Improved 3D Perception based on Color Monocular Camera for MAV exploiting Image Semantic Segmentation

Andrei Baraian and Sergiu Nedevschi
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Email: Sergiu.Nedevschi@cs.utcluj.ro

Abstract*—In this paper, we propose an improved 3D perception for MAV based on color monocular camera. We exploit the semantic segmentation of scene color images to derive semantic and geometric constraints on the 2D keypoints and 3D reconstruction. The high-level components of the system are ego-motion estimation of the drone and sparse 3D reconstruction and segmentation of the explored environment. The ego-motion and 3D structure components are highly coupled and any significant improvement in either one greatly influences the other one. The proposed method aims at reducing the outliers that come from dynamic objects by avoiding extraction of keypoints from entities that could potentially exhibit dynamic behavior (cars, trucks, pedestrians, etc.) exploiting the semantic segmentation of the image. Also, we derive a set of semantic and geometric constraints which improve the 3D reconstruction. The resulting 3D point cloud is semantically segmented by transfering the semantic class of the originating points from which it was triangulated. Ultimately, all these improvements have a great impact on the speed-up and accuracy of the optimization method used, in our case windowed bundle adjustment.

I. Introduction

Autonomous Micro Aerial Vehicles (MAVs) are getting more and more attention because of their versatility, flexibility and innovative ways of use. They are already used in a variety of applications, such as industrial processes, damage inspection, building assessment or delivery of goods. As drones get a higher level of autonomy, they need better perception of the surrounding environment, while maintaining their low power consumption and reduced weight. It has been proven that mounting different sensors on drones enables them to explore the environment, but some of these solutions require equipment that can be too heavy, expensive or having a high power consumption, such as a stereo setup or range sensors [1] [2].

Instead of these options, a more lightweight system can be used and still achieve great accuracy, composed of one camera and an Inertial Measurement Unit (IMU) [3]. Because of drone's agility given by the six degrees of freedom (DoF), a camera proves to be a good depth sensor since it can record measurements from different points of view and their uncertainties can converge to a good estimate of the ground truth. The problem of recovering the 3D structure of the explored environment and the camera poses at each frame is known as Structure from Motion [4], [5] in Computer Vision. Given the incremental way in which the 3D structure and the camera poses are estimated, the algorithm is partical case of SfM, known as Visual Odometry [6] and focuses on recovering especially the camera poses in real-time, resulting in a very sparse 3D structure.

One of the main problem that arises in Structure from Motion algorithms is the presence of outliers in various stages of the pipeline that can bring the system to an irrecoverable state [7]. They can arise in the feature extraction and matching process, in the motion estimation process or in the optimization step. To deal with them, RANSAC [8] is the most used algorithm to reject outliers. For RANSAC to be as robust as possible, we need to take sufficiently many measurements in order to hope that we have a high percentage of inliers. This results in increased processing time and also decreased accuracy in case of large number of outliers.

We propose an improved 3D perception algorithm that enforces constraints based on semantic classes on 2D points as well as on 3D points in the environment. We aim at improving the accuracy of the 3D reconstruction and the rigid body transformation between successive frames, that would ultimately lead to a decrease of drift propagation in ego-motion estimation. A consequence of these improvements would be increased speed-up and accuracy for windowed bundle adjustment.

A. Related Work

When recovering the camera poses and the 3D structure of the environment, there are two main methods: direct and feature-based methods.

Direct methods minimize the photometric error of corresponding pixels in subsequent frames, recovering the motion and the structure. It operates directly on image intensity values of the image, allowing to be robust even in low-textured areas, but being more expensive to compute than the reprojection error.

Feature-based methods first identify sparse, salient keypoints and then match them in the next frames either by robust feature descriptors or by tracking. The pose of the camera

978-1-7281-4914-1/19/$31.002019 European Union

and the structure are recovered by minimizing the reprojection error, usually using the Levenberg-Marquardt algorithm [9] [10].

In [11], the authors propose a semi-direct method of sparse image alignment, and they deal with outliers by using a Bayesian filter. A new 3D point is inserted after its depth filter has converged, requiring multiple measurements. However, especially in dynamic environments, the algorithm will have to take a lot of erronous measurements until it can reject the outliers and the number of tracked points may decrease below a threshold, after which camera pose cannot be recovered. They further use the same idea in [12], where they propose a dense mapping of the pixels to a point cloud, by knowing the transformation between camera poses from the earlier algorithm.

Another known algorithm is PTAM [13], relying on feature-based correspondence that performs motion estimation and mapping in parallel. However, it works predominantly in closed, small sized areas and is not supposed to be run in outdoor scenarios.

The above methods do not take into considerations any information regarding the semantics of the environment. In [14], the authors use the semantic cues from 2D images to constrain the 3D structure where multiview constraints are weak. They derive a Conditional Random Field in the 3D space to infer at the same time semantic information and occupancy. [15] also uses semantic information, but having as input RGB-D images.

In this type of applications, the type of the resulted point cloud is also very important. If Visual Odometry is the sole purpose of the application, then a few number of tracked points is enough, but the 3D point cloud resulted will be very sparse. Opposed to that are dense methods that take into consideration almost all pixels from an image [12], but their performance significantly drops.

B. Contributions and Outline

The proposed system is a monocular 3D perception algorithm that uses Structure from Motion techniques combined with the semantic segmentation of images to enforce semantic and geometric constraints on 2D keypoints as well as on 3D points in the resulting point cloud. The algorithm builds incrementally a point cloud that is locally optimized after a number of iterations.

To generate semantic segmented images, we use Airsim simulator for drones, an open-source research platform [16]. Airsim allows for retrieval at the same instance of scene image, semantic annotated image, depth and disparity image, IMU data and GPS coordinates, therefore making it an excellent tool for development and testing. To test the algorithm on real world images, there are current solutions [17] that deliver fast and accurate semantic segmentation and could be easily integrated in our system.

We improve the ego-motion estimation and the 3D structure by proposing the following methods. In the feature extraction phase, we impose the extracted features to lie on static objects and avoid objects that may exhibit dynamic behavior (cars, trucks, pedestriants, etc.). In the tracking phase, their correspondent keypoint must have the same semantic class, thus also reducing outliers from incorrect tracking on static objects. The resulting 3D points that we obtain are also semantically segmented by transferring the semantic class from the originating 2D pixels from which they were triangulated. Further on, we propose geometric constraints derived from semantic information of the 3D points, more explicitly estimating the hyperplane of the 3D points that have the semantic class of the ground, using RANSAC [8]. After estimating the hyperplane, we use it to enforce depth constraints for newly triangulated keypoints that have the semantic class of the ground. If the points have a large depth error with respect to the hyperplane, then they are discarded, otherwise they are constrained to reside at the correct depth.

These improvements also influence the speed and accuracy of a windowed bundle-adjustment algorithms and they even prove a viable alternative to such optimization frameworks. For performing bundle-adjustment, Ceres solver [18] was used.

Section II provides an overview of the algorithm and Section III describes data acquisition. Section IV is related to motion estimation using essential matrix for the first two frames and P3P for all subsequent frames, Section V describes 3D structure improvements, while Section VI presents the optimization method used. Section VII describes the experimental results.

II. System Overview

The proposed Structure from Motion pipeline is conceived to work on a drone navigating at 3 m/s, with camera oriented downard or at 45 degrees inclination. Fig. 1 provides an overview of the algorithm. Features are extracted in the first frame and then tracked in a second frame. A mask for possible dynamic objects is created and features are not extracted in those regions. The first set of outliers come from keypoints that have been tracked and do not have the same semantic class. The essential matrix is computed and based on how many outliers do not respect the computed transformation, the frame is either rejected and the same procedure is applied to the following frame or we consider it for the next step, triangulation.

After the first two frames are processed and we obtain the initial point cloud, we estimate the hyperplane of the ground points. For each new frame, we match existing 3D points from the point cloud with the 2D points from the originating view and the 2D points tracked in the new frame, obtaining 3D-to-2D correspondences [19], being able to solve the Perspective-n-Point (PnP) problem and get the camera motion. If the number of tracked 3D points falls below a threshold, the system extracts new keypoints which will be tracked and triangulated based on the estimated motions.

To achieve real-time processing, the system implements windowed (local) bundle adjustment. Whenever a new 3D point is inserted to the point cloud, it also has the semantic class of the originating 2D pixel from which it was triangu-

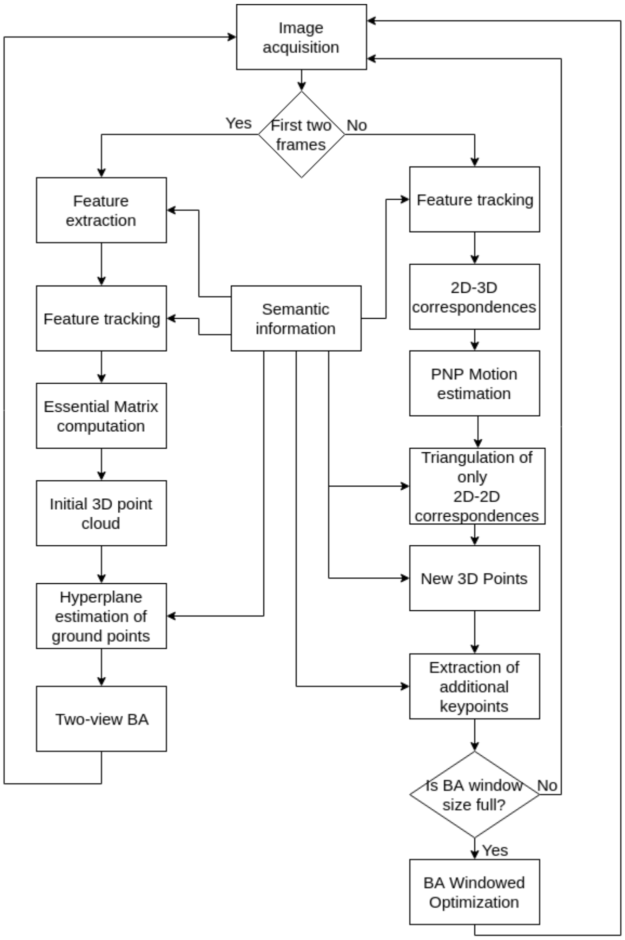


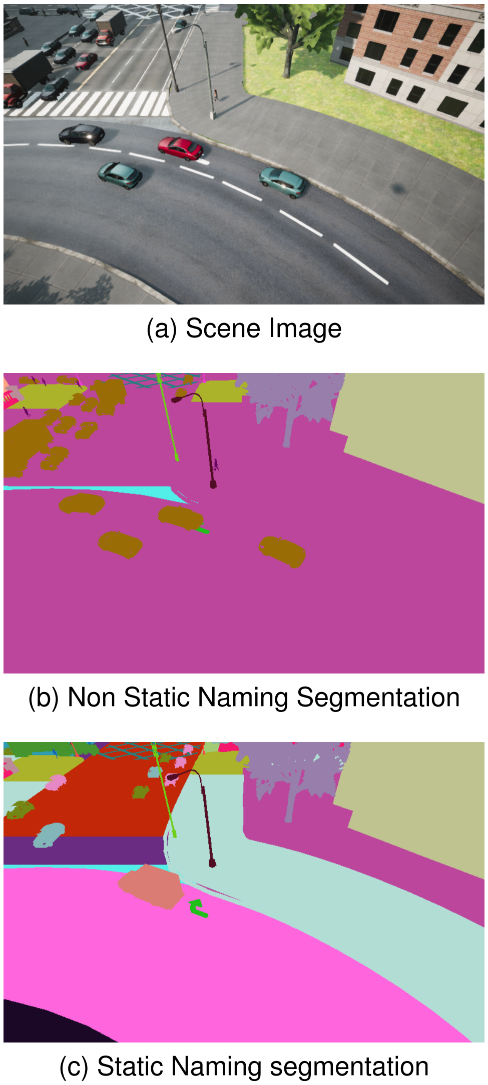
Fig. 1. System architecture.

lated, therefore the system can obtain a semantic segmented point cloud.

If a real semantic segmentation module would be used, like [17], the module can be decoupled from the pipeline to compute the segmentation of the acquired scene image in parallel with processing of the last acquired frame. That way, the delay caused by the semantic segmentation would appear only for the first frame.

III. Data acquisition

The AirSim simulator allows for retrieval of scene, segmentation, depth and disparity images at the same time instance. The way segmentation is done is based on mesh names, that are assigned by developers that create the Unreal environments to which AirSim is bidden. On the other hand, more meshes may share the same name, which often leads to inaccurate semantic segmentation, as it was our case when working with the City environment. Few workarounds exist, such as changing the mesh naming method. When using Static Mesh naming option, all vehicles share the same color and the pedestrian as well, but many other objects are not correctly segmented, such as the fence, grass and sidewalk sharing the same color with the ground (asphalt). Another method is to use the API provided for AirSim and manually set the labels for each mesh, but for closed-source environments, it is not possible to get mesh names.


(a) Scene Image

(b) Non Static Naming Segmentation

(c) Static Naming segmentation

Fig. 2. Diferences between semantic segmentation options in AirSim

This proved to be problematic when trying to enforce depth constraints on the ground, since points corresponding to the fence would have higher height than points corresponding to the ground, but they would still share the same semantic class. An alternative would be to use the Non Static Mesh naming option, where more objects are semantically segmented, but objects from the same semantic class are colored with different colors. However, for the sole purpose of testing, it can be assumed that objects from the same class may share a range of colors.

Especially in the case of cars, the results were not satisfactory, as pre-processing steps like region filling and dilations had to be performed, so that the car would be represented by a closed polygon.

IV. Motion Estimation and 3D Reconstruction

To estimate camera motion between frames, just for the first two frames we calculate the essential matrix, then we solve the PnP problem for each subsequent camera pose. This way, we do not have to concatenate successive transformations, since the reprojection error is with respect to already calculated 3D points, which are expressed relative to the first camera frame.

Our main contributions towards an improved ego-motion estimation and 3D reconstruction are using the semantic segmentation of input images. The semantic segmentation is exploited for dynamic features avoidance and for introducing geometric constraints, especially for planary surfaces. This way, we obtain an efficient method for outliers removal and enforcing depth constraints for new triangulated 3D points.

A. Feature extraction and tracking using semantic segmentation for essential matrix and initial 3D point cloud

To estimate the essential matrix between the first two matrices, the system extracts N strongest corners by Shi-Tomashi method, known as *goodFeatureToTrack* [20]. A mask is computed prior to the extraction step from the semantically segmented image, to avoid dynamic objects. We label as dynamic those objects that may exhibit dynamic behavior such as cars, pedestrians, animals, even if they may be static at that moment. The next step is tracking the keypoints extracted in the first frame, by using the Lucas-Kanade algorithm [2]. To deal with large motions, the tracking is applied in a coarse-to-fine scheme, by creating three image pyramid levels. The resulting tracked features are compared to have the same semantic class as the originating features and mismatches are discarded. Based on these correspondences, we estimate the rigid body transformation between the two frames using the essential matrix. The method uses the five-point solver in [19] and RANSAC for outlier rejection. We proceed with the pipeline only if the inliers ratio is bigger than 90%, otherwise we discard the second frame and start computing the tracking and essential matrix for the next frame.

After finding a reliable second frame and estimate the transformation, we use it to triangulate the tracked keypoints. For each point, the reprojection error is calculated and if it is higher than one pixel, the corresponding point is discarded. For each good 3D point, the semantic class and pixel color is saved, as well as the originating views in the form of a map, the key being the identifier of the frame and the value being an index in the features vector of that frame.

B. Perspective-n-Point motion estimation and incremental triangulation of new 3D points

Once having the initial 3D point cloud, 2D-to-3D correspondences can be found. When a new frame is captured, keypoints from the last frame are tracked into the new frame. At this point, a connection between the 2D keypoint in the last frame, the 3D point and the 2D tracked keypoint in the new frame can be established. This connections serve as input to the PnP problem, that recovers the new camera pose relative to the 3D points, hence it is relative to the other recovered camera poses. The pose is recovered by minimizing the reprojection error in the new frame. We use the AP3P method in OpenCV [21] rather than P3P [22], since it proves to be more robust and efficient and it is based on [23]. After estimating the camera pose, we triangulate tracked features for which we did not have a 2D-3D correspondence. This is the point where the system can evolve towards an irrecoverable state, since the new camera pose recovered is relative to the 3D points, hence any outlier may affect the new pose, which in turn will generate new 3D points that are not accurate. This is why adding depth constraints helps the system from not drifting too much. Any new 3D point having the semantic class of the ground, is tested against the estimated depth average of the ground and it is either discarded if the error is too large or constrained to reside on the plane if the error is small.

V. Refinement of 3D point cloud

A. Point cloud semantic segmentation

To obtain a semantic segmented point cloud, we transfer the semantic class of 2D pixels that have been triangulated and share the same semantic class, to the 3D resulted point. This step also improves the 3D structure since it does not triangulate 2D pixels that do not have the same semantic class, hence they represent outliers in the tracking phase. Moreover, having the semantic segmented cloud point means deriving geometric constraints for some semantic classes. New triangulated 3D points that are added to the point cloud can be tested against these constraints and can even be optimized to reside at the correct location.

B. Hyperplane estimation using RANSAC and semantic information for 3D point cloud refinment

One of the key aspects of any Visual Odometry algorithm is a good initialization. Assuming that the drone starts by looking at a reasonably large area of ground, we apply RANSAC to estimate the plane on which ground points should reside. We do this step after the initial triangulation. After applying RANSAC, we also filter the outliers. The same technique can be applied to sidewalks, which are a bit higher. This step can greatly improve the initialization, hence reducing the drift.

$$z_{constraint} = \frac{\sum\limits_{i \in \{Ground\_class\}} depth(p_i)}{n_{Ground\_class}}$$

The above equation calculates the depth constraint by averaging all ground points after RANSAC filtering has been performed. Moreover, this assumes that the road is planar and will not change its depth. This constraint will be used when triangulating new ground points.

VI. Improved Windowed Bundle Adjustment using semantic segmentation

To prevent the system from evolving to an irrecoverable state, but still maintain real-time processing, we use windowed bundle adjustment, a local optimization on the most recent *n* frames and the 3D points seen from those frames. Bundle adjustment is the process of joint optimizing the camera poses C_k and the 3D structure points by minimizing the reprojection error between the projection of 3D points X^i using the function g for projection and the 2D point p_k^i seen in frame k and a loss function ρ_i to reduce the influence of outliers.

$$\arg\min_{X^i, C_k} \sum_{i,k} \rho_i \left(\| p_k^i - g(X^i, C_k) \|^2 \right)$$

To solve such a system, Levenberg-Marquardt algorithm is used. Due to the reduced number of camera poses relative to the number of tracked points and the special structure of parameters in the non-linear function, this motivates for using the Schur complement trick [24]. As a loss function, we have used the Cauchy loss function.

$$\rho(s) = log(1 + s)$$

By performing the initialization with RANSAC and semantic segmentation and keeping the ground points as fixed, we improve the time to convergence and accuracy, since the number of parameters in the optimization steps decreases. For testing purposes of the BA, it can be checked if the optimized ground points still satisfy the depth constraint.

VII. Experimental Results

In this paper, we proposed an improved 3D perception system for micro-aerial vehicles based on monocular cameras. These improvements are aimed at ego-motion estimation and 3D sparse reconstruction in dynamic scenes, speed-up of windowed bundle adjustment and enhancing the 3D point cloud resulted with semantic information.

For testing the new methods, we have generated different datasets using the City environment in AirSim. This environment consists of scenes having dynamic entities such as cars, trucks, pedestrians, etc. The camera on the drone was oriented looking downward and at 45 degrees. In Fig. 3c we present the 3D point cloud which results from the 3D reconstruction along with the camera locations at each frame acquisition and in Fig. 3d we show the semantically segmented 3D point cloud.

In Fig. 4 we show how the ego-motion estimation and 3D structure are affected using various setups, without performing windowed bundle adjustment. It can be easily seen that the system starts to drift towards an irrecoverable state if no semantic information is used. The hyperplane estimation using the semantic information, combined with the derived geometric constraints can greatly reduce the drift and keep the system consistent, while filtering the dynamic objects also improves the result. In case of predominantly dynamic scenes, filtering of dynamic objects is of paramount importance. In the case of filtering tracked points based on their semantic class, we have found out that about 2-4% of the points are tracked incorrectly.

Fig. 5 describes the ego-motion drift. To calculate the error, we used AirSim API to set the trajectory of the drone on a horizontal line in the direction of Y axis, parallel with the hyperplane defined by the X and Y axis. Ideally, the drone should have a constant position on the Y-axis. The error is measured by calculating any drift from the constant position.

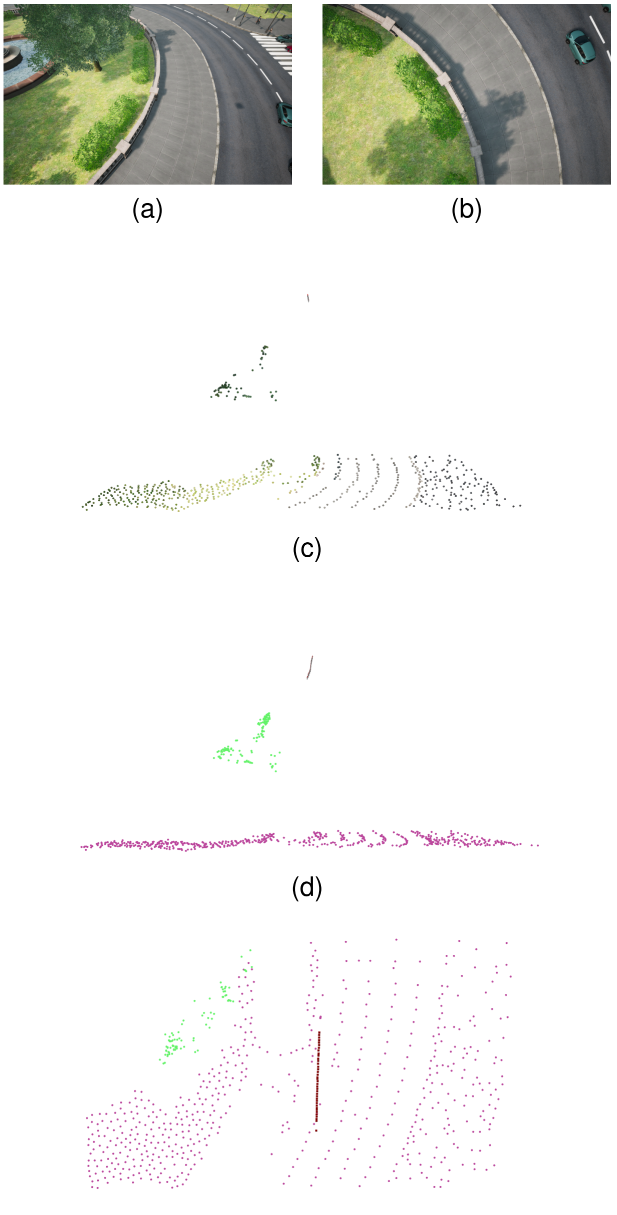

(a)
(b)
(c)
(d)
(e)

Fig. 3. Camera orientation: (a) 45 degree camera angle (b) downward looking camera. Point cloud result: (c) Point Cloud with pixel colors observed from 45 degrees (d) Semantic segmented point cloud (e) Point Cloud and camera poses seen from above

From the carried experiments, we deduce that a fusion between semantic constraints on 2D and 3D points and geometric constraints may greatly improve the system and may also represent a viable alternative to bundle adjustment. The total cost of BA without any improvements from semantic segmentation is around 400-500 ms per window. While the new methods improve the BA, the cost of BA is still quite expensive (around 250-300 ms per window, after improvements). The best tradeoff between performance and a high

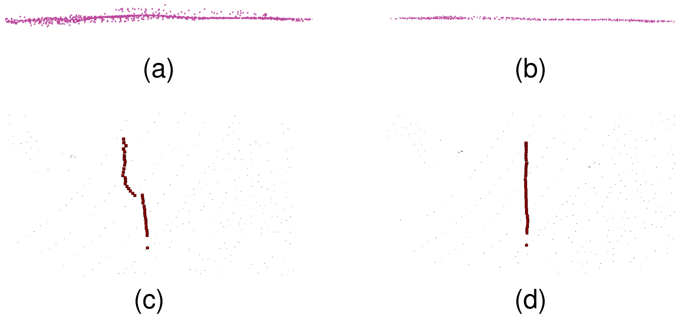


Fig. 4. Diferences between traditional Structure from Motion and using hyperplane estimation based on RANSAC and semantic classes. An improved 3D structure automatically results in a better ego-motion estimation.

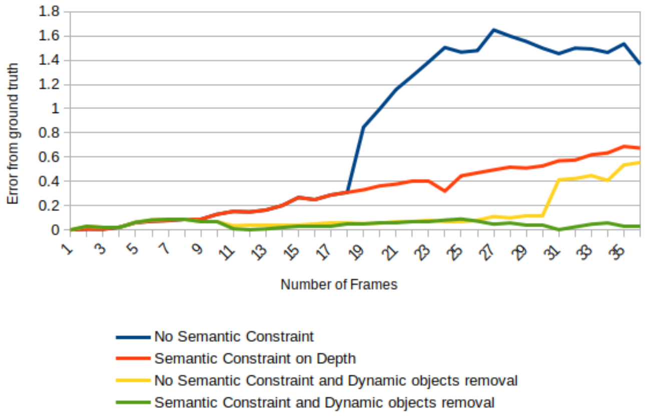


Fig. 5. Ego-motion error on Y-axis

level of accuracy is done using only the fusion of the semantic segmentation and geometric constraints. This method achieves about 5 frames per second processing time.

VIII. Conclusion

In this work, we propose an improved 3D perception from color monocular camera for MAV. The system exploits the semantic segmentation of input images for a better understanding of the reconstructed scene. The improved ego-motion deals with dynamic entities by using semantic segmentation and avoiding extraction of keypoints that reside on them. Further on, we transfer the semantic class of 2D points to the triangulated 3D points. Based on the semantic segmentation of the 3D point cloud, we can estimate the hyperplane describing the ground and can derive geometric constraints that newly triangulated points should respect. Having a semantic segmented 3D point cloud allows for better understanding and perception of the explored environment. As a result of these improvements, the speed-up and level of accuracy of windowed bundle adjustment are increased.

IX. Acknowledgment

This work was supported by the Integrated Semantic Visual Perception and Control for Autonomous Systems SEPCA, CNCS-UEFISCDI, PN-III-P4-ID-PCCF-2016-0180, grant no. 9/2018.

References

[1] A. Johnson, J. B. Collier, A. R. Klumpp, and A. Wolf, "Lidar-based hazard avoidance for safe landing on mars," *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, vol. 25, 11 2002.

[2] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework: Part 1," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-02-16, July 2002.

[3] C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza, "Continuous on-board monocular-visionbased elevation mapping applied to autonomous landing of micro aerial vehicles," vol. 2015, 05 2015.

[4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[5] S. Emami, D. Baggio, K. Ievgen, and N. Mahmood, *Mastering OpenCV with Practical Computer Vision Projects*, ser. Community experience distilled. Packt Publishing, 2012. [Online]. Available: https://books.google.ro/books?id=GXewmAEACAAJ

[6] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Automat. Mag.*, vol. 18, pp. 80–92, 12 2011.

[7] J. L. Schnberger and J. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4104–4113.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[9] K. Levenberg, "A method for the solution of certain problems in least squares." *Quaterly Journal on Applied Mathematics*, no. 2, pp. 164–168, 1944.

[10] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963. [Online]. Available: http://dx.doi.org/10.1137/0111030

[11] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 15–22.

[12] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[13] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 1–10. [Online]. Available: https://doi.org/10.1109/ISMAR.2007.4538852

[14] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 703–718.

[15] A. Hermans, G. Floros, and B. Leibe, "Dense 3d semantic mapping of indoor scenes from rgb-d images," 05 2014, pp. 2631–2638.

[16] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[17] A.Petrovai and S. Nedevschi, "Efficient instance and semantic segmentation for automated driving," in *2019 International Conference on Intelligent Vehicles (IV)*, June 2019, pp. 2153–2347.

[18] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[19] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004, pp. I–I.

[20] J. Shi and C. Tomasi, "Good features to track," Ithaca, NY, USA, Tech. Rep., 1993.

[21] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[22] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, Aug 2003.

[23] T. Ke and S. I. Roumeliotis, “An efficient algebraic solution to the perspective-three-point problem,” *CoRR*, vol. abs/1701.08237, 2017. [Online]. Available: http://arxiv.org/abs/1701.08237

[24] D. Brown, *A Solution to the General Problem of Multiple Station Analytical Stereotriangulation*, ser. RCA Data reducation technical report. D. Brown Associates, Incorporated, 1958. [Online]. Available: https://books.google.ro/books?id=FikPPwAACAAJ