Improving Constrained Bundle Adjustment Through Semantic Scene Labeling

Achkan Salehi[1(✉)], Vincent Gay-Bellile[1], Steve Bourgeois[1], and Frédéric Chausse[2]

[1] CEA LIST, Vision and Content Engineering Lab, Point Courrier 94, 91191 Gif-sur-Yvette, France
{achkan.salehi,vincent.gay-bellile,steve.bourgeois}@cea.fr
[2] Institut Pascal, UMR 6602 CNRS, Clermont-Ferrand, France
frederic.chausse@univ-bpclermont.fr

Abstract. There is no doubt that SLAM and deep learning methods can benefit from each other. Most recent approaches to coupling those two subjects, however, either use SLAM to improve the learning process, or tend to ignore the geometric solutions that are currently used by SLAM systems. In this work, we focus on improving city-scale SLAM through the use of deep learning. More precisely, we propose to use CNN-based scene labeling to geometrically constrain bundle adjustment. Our experiments indicate a considerable increase in robustness and precision.

Keywords: SLAM · VSLAM · Bundle adjustment · Deep learning · Scene labeling

1 Introduction

The problem of the drift of monocular visual simultaneous localization and mapping (VSLAM) in seven degrees of freedom is well-known. Fusion of VSLAM, in particular key-frame bundle adjustment (BA) [1,2] with data from various sensors (*e.g.* GPS, IMU [3–5]) and databases (*e.g.* 3d textured or textureless 3d models, digital elevation models [6–8]) has proven to be a reliable solution to this problem. In this paper, we focus on fusion through constrained BA [2–4,6]. Among the available sensors and databases that can be used in constrained BA, textureless 3d building models are of particular interest, since the geometric constraints they impose on the reconstruction can prevent scale drift and also help in the estimation of camera yaw. Furthermore, they can be used to limit the impact of GPS bias on the reconstruction [9]. They are also, as opposed to textured models, widespread and easily (usually freely) available. However, methods that make use of such partial knowledge of the environment [6,8] face the problem of data association between 3d points and 3d building planes, that is, they must design a reliable method to segment the 3d point cloud and determine which points belong to buildings. In previous works [6,7], data association between 3d points and building models has been made by means of simple geometric constraints instead of photometric ones. This is due to the high cost of

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 133–142, 2016.
DOI: 10.1007/978-3-319-49409-8_13

scene labeling algorithms. Unfortunately, these simple geometric criteria often introduce high amounts of noise, which can lead to failure even when used in conjunction with M-estimators or RANSAC-like algorithms. This is especially true when building facades are completely occluded by nearby objects on which an important number of interest points are detected (*e.g.* trees, advertising boards, etc.). Since these methods clearly reach their limits in such environments, we must investigate the alternative solution, namely scene labeling. While current state of the art scene labeling algorithms allow a highly accurate segmentation, their cost often remains prohibitive for real-time use, even on the GPU. However, some state of the art segmentation methods such as [10] operate in two steps: first, a (reasonably fast) convolutional neural network (CNN) provides a crude and often spatially incoherent segmentation, which is refined using graph-based methods in a second time consuming step. This observation leads to the idea that the raw outputs of a CNN can be used with key-frame bundle adjustment as an a priori in data association, without much overhead, and possibly in real-time.

In this paper, we propose the use of scene labeling for data association in bundle adjustment constrained to building models. We segment each key-frame using a CNN inspired by the first stage of [10]. In order to reduce time complexity, we do not refine the outputs of the CNN, which, as we mentioned previously, are tainted by high levels of uncertainty. Instead, we replace the compute-intensive regularizations by a fast likelihood computation, with respect to a density function that we have previously learned by modeling our particular CNN as a Dirichlet process.

Roadmap. The following section (Sect. 2) is dedicated to notations and preliminaries. We discuss related works in Sect. 3, and present our approach in Sect. 4. Experiments are presented in Sect. 5 and we conclude the paper in Sect. 6.

2 Notations and Preliminaries

2.1 Local Key-Frame Based Bundle Adjustment

Bundle adjustment (BA) refers to the minimization of the sum of reprojection errors, *i.e.* the minimization of:

$$B(x) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{C}_i} ||x_{ij} - \pi_j(X_i))||^2 \tag{1}$$

where $\mathcal{M}$ denotes the set of all 3d point indexes, and C_i the set of cameras from which the point i is observable. The function π_j maps each 3d point X_i to its normalized 2d coordinates in the image plane of camera j, where the observation of X_i is noted x_{ij}. Successive images in a sequence are often similar, therefore keeping every single camera pose, apart from being very inefficient, is redundant. Thus, it is usual to perform key-frame bundle adjustment [1], in which only frames that present a significant amount of new information are kept. We also distinguish between global BA, in which all the cameras and points are optimized, and local BA in which only the $n \in \mathbb{N}$ last camera poses and the

points they observe are optimized. In this paper, the term bundle adjustment, unless otherwise stated, will refer to local key-frame based BA.

2.2 The Dirichlet Distribution

The Dirichlet distribution is a continuous distribution of discrete distributions, that can be seen as a generalization of the Beta distribution to higher dimensions. We will note

$$\mathfrak{D}(x_1, ...x_k, \alpha_1, ...\alpha_k) = \frac{1}{\beta(\alpha)} \prod_{s=1}^{k} x_s^{\alpha_s - 1} \tag{2}$$

the Dirichlet probability density function of parameters $\alpha = \{\alpha_1, ..., \alpha_k\}$. Here, β is the multinomial Beta function.

3 Related Work

Our work is firstly related to those that combine deep learning methods with geometrical SLAM algorithms, and secondly to SLAM approaches that make use of 3d textured or texture-less building models. Works such as [11] are based on a pure machine learning approach, but can benefit from more precise datasets generated by VSLAM. Such methods can thus be considered as dual to approaches such as ours. In their work, Costante et al. [12] present a deep network architecture that learns to predict the relative motion between consecutive frames, based on the dense optical flow of input images. Their method and ours can be seen as complementary to each other, since theirs intervenes at a lower level and can be used as part of a more general framework (for example by replacing the PnP-solving methods in traditional key-frame based systems), while our work targets the optimization (BA) that refines the results of such pose computations. The loop closure detection of [13] is related to the present paper in the sense that it makes use of deep nets to improve visual SLAM, but also operates on a lower level that bundle adjustment.

In previous works, data association for city-scale SLAM, as far as we know, has been either carried out via simple geometric criteria [6–8], or left to outlier-elimination processes such as RANSAC (*e.g.* in [14]). In [6–8], a point p is associated to a building if the ray cast from the center of the camera that goes through p intersects a building plane. Although the time complexity of such an evaluation remains negligible, its results naturally tend to contain a considerable amount of noise. In [14], Google street view images with known poses are first back-projected on 3d building models in an off-line pre-processing step. In other terms, each pixel in the street view image is associated to a 3d position. The database obtained via this procedure is then used in an on-line localization algorithm, by matching SIFT features between each new image and the ones in the database. The resulting $2d \leftrightarrow 3d$ matches define a PnP problem that is solved using standard techniques. Poor matches and wrong 3d positions are eliminated by a RANSAC, But no explicit attempt at filtering points that do not belong to buildings are made during the back-projection step.

4 Proposed Method

We seek to correct the results of local key-frame based BA [1] by constraining the reconstruction using 3d building models. To this end, we minimize a cost function inspired by [3,4]. Intuitively, we seek to respect the geometric constraints provided by 3d building models, so long as the sum of squared reprojection errors $B(X)$ remains below a certain threshold t. More precisely, we solve:

$$\arg\min_{X} \quad \frac{1}{t - B(X)} + \sum_{q \in Q} W_q d(q, N_q) \tag{3}$$

In the expression above, X is the vector that concatenates the parameters of all camera poses and the 3d positions of all 3d points. Q is the subset of 3D points from the map that have been classified as belonging to a building, $d(.)$ denotes the squared Euclidian distance, and N_q denotes the building plane closest to $q \in Q$. Each W_q is a weight, and its computation will be discussed at the end of section Sect. 4.2. This optimization problem has to be initialized with the minimizer of B. Thus, we first perform non-constrained BA and use its result as the initial value for X. We use the standard Levenberg-Marquardt algorithm to minimize the cost function of Eq. 3. We propose to use a fast CNN to determine the set Q.

4.1 Scene Labeling

The scene labeling algorithm we use is based on the first stage of the method presented in [10], but operates in a single scale, as opposed to the multiscale approach of the aforementioned paper. A Convolutional Neural Network (CNN) is trained on labeled data. The CNN assigns each pixel x in the input image I to a probability vector P_x of length 8. The $i-th$ component of P_x is the probability that x belongs to class i. The outputs of such of a CNN usually require post-processing in order to be regularized. Unfortunately, such methods are too time-consuming to be used in any BA system that runs in reasonable time. Thus, we are left with the raw outputs of the CNN, which more often than not lack spatial consistency.

To classify a pixel x mapped by the CNN to a distribution P_x, the most straight-forward approach would be to take the arg max of P_x. However, we think that a better approach is to take into account the general shape of the distribution. If a pixel x truly belongs to the class building, its distribution must have a specific form, and particular modes. Thus, by learning the expected form of this distribution, we can eliminate false positives, that is, distributions which reach their peak on the wrong label. To that end, we consider each distribution as a random variable, and given a set of labeled data, learn the Dirichlet distribution (defined in Sect. 1) from which the set

$$D_{\text{build}} = \{P_i \mid i \text{ belongs to a building in the image}\} \tag{4}$$

is a sample. Given a set of labeled data, the problem is to find the set of parameters $\alpha = \{\alpha_1, ..., \alpha_k\}$ for which the Dirichlet distribution fits best. This can be written as a maximum likelihood problem:

$$\arg\min_{\alpha} \prod_{X \in D_{\text{build}}} \mathfrak{D}(X|\alpha) \tag{5}$$

where $\mathfrak{D}$ is the Dirichlet density function of parameters α. To avoid underflow and also to simplify the notations, we solve the equivalent problem

$$\arg\min_{\alpha} \{-\ln(\prod_{X \in D_{\text{build}}} \mathfrak{D}(X|\alpha))\} \tag{6}$$

It can be shown using a few basic algebraic operations, that this can be written as

$$\arg\min_{\alpha} \{m \ln(\beta(\alpha)) + \sum_{i=1}^{k} (1 - \alpha_i) \ln(t_i)\} \tag{7}$$

where $t_i = \prod_{q \in D} \prod_{s=1}^{k} q(s)$ and k is the number of classes. We have $\alpha_i \geq 0$ for all i from the definition of the Dirichlet distribution. Therefore, we need to add k terms that will act as barriers, preventing the value of the α_i variables to become negative. The final cost function takes the form:

$$C(\alpha) = m \ln(\beta(\alpha)) + \sum_{i=1}^{k} (1 - \alpha_i) \ln(t_i) + \lambda \sum_{i=1}^{k} e^{-\alpha_i} \tag{8}$$

with $\lambda \in \mathbb{R}^+$ influencing the impact of the exponential terms. The Jacobian of this cost function is given by

$$\frac{\partial C}{\partial \alpha_l} = m(\psi_0(\alpha_l) - \psi_0(\sum_{i=1}^{k} \alpha_i)) - \ln(t_l) - \lambda e^{-\alpha_l} \tag{9}$$

where ψ_0 denotes the digamma function. We used the well known L-BFGS minimization algorithm [15] to learn the parameters α.

4.2 Integration in Constrained Bundle Adjustment

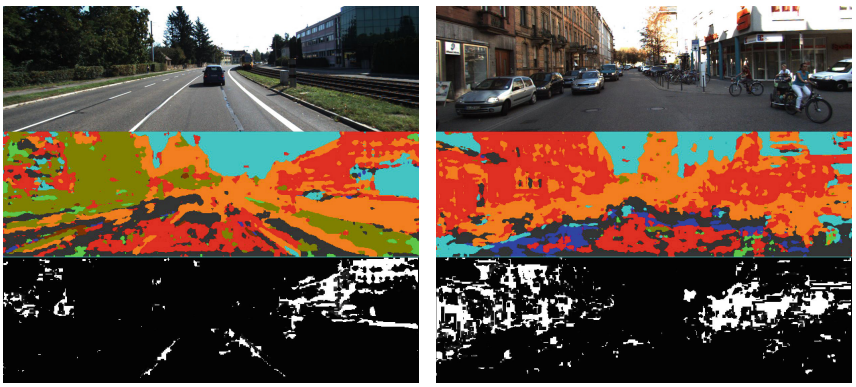
Each 3d point Z in the map has often more than one observation. Noting $\{I_0, I_1, .., I_n\}$ the key-frames in which Z is observed, and $\{z_0, ..., z_n\}$ its 2d observations, we seek to determine the class to which Z most likely belongs. As mentioned previously, our scene labeling algorithm runs once for each key-frame. This results in a set of probability distributions for each of the observations of Z, that we will note $M = \{P_0, P_1, ..., P_n\}$. In practice, this distributions can differ. We combine these distributions in the simplest possible manner, by computing a mean distribution

$$P_Z = \frac{1}{N_c} \sum_{i=1}^{n} P_i \tag{10}$$

Next, we compute $\mathfrak{D}(P_Z)$ and classify Z as belonging to Q if and only if $\mathfrak{D}(P_Z) > t_{\mathfrak{D}}$, where $t_{\mathfrak{D}}$ is a threshold. Finally, we set $W_q = \mathfrak{D}(P_Z)$ in Eq. 3.

5 Experimental Evaluation

We used a CNN implementation written in Torch 7 [16], based on the first stage of [10], but operating on only one scale, as opposed to the multiscale approach of the aforementioned paper. We used the following eight labels: {1*: sky,* 2*: tree,* 3*: road,* 4*: grass,* 5*: water,* 6*: building,* 7*: mountain,* 8*: object*}. The implementation was reasonably fast: the average segmentation time for a test image of size 640×480 was on average 0.6 s, when executed on a single core (on a single Intel(R) Core(TM) i7-4710HQ CPU @ 2.50 GHz) running at 1.6 GHz, under linux (Ubuntu 14.04). We used a subset of annotated images from the kitti dataset [17] to demonstrate the advantage of using the Dirichlet distribution $\mathfrak{D}$ to classify pixels compared to simply taking the arg max. On average, our method eliminated more than 71 % of false positives (*i.e.* points falsly classified as belonging to buildings by the CNN), while rejecting a small percentage (less than 5 %) of correctly classified building points. Examples are shown in Fig. 1. We conducted experiments on synthetic and real sequences in order to validate our bundle adjustment approach.



(a) (b)

Fig. 1. Examples illustrating the advantage of using the Dirichlet distribution to filter out poor classification results. Row 1: images fed as input to the CNN. Row 2: the raw result of the CNN, determined by taking the arg max of the distribution for each pixel. Each color corresponds to a class, and red pixels are those that have been classified as belonging to buildings. Row 3: the results of our filtering (i.e. using $\mathfrak{D}$ instead of the arg max). This binary image shows pixels that present a higher than 80 % probability of being building pixels according to the Dirichlet distribution. It can be seen that most false positives (mostly, building detection on the road plane) have been eliminated, at the cost of discarding a small portion of correctly classified building points. (Color figure online)

5.1 Synthetic Sequence

We generated an urban scene with important levels of occlusion that was realistic enough to be segmented with good accuracy by scene labeling algorithms.

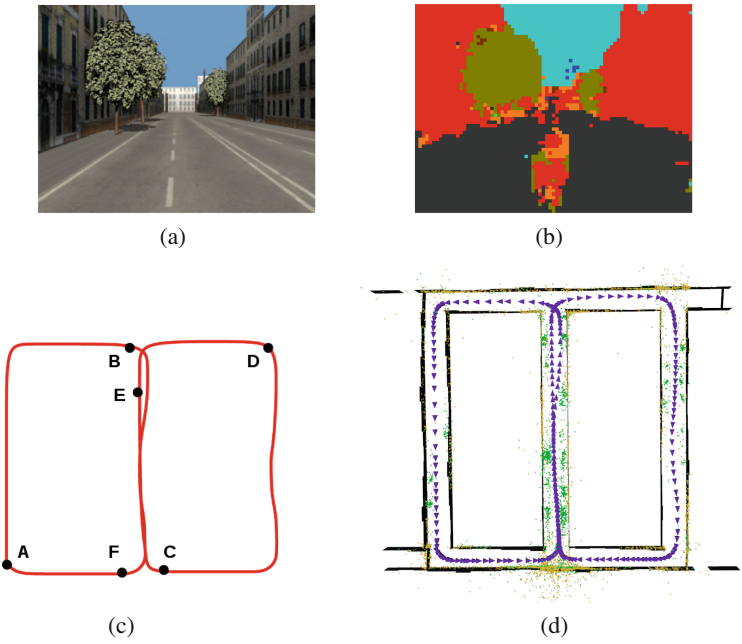


Fig. 2. (a) An example image of the synthetic sequence. (b) The result of the CNN (with the arg max taken) on the example image. Red pixels are those that correspond to the buildings. (c) The ground truth trajectory. The camera movement is given by A-B-C-D-E-F-A (d) the trajectory and the point cloud as refined by our method. The golden points are those that are classified as belonging to buildings. Other points are represented in green. (Color figure online)

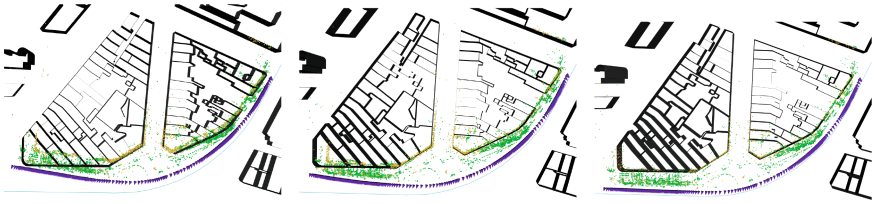
The sequence was ∼1200 m long and it included multiple loops. An example image from the sequence and its segmentation by the CNN is given in Fig. 2(a) and (b). The ground truth trajectory is given in Fig. 2(c). The trajectory and the point cloud as refined by our constrained BA approach is illustrated in Fig. 2(d). On this sequence, the mean translational error of our method was 1.3 cm, while the rotational error was 0.05 radians.

On this sequence, constrained BA without pixel-wise scene labeling fails. BA constrained to 3d building models with geometric segmentation of the point cloud (with ray-tracing and proximity criteria as in [6,8]) leads to a rapid deterioration of the geometric structure and ultimately to pose computation failure.

5.2 Real Data

In this section, we present BA results with and without scene labeling on a short but particularly challenging outdoors sequence[1], mainly because building

[1] We could not use the Kitti dataset, because it does not provide 3d building models.



(a) (b) (c)

Fig. 3. Comparison of reconstruction results. The blue curve represents ground truth. Golden points are those that have been classified as belonging to buildings. All the other points are shown in green. (a) The reconstruction we obtained without scene labeling (using a simple ray-tracing similar to the approach of [7] instead). (b) The result with scene labeling and classification using the arg max of the distributions. (c) The result we obtained with scene labeling and classification using the distribution $\mathfrak{D}$. (Color figure online)

Fig. 4. Building model reprojections after bundle adjustment on the real sequence. This examples show that our solution is robust to occlusions, which are omnipresent on that sequence. Once again, note that the building models suffer from inaccurate heights.

facades are often almost completely occluded by trees, billboards, etc. The camera is approximately orthogonal to the trajectory. Additionally, the height of the 3d building models we used in this experiment were inaccurate. Unfortunately, we only had access to the in-plane positional ground truth (shown as a blue curve in Fig. 3) but did not have access to altitude or orientation ground truth. Instead, we used the proper alignment of building contours with the projection of 3d building models as a criteria for evaluating the precision of each solution. Figure 3 shows a comparison between the result we obtained without and with scene labeling. Additionally, this figure shows the trajectory that we obtained when classifying the pixels using the arg max of their distribution instead of the Dirichlet distribution $\mathfrak{D}$. It can be seen that in that case, as when scene labeling is not used, the high number of false correspondences between points and buildings causes an inacceptable error. Figure 4 shows building models reprojections after bundle adjustment with our method.

6 Conclusion and Future Directions

In this paper, we demonstrated that important accuracy gains can result from incorporating scene labeling into constrained bundle adjustment. We filtered out poor segmentation results by modeling our particular CNN as a Dirichlet process.

This method proved to be more efficient than simply taking the arg max. The computational complexity of the segmentation module prevented us from reaching real-time performance in a sequential implementation. However, it is possible to run the segmentation algorithm on a dedicated thread, and update the on-line reconstruction as soon as a result becomes available (similar approaches for combining real-time slam with high-latency solutions exist [18]). Thus, we will direct our future efforts toward developing such an architecture, while independently optimizing our CNN implementation.

References

1. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Real time localization and 3D reconstruction. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 363–370. IEEE (2006)
2. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment — a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) IWVA 1999. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000). doi:10.1007/3-540-44480-7_21
3. Lhuillier, M.: Incremental fusion of structure-from-motion and GPS using constrained bundle adjustments. IEEE Trans. Pattern Anal. Mach. Intell. **34**(12), 2489–2495 (2012)
4. Lhuillier, M.: Fusion of GPS and structure-from-motion using constrained bundle adjustments. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3025–3032. IEEE (2011)
5. Leutenegger, S., Furgale, P.T., Rabaud, V., Chli, M., Konolige, K., Siegwart, R.: Keyframe-based visual-inertial SLAM using nonlinear optimization. In: Robotics: Science and Systems (2013)
6. Larnaout, D., Bourgeois, S., Gay-Bellile, V., Dhome, M.: Towards bundle adjustment with GIS constraints for online geo-localization of a vehicle in urban center. In: 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), pp. 348–355. IEEE (2012)
7. Lothe, P., Bourgeois, S., Dekeyser, F., Royer, E., Dhome, M.: Towards geographical referencing of monocular SLAM reconstruction using 3D city models: application to real-time accurate vision-based localization. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 2882–2889. IEEE (2009)
8. Tamaazousti, M., Gay-Bellile, V., Collette, S.N., Bourgeois, S., Dhome, M.: Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3073–3080. IEEE (2011)
9. Larnaout, D., Gay-Belllile, V., Bourgeois, S., Dhome, M.: Vision-based differential GPS: improving VSLAM/GPS fusion in urban environment with 3D building models. In: 2014 2nd International Conference on 3D Vision (3DV), vol. 1, pp. 432–439. IEEE (2014)
10. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1915–1929 (2013)
11. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)

12. Costante, G., Mancini, M., Valigi, P., Ciarfuglia, T.A.: Exploring representation learning with CNNs for frame-to-frame ego-motion estimation. IEEE Robot. Autom. Lett. **1**(1), 18–25 (2016)
13. Gao, X., Zhang, T.: Loop closure detection for visual SLAM systems using deep neural networks. In: 2015 34th Chinese Control Conference (CCC), pp. 5851–5856. IEEE (2015)
14. Majdik, A.L., Verda, D., Albers-Schoenberg, Y., Scaramuzza, D.: Micro air vehicle localization and position tracking from textured 3D cadastral models. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 920–927. IEEE (2014)
15. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(1–3), 503–528 (1989)
16. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: a matlab-like environment for machine learning. In: BigLearn, NIPS Workshop (2011)
17. Xu, P.: KITTI semantic segmentation. https://www.hds.utc.fr/xuphilip/dokuwiki/en/data
18. Oleynikova, H., Burri, M., Lynen, S., Siegwart, R.: Real-time visual-inertial localization for aerial and ground robots. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3079–3085. IEEE (2015)