. However, a reference image needs to be manually annotated.

C. Segmentation for 3D Registration

[17] segments the façades in the input image as we do, and aligns a 2.5D map with the segmentation to find the 3D pose. However, this requires an optimization in the 6D pose space, and the authors have to introduce a swarm-based optimization. The use of panoramic images also helps the optimization as discussed above. With regular cameras, considering only the façades is often ambiguous: For example, if a street has only aligned similar buildings as in the second row of Fig. 3, the translation along the direction of the street is not constrained. To prevent this, [1] also considers the corners of the buildings, but extract them as vertical straight line segments, which is a very fragile approach since corners do not necessarily appear as easily extractable line segments. By contrast, we show that we can use state-of-the-art segmentation methods to extract the edges of the buildings.

[12] and [19] also consider the buildings' edges, however, [12] relies on orthographic aerial images, which makes the task easier, and [19] assumes that the façades are highly repetitive. [9] also uses auto-context for façade segmentation, however, they also apply their method to very repetitive façades as they also use grammar methods, and they have to introduce complex handcrafted features. By contrast, we make use of recent advances in semantic segmentation based on CNNs, which automatically learn appropriate image features.

In the next section, we describe how we segment the input image. We then describe how we obtain the training data to learn to segment the buildings' edges and façades, and finally, we show how to use the segmentation to accurately geo-localize an input image.

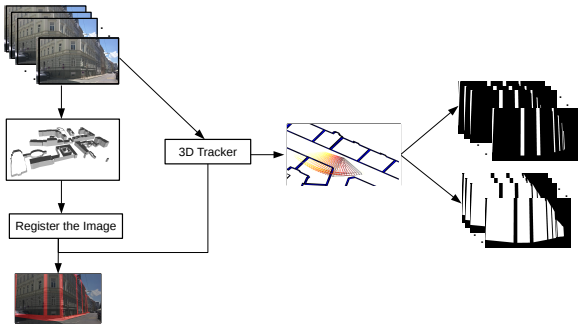

Fig. 2. Efficiently labeling images. (a) We manually register simple 3D models created from 2.5D maps in the first frames of several video sequences. (b) We automatically track these 3D models over the sequences. (c) This gives us the labels for the façades and their edges for all the frames.

III. Semantic Segmentation

Given a color input image I, we train a fully convolutional network (FCN) [11] to perform a semantic segmentation. FCN applies a series of convolutional and pooling layers to the input image, followed by deconvolution layers to produce a segmentation map of the whole image at the original resolution. Other recent works have a similar architecture and performance [4] and [14].

In our case, we aim at segmenting the façades and the vertical edges at building corners or between different façades. Everything else is referred to as 'background'. We therefore consider three classes: façade, vertical edge, and background.

For training, we use a stage-wise procedure, where we start with a coarse network (FCN-32s) initialized from VGG-16 [16], fine-tune it on our data, and then use the thus generated model to initialize the weights of a more fine-grained network (FCN-16s). This process is repeated in order to compute the final segmentation network having an 8 pixels prediction stride (FCN-8s).

The output of the segmentation for a given color image I is a set of probability maps having the same resolution as I, one for each of our classes:

$$S(I) = \{P_{\text{facade}}, P_{\text{vertical\_edge}}, P_{\text{background}}\}\,. \tag{1}$$

IV. Acquisition of Training Data

Deep-learning segmentation methods require a large number of training images to generalize well, however, manual annotation is costly. As shown in Fig. 2, we therefore use a 3D tracking system [1] to easily annotate frames of video sequences. First, we create simple 3D models from the 2.5D maps. These models are not very detailed but sufficient for tracking. Then, for each sequence, we initialize the pose for the first frame manually, and the tracker estimates the poses for the remaining frames. This allows us to label façades and their vertical edges very efficiently.

More precisely, we recorded 95 video sequences using a smart device (an *Apple iPhone 6s*) with an average length of about 10 seconds. In order to ensure an accurate labeling, in particular for the vertical edges, we exploit our model rendering pipeline: We only keep frames in which the reprojection of the 3D model is well aligned with the real image, and remove

those frames that suffer from tracking errors or drift. In this way, we obtain a training set of 289 images with minimal manual effort, except for the registration of the 95 initial frames. Finally, we augment the training set by horizontally mirroring each image, yielding a training set of 578 samples in total.

V. 3D Localization

We use the probability maps $S(I)$ to geo-localize an input color image I, starting from an initial estimate $\tilde{\mathbf{p}}$ of the pose provided by the sensors of the device, and a 2.5D map of the surrounding. In practice, $\tilde{\mathbf{p}}$ can be far away from the correct pose, but it still gives us a coarse estimate of the correct pose. We then look for the pose around $\tilde{\mathbf{p}}$ with the largest log-likelihood given the input image:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \mathcal{L}(\mathbf{p}) \,, \tag{2}$$

where $\mathcal{L}(\mathbf{p})$ is the log-likelihood:

$$\mathcal{L}(\mathbf{p}) = \sum_{\mathbf{x}} \log P_{c(\mathbf{p},\mathbf{x})}(\mathbf{x}) \,. \tag{3}$$

The sum runs over all image locations $\mathbf{x}$; $c(\mathbf{p}, \mathbf{x})$ is the class at location $\mathbf{x}$ when rendering the model under pose $\mathbf{p}$, and $P_c(\mathbf{x})$ is the probability for class c at location $\mathbf{x}$ where P_c is one of the probability maps predicted by the semantic segmentation step in Eq. (1).

As the log-likelihood function in Eq. (3) is not differentiable and may have many local maximums, we sample poses around the sensor pose $\tilde{\mathbf{p}}$ on a regular grid, and keep the one with the largest log-likelihood.

VI. Evaluation

We evaluated our approach on the dataset of [1], which contains 32 images taken by an *Apple iPad Air* in urban and suburban environments of Graz, Austria. The original resolution of the images is 1280×720, but we downscaled them to 640×360. The resolution of the renderings is the same as the resolution of the input images after downscaling.

The sensor positioning errors of the dataset range from about 0.4 m to about 16.5 m, with an average error of about 8 m. The rotational errors of the gyroscopes are small, however, the orientation error around the up direction, given by the compass, can be as large as 30°. We sample the location in a squared region of 20 m×20 m with a step size of one meter in each direction. We also sample the rotation of the camera around the up direction every 3° over a range of $[-30°; +30°]$ centered on the orientation provided by the compass.

Fig. 3 shows the full results, and Fig. 4 shows one failure case due to a segmentation error. We quantitatively evaluated our method for both positioning and orientational errors. Our method decreases the mean error to 4.5 m; 56% of the images have an error below 2 m, and 78% below 5 m. Our method performs well on decreasing the orientation errors as well: After applying our method, 62% of the images have an orientation error below 2°, 75% are below 5°. Table I gives the time spent by the significant steps of our method. Note that the input image is only segmented once and this segmentation is used for each pose evaluation. The overall time depends on the number of poses evaluated. This number is a meta-parameter of our method: Increasing it will improve the accuracy of the final pose estimate, but the computation time will also increase linearly.

Step	Computation Time per Frame (ms)
Semantic segmentation	120
Rendering	5
Log-Likelihood	10
Total	135

TABLE I. Computation time for each step of our method.

VII. Conclusion

We described an approach for geo-localization based on the semantic segmentation of the input image. The segmentation allows us to robustly align a simple 2.5D model of the surroundings with the image, even though the model is not textured. One possible direction to speed up our approach is to use a multiresolution framework, where the search starts from a big and sparse grid and is then successively refined on smaller and denser grids.

Acknowledgment

This work was funded by the Christian Doppler Laboratory for Semantic 3D Computer Vision.

References

[1] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. Instant Outdoor Localization and SLAM Initialization from 2.5D Maps. In *ISMAR*, 2015.

[2] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In *ECCV*, 2012.

[3] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Leveraging Topographic Maps for Image to Terrain Alignment. In *DIMPVT*, 2012.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *CoRR*, 2015.

[5] M. Bansal and K. Daniilidis. Geometric Urban Geo-Localization. In *CVPR*, 2014.

[6] T. Cham, A. Ciptadi, W. Tan, M. Pham, and L. Chia. Estimating Camera Pose from a Single Urban Ground-View Omnidirectional Image and a 2D Building Outline Map. In *CVPR*, 2010.

[7] H. Chu, A. Gallagher, and T. Chen. GPS Refinement and Camera Orientation Estimation from a Single Image and a 2D Map. In *CVPRW*, 2014.

[8] P. David and S. Ho. Orientation Descriptors for Localization in Urban Environments. In *IROS*, 2011.

[9] V. Jampani, R. Gadde, and P. V. Gehler. Efficient Facade Segmentation Using Auto-Context. In *WACV*, 2015.

[10] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DoF Camera Relocalization. In *ICCV*, 2015.

[11] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, 2015.

[12] P. Meixner, A. Wendel, H. Bischof, and F. Leberl. Building Façade Separation in Vertical Aerial Images. *ISPRS*, I-3:239–243, 2012.

[13] S. Ramalingam, S. Bouaziz, and P. F. Sturm. Pose Estimation Using Both Points and Lines for Geo-Localization. In *ICRA*, 2011.

[14] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015.

[15] G. Schindler, M. A. Brown, and R. Szeliski. City-Scale Location Recognition. In *CVPR*, 2007.

[16] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, 2014.

[17] A. Taneja, L. Ballan, and M. Pollefeys. Registration of Spherical Panoramic Images with Cadastral 3D Models. In *DIMPVT*, 2012.

[18] G. Vaca-Castano, A. R. Zamir, and M. Shah. City Scale Geo-Spatial Trajectory Estimation of a Moving Camera. In *CVPR*, 2012.

[19] A. Wendel, M. Donoser, and H. Bischof. Unsupervised Facade Segmentation Using Repetitive Patterns. In *DAGM*, 2010.

[20] A. R. Zamir and M. Shah. Accurate Image Localization Based on Google Maps Street View. In *ECCV*, 2010.

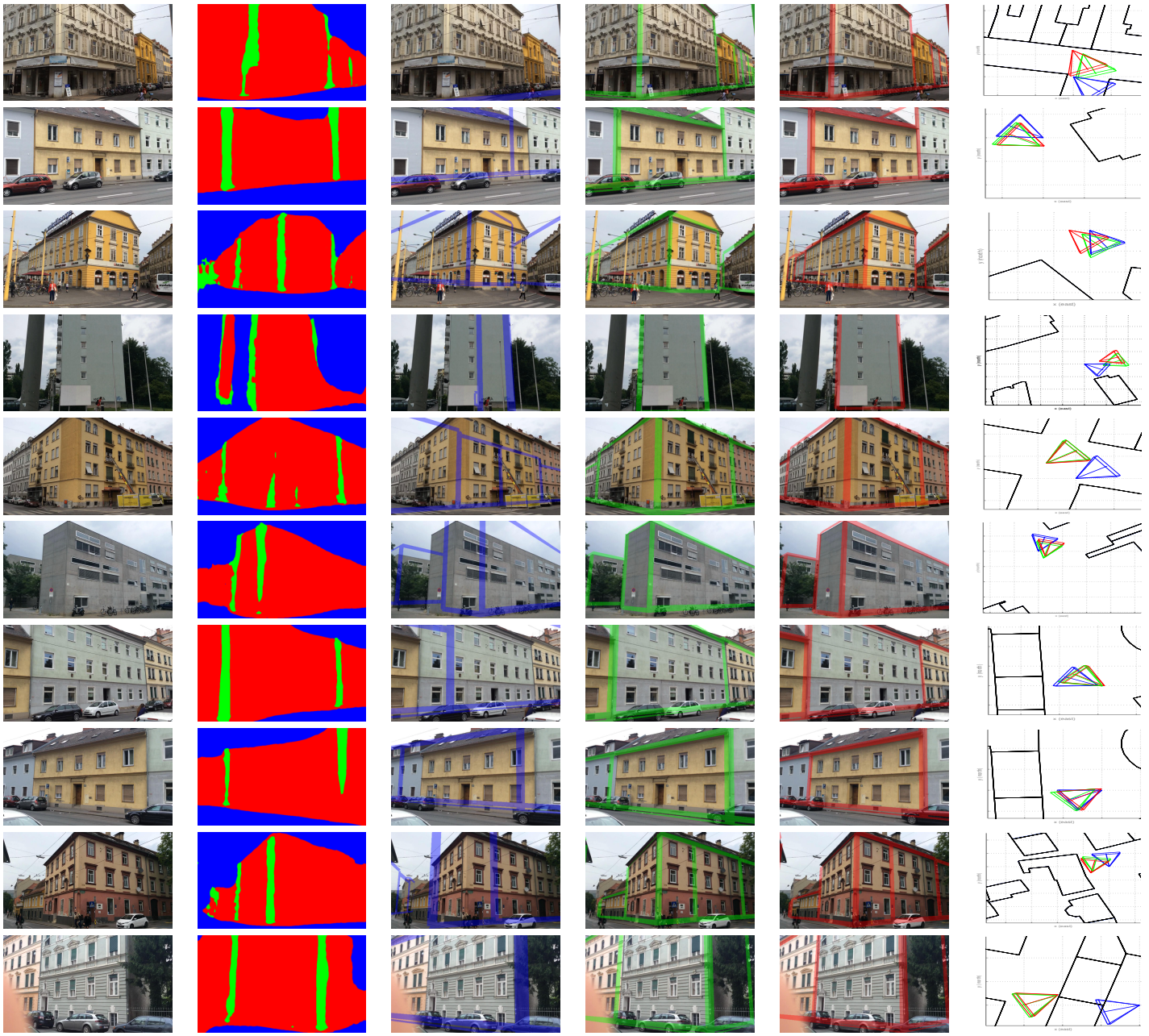

Fig. 3. Visual comparison of poses. Each row shows the results for a test image. (a) Test image, (b) segmented image using our network (red: façade, green: vertical edges, blue: background), (c) rendering of the 2.5D map using the sensor pose, (d) rendering using the best pose found with our method, (e) rendering using the ground truth pose, (f) the different camera poses shown on a map (blue: sensor pose, green: pose found with our method, red: ground truth pose).

Fig. 4. Failure case of our method. The segmentation fails to find one of the edges, resulting in an incorrect evaluation of the model-image alignment quality.