3D Localization in Urban Environments from Single Images *

Anil Armagan[1], Martin Hirzer[1], Peter M. Roth[1] and Vincent Lepetit[1,2]

***Abstract*— In this paper, we tackle the problem of geo-localization in urban environments overcoming the limitations in terms of accuracy of sensors like GPS, compass and accelerometer. For that purpose, we adopt recent findings in image segmentation and machine learning and combine them with the valuable information given by 2.5D maps of buildings. In particular, we first extract the façades of buildings and their edges and use this information to estimate the orientation and location that best align an input image to a 3D rendering of the given 2.5D map. As this step builds on a learned semantic segmentation procedure, rich training data is required. Thus, we also discuss how the required training data can be efficiently generated via a 3D tracking system.**

I. INTRODUCTION

Accurate geo-localization of images is a very active area in Computer Vision, as it can potentially be used for applications such as autonomous driving and Augmented Reality. As the typically available GPS and compass information are often not accurate enough for such applications, we recently proposed a method that builds only on untextured 2.5D maps [3]. In general, 2.5D maps hold the 2D information about the environment, more precisely the buildings' outlines and their heights. However, this approach is limited in practice, as it heavily relies on the often unreliable and error prone extraction of straight line segments to find the re-projections of the corners of the buildings.

To overcome this limitation, as shown in Fig. 1, we replace this step by semantic segmentation (i.e., [4] and [5]) to extract the visible façades and their edges, which is described in more detail in Sec. II. Since learning the necessary model requires a large amount of training data, as detailed in Sec. III, we use a 3D tracking algorithm to semi-automatically label the huge amount of required training images. In order to estimate the correct pose, we introduce two strategies. The first strategy samples random poses around the initial pose given by the sensors and selects the best one. The second strategy builds on a more advanced search algorithm by using CNNs to iteratively update the pose. Both approaches are discussed in Sec. IV.

II. SEMANTIC SEGMENTATION

Given a color input image I, we train a fully convolutional network (FCN) [5] 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. In our case, we aim at segmenting the façades and the edges at building corners or between different façades. Everything else is referred to as 'background'. We therefore consider four classes: façade, vertical and horizontal edges and background. We use a stage-wise training procedure, where we start with a coarse network (FCN-32s) initialized from VGG-16 [6], 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).

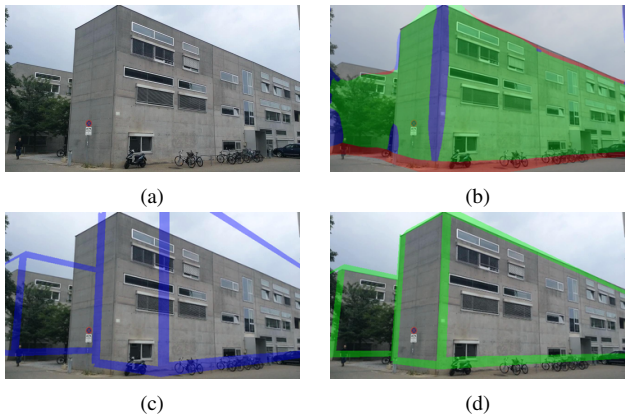

Fig. 1. Overview of our approach: Given an input image (a), we segment the façades and their edges (b). We can either sample poses around the pose provided by the sensors or use CNNs to move the camera starting from the sensor pose (c), and keep the pose that aligns the 2.5D map and the segmentation best (d).

III. ACQUISITION OF TRAINING DATA

Deep-learning segmentation methods require a large number of training images to generalize well, however, manual annotation is costly. We therefore use a 3D tracking system [3] to easily annotate frames of video sequences. First, we create simple 3D models from the 2.5D maps. 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 short video sequences using a smart device. In order to ensure an accurate labeling, in particular for the vertical edges, we only keep frames in which the re-projection of the 3D model is well aligned with the real image, and remove those frames that suffer from tracking errors or drift.

* This work is a spotlight for the already published/accepted publications JURSE'17 [1] and CVPR'17 [2].

[1] The authors are with the Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria {armagan,hirzer,pmroth,lepetit}@icg.tugraz.at

[2] The author is with the Laboratoire Bordelais de Recherche en Informatique, Universit de Bordeaux, Bordeaux, France

(a) (b) (c) (d)

Fig. 2. Converging from an initial sensor estimate. (a) Test image with the ground truth pose overlaid, (b) segmented image, (c) noisy sensor pose, (d) pose found with our method.

IV. 3D Localization

Building on the same segmentation approach trained using the training data as described in Secs. II and III we proposed two different approaches for pose estimation.

A. Direct Pose Selection [1]

Given a coarse initial estimate $\tilde{\mathbf{p}}$ of the pose provided by the sensors and a 2.5D map of its surrounding, the goal is to finally estimate the correct pose $\hat{\mathbf{p}}$. Therefore, we sample poses in a regular grid around $\tilde{\mathbf{p}}$ and estimate

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \mathscr{L}(\mathbf{p}), \tag{1}$$

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

$$\mathscr{L}(\mathbf{p}) = \sum_{\mathbf{x}} \log P_{c(\mathbf{p},\mathbf{x})}(\mathbf{x}). \tag{2}$$

The sum runs over all image locations $\mathbf{x}$, where $c(\mathbf{p},\mathbf{x})$ 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.

B. CNN-based Refinement [2]

As this brute-force strategy is not very efficient we additionally proposed a CNN-based approach for iterative pose refinement. We discretize the directions along the ground plane into 8 possible directions, defined in the camera coordinate system, and train a network to predict the direction that improves the estimated location, given the semantic segmentation of the image and a rendering of the 2.5D map from the current estimate. We also add a class that indicates that the estimated location is already correct and should not be changed. Thus, the network, denoted by CNN_t, yields a 9-dimensional output vector

$$\mathbf{d}_t = \mathrm{CNN}_t(R_\mathrm{F}, R_\mathrm{HE}, R_\mathrm{VE}, R_\mathrm{BG}, S_\mathrm{F}, S_\mathrm{HE}, S_\mathrm{VE}, S_\mathrm{BG}), \tag{3}$$

where S_F, S_HE, S_VE, and S_BG are the probability maps computed by the semantic segmentation for the input image for the classes façade, horizontal edge, vertical edge, and background, respectively. R_F, R_HE, R_VE, R_BG are binary maps for the same classes, created by rendering the 2.5D map for the current pose estimate.

In addition, we train a second network to refine the orientations:

$$\mathbf{d}_o = \mathrm{CNN}_o(R_\mathrm{F}, R_\mathrm{HE}, R_\mathrm{VE}, R_\mathrm{BG}, S_\mathrm{F}, S_\mathrm{HE}, S_\mathrm{VE}, S_\mathrm{BG}), \tag{4}$$

where $\mathbf{d}_o$ is a 3-dimensional vector, covering the probabilities to rotate the camera to the right, to the left or not rotate it at all.

Starting from the initial estimate $\tilde{\mathbf{p}}$, we iteratively apply CNN_t and CNN_o and update the current pose after each iteration. These steps are iterated until both networks are converged and predict not to move. In particular, there are two main advantages of having two networks: (a) As the networks for translation and orientation are treated separately, we do not need to balance between them. (b) The two detached problems are much easier, reducing both, the training and the inference effort.

V. Results and Summary

Two illustrative results obtained by the approach described in Sec.IV-B are shown in Fig. 2. It clearly can be seen that the initial sensor poses (Fig. 2(c)) does not cover the groundtruth (Fig. 2(a)) very well, whereas the finally estimated poses (Fig. 2(c)) using the segmentation results (Fig. 2(b)) perfectly fit the buildings. Overall, this demonstrates that adopting ideas from semantic segmentation in combination with convolutional neural networks and the information provided by 2.5D maps can successfully be used for estimating the poses of buildings and thus their exact location. For more details, we would like to refer to [1] and [2].

References

[1] A. Armagan, M. Hirzer, and V. Lepetit, "Semantic Segmentation for 3D Localization in Urban Environments," in *JURSE*, 2017, (Best Paper Award).

[2] A. Armagan, M. Hirzer, P. M. Roth, and V. Lepetit, "Learning to Align Semantic Segmentation and 2.5D Maps for Geolocalization," in *CVPR*, 2017, (accepted).

[3] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit, "Instant Outdoor Localization and SLAM Initialization from 2.5D Maps," in *ISMAR*, 2015, Best Paper Award.

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

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

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