3D Sensor-Based UAV Localization for Bridge Inspection

Burak Kakillioglu, Jiyang Wang and Senem Velipasalar
Department of Electrical Engineering and Computer Science
Syracuse University
Syracuse, NY, USA
{bkakilli,jwang127,svelipas}@syr.edu

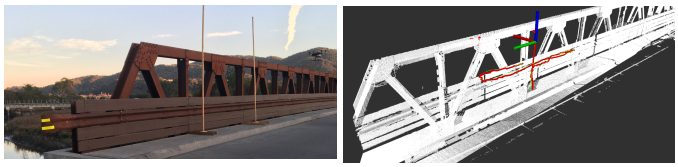
Alireza Janani and Edward Koch
Automodality INC.
San Rafael, CA, USA
{alireza,ed}@automodality.com

***Abstract*—Autonomous vehicles often benefit from the Global Positioning System (GPS) for navigational guidance as people do with their mobile phones or automobile radios. However, since GPS is not always available or reliable everywhere, autonomous vehicles need more reliable systems to understand where they are and where they should head to. Moreover, even though GPS is reliable, autonomous vehicles usually need extra sensors for more precise position estimation. In this work, we propose a localization method for autonomous Unmanned Aerial Vehicles (UAVs) for infrastructure health monitoring without relying on GPS data. The proposed method only depends on depth image frames from a 3D camera (Structure Sensor) and the 3D map of the structure. Captured 3D scenes are projected onto 2D binary images as templates, and matched with the 2D projection of relevant facade of the structure. Back-projections of matching regions are then used to calculate 3D translation (shift) as estimated position relative to the structure. Our method estimates position for each frame independently from others at a rate of 200Hz. Thus, the error does not accumulate with the traveled distance. The proposed approach provides promising results with mean Euclidean distance error of 13.4 cm and standard deviation of 8.4cm.**

I. INTRODUCTION

Small Unmanned Aerial Vehicles (UAVs), a.k.a drones, are getting increasingly popular and they have already been employed in many applications such as homeland security, rescue missions, disaster monitoring, film making, sports broadcasting, and journalism [1]–[3]. Reasons for this increasing popularity include their agility, ever decreasing costs, and the availability of more powerful onboard embedded processing capabilities. Another interesting application area of UAVs is inspection and health monitoring of civil infrastructures [4]–[7]. This type of application follows a well-defined procedure, and requires auditors to observe many, sometimes difficult to access parts of the structure. Thus, this application area is a perfect fit for UAVs; since they can access almost every part of the structure and carry out an autonomous inspection mission on a pre-defined path and procedure.

GPS is the main localization component used for autonomous missions. However, GPS data is either unavailable or unreliable in indoor environments as well as near or under some outdoor structures. Besides, precision of GPS is only 3-4 meters in good conditions unless new RTK-based modules, which are significantly more expensive, are used. The immediate approach for position estimation in GPS-denied areas is the use of odometry from inertial measurements. However, this approach is rarely used on real-time, practical systems, since it introduces a significant amount of noise drift, which increases with the distance traveled. To overcome the noise drifting, visual odometry can be fused with Inertial Measurement Units (IMUs) [8]–[11]. Many Simultaneous Localization and Mapping (SLAM)-based techniques use this approach, as discussed in Section II.



(a) Test site on San Rafael Bridge and our UAV platform.

(b) RViz 3D visualization software screen.

Fig. 1. Proposed approach precisely localizes the drone by estimating the vehicles position by using only the Structure Sensor. (b) shows drone position with the estimated path (red line) and the ground truth (green line)

In this paper, we propose a method for autonomous localization of UAVs in 3D space without relying on GPS data. Without loss of generality, we use the inspection and monitoring of bridges as an example application. The proposed method utilizes a Structure Sensor, which is a special, inexpensive and widely available camera that can acquire 3D model of the scene. The proposed method uses the existing 3D model of the space that it is intended to work in. By knowing the large 3D map of the whole space and continuously capturing the 3D region in front of the camera, our method robustly localizes the 3D position of the UAV on the map. The proposed method is designed to be used for 3D position estimation of UAVs, which can operate in large, GPS-denied, complex and cluttered environments/structures on a regular basis.

Our approach presented in this paper is applied to the autonomous bridge inspection using UAVs. In our experiments, the 3D scan of San Rafael Bridge at San Rafael, California is used (see Fig. 1(a)), and an actual UAV is employed to record data during several flights on the bridge. The UAV used for the flights can be seen in Fig. 1(a).

This work has been funded in part by National Science Foundation (NSF) under SBIR grant IIP-1746729 and CNS-1739748.

The presented approach enables the UAV to know its position and orientation with respect to the bridge when GPS data is unavailable or unreliable so that it can continue inspection without needing a stable GPS connection. The application areas can be extended to factory sites, mines, rescue missions in damaged structures, indoor surveillance and any other complex indoor environments.

II. RELATED WORK

Pose estimation of a vehicle, or a robot in general, is one of the main targets of the SLAM approaches, which is one of the most studied areas in robotics research in recent decades. There has been ever-increasing amount of work published in the SLAM area. G-mapping [12], Hector SLAM [13], Cartographer [14] are among the popular approaches in addition to numerous others. Grid maps are used together with Rao-Blackwellized Particle Filters for 2D SLAM in [8]. This approach is more robust compared to its predecessors thanks to the adaptive re-sampling approach [12]. A robust and lightweight system for online 3D SLAM is presented in [13]. It learns the map and efficiently localizes the robot by matching the scans captured by LIDAR. A lightweight and precise loop closure technique is presented for real-time 2D SLAM applications in [14]. It achieves 5 cm precision in real-time indoor loop closure by using branch-and-bound approach for computing scan-to-submap matches as constraints. However, most of the existing SLAM approaches are not readily applicable in complex 3D environments. Many of them depend on floor-plan-like 2D maps. If one takes different 2D slices, similar to those 2D maps, at every height of a complex 3D environment, each of those slices will be completely different from each other. If a UAV were to use one of these approaches, it must keep track of different maps at every altitude and potentially at different sensor orientations that are not parallel to the ground. On the other hand, there are some 3D approaches, which simply are extensions of their 2D counterparts. They, indeed, build a 3D map of the environment, but still localize themselves on a 2D map as they heavily depend on the assumption that robot (usually a UAV in 3D case) will be cruising at a certain height from the ground level. In fact, a real 3D SLAM technique must account for both mapping and localization in 3D. Therefore, these approaches would not be feasible for UAV applications, where the vehicle will navigate through complex structures at arbitrary altitudes.

3D Match [15] and SegMatch [16] are two promising approaches for 4-Degrees of Freedom (DoF) (6-DoF under level-world assumption) pose estimation. In fact, in our preliminary studies, we applied these approaches to our application as they seem to be great fits. They follow a similar pipeline with different techniques. The approach in [15] encodes the occupied point cloud regions by Truncated Distance Function and voxelizes into $30\times30\times30$ volumetric grids. Then, it uses a 3D-Convolutional Siamese Neural Network to match different pairs of voxelgrids. Finally, it filters out extra matchings by looking for geometric correspondences. The approach in [16], on the other hand, uses a region growing based algorithm to extract segments in point clouds. Then a global feature descriptor is used to describe those segments, and they are matched with a random forest classifier. Similarly, it applies a geometric verification step to remove false matches.

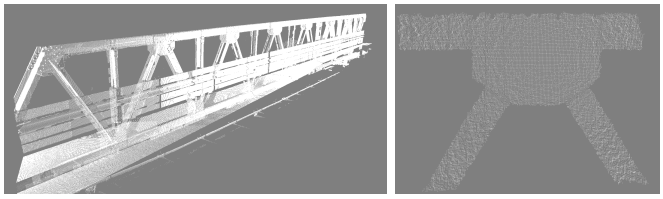


Fig. 2. 3D renders of San Rafael Bridge point cloud (left) and captured point cloud from Structure Sensor (right).

After using the approaches presented in [15] and [16] in our preliminary studies, we observed that these methods are not applicable to a scenario like bridge inspection due to several reasons. First, although the acquired information is a depth map or 3D point cloud, the nature of it is more like 2D, since the area being captured on the bridge is flat and spans the field of view of the camera. Therefore, the data looks like a flat region in 3D space. Second, the scale of the entire structure (in this case the entire bridge) is much larger compared to the field of view and what is captured by the sensor. Therefore, only small parts of the segments or feature points can be captured at a time. Third, features/segments are not unique. Since most bridges are engineered in a very repetitive manner, one feature or captured segment will possibly match correctly with many other locations on the bridge. Last but not least, computational complexity should be minimal due to the requirement of real-time on-board processing, and the aforementioned methods require substantial amount of resources such as high-end GPUs.

III. PROPOSED METHOD

We propose an approach for 3D position estimation of UAVs from 3D cameras, such as a Structure Sensor, without relying on GPS data. We present a robust algorithm, which will run onboard and provide position information at a sufficient rate at bridge inspection missions. Therefore, the algorithm must take the nature of the data into account and should be lightweight in order to run on an onboard processor. Our method relies on a pre-captured 3D map of the structure or area where the drone will monitor. A possible analogy can be a person locating himself on the city map by examining the landmarks nearby.

In a nutshell, our method applies 2D template matching on projected 3D data of bridge model for 3D localization. First, a depth image is acquired from a scene with the Structure Sensor installed on the UAV, and converted to a 3D point cloud. Point clouds of the scanned bridge model and the captured scene can be seen in Fig. 2. Then, the captured point cloud of the scene is projected onto a 2D binary projection image and matched on the large 2D binary projection of the entire bridge model. Mean of the 3D points which are projected on the same 2D pixel are calculated for each pixel and average 3D vector from model projection to scene projection is calculated as the estimated position.

A. Projection

The first step after acquiring the 3D point cloud is to project it onto a 2D binary image. An important consideration during this procedure is to perform it without loss of any information. A projection would normally cause loss of a dimension and quantization of data into a matrix structure. Once a projection is performed, the original data cannot be recovered. In our approach, we avoid this issue by defining a container for each pixel of the binary image to store the actual 3D points that projected to that pixel's coordinates. In other words, every pixel of the binary image is a mini point cloud, which keeps the 3D points that are projected on itself. As a result, the combination of these containers or mini point clouds will give the actual 3D point cloud. By doing this, we know exactly which 3D points belong to which pixel on the binary image after projection, and the 3D point cloud can be fully recovered. Projection $\boldsymbol{P'}$ of point cloud $\boldsymbol{P}_{N\times 3}$ onto projection plane $ax+by+cz+d=0$ is calculated as follows:

$$\vec{\boldsymbol{n}} = [a,b,c] \qquad \vec{\boldsymbol{d}}_{\boldsymbol{N\times 1}} = \frac{\boldsymbol{P}_{\boldsymbol{N\times 3}} \cdot \vec{\boldsymbol{n}}}{\|\vec{\boldsymbol{n}}\|}$$

$$\boldsymbol{V}_{\boldsymbol{N\times 3}} = \vec{\boldsymbol{d}}_{\boldsymbol{N\times 1}} \cdot \vec{\boldsymbol{n}} \qquad \boldsymbol{P'} = \boldsymbol{P} - \boldsymbol{V}$$

In our experiments, we used $y=0$ as projection plane.

Our assumption for projection of the bridge model is that the drone's heading is fixed to the bridge and the side of inspection is known. More specifically, the heading is known with respect to the bridge, which enables the sensor projections to be aligned with the bridge model. We project only the visible surface of the model. Surface points are the ones that lie in the median value of an empirically defined depth from the visible surface of the model.

B. Projection Matching

There are several reasons that we employ 2D alignment instead of the 3D approaches, such as [15] and [16]. In order to align/register two sets of point cloud data, first, some 3D features should be calculated in both point clouds and these features should be matched across the two clouds. Descriptions of these features must be invariant to rotation, and they should be replicable and unique. The nature of data in our application violates some of these criteria, which makes it hard to extract reliable features. Many identical features would be extracted because of repetitive patterns. Because of limited visible surface, it gets harder to replicate the same feature description for every location. Besides, the onboard operation criteria limits the computational resources, and this makes it harder to employ and process 3D approaches.

(a) Scene projection

(b) Matching result

Fig. 3. Scene projection image and matching result is shown as the black bounding box on the projection image of the bridge model

An OpenCV implementation of correlation coefficient-based template matching algorithm is used to match the projections [17]. The scene projection, which is the 2D binary projection of the point cloud captured by the Structure Sensor, is used as the template to be matched on the projection of the bridge model. Projection matching step returns a bounding box where the scene projection best fits on the model projection. This bounding box is then fed into position estimation step that correlates mapped points in the bounding box region.

One parameter that can be tuned for projection matching is the area of search region. As can be seen in Fig. 3, the bridge has a repetitive pattern, and the projection of the captured part can perfectly fit to many different sections of the bridge. In order to prevent this, and reduce computational burden, we limit the area that a scene can be matched by defining a search region around the previously matched position, and applying projection matching only in the limited search region. The initial search window is provided manually in our experiments.

C. Position Estimation

Projection matching registers captured projection onto model projection. This will provide a rough positioning as a form of 2D bounding box. In order to fully register captured point cloud onto model point cloud in 3D, and to estimate the 3D position of the vehicle, we make use of the correspondences of the points that are stored in the container of each 2D pixel location.

As the first step, the mean of the points which are projected on the same pixel are calculated for all pixels. These means are depicted in Fig. 4 as blue and green points for model image and scene projection image, respectively. Then, the Euclidean distance vector between each corresponding mean points is calculated. Since our assumption before this procedure is that orientations of both point clouds are already matching, captured point cloud can be viewed as a translated version of a patch on the model, which is parallel to its original position in 3D space. Thus, distance vectors must be very close to each other; or in the ideal case they must be the same vector. Therefore, as the final step, the average of all distance vectors is calculated as the estimated position by the following:

$$\vec{p}_{ij} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{C}_{ijk} \qquad \vec{P}_E = \frac{1}{R^2}\sum_{i}^{R}\sum_{j}^{R} \vec{p}^{\,M}_{ij} - \vec{p}^{\,S}_{ij}$$

where R is the resolution of the projection, $\boldsymbol{C}_{ijk}$ is the k^{th} 3D point in the container $\boldsymbol{C}_{ij}$ with K points at location (i,j) on

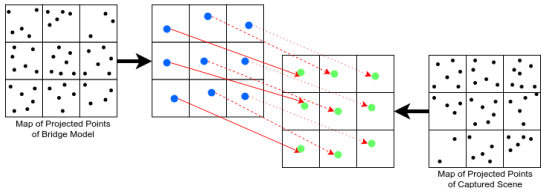


Fig. 4. Illustration of the position estimation from matched 2D binary projections. Each binary pixel holds a point container, which keeps the points that are projected on the pixel. Mean of each container is depicted as blue and green points for model and captured scene, respectively. The mean (average) of the difference of the mean tables is defined as the estimated position.

the binary images and $\vec{p}_{ij}$ is the mean point of the container at pixel location (i,j). M and S denote model projection and scene projection, respectively.

Our algorithm provides independent and absolute position estimation for each captured point cloud. In other words, it does not depend on the previous captures and the output is the exact position of the drone with respect to the bridge model. Therefore, there is no need for loop closure and map correction as opposed to most SLAM approaches. In contrast to most prior work, our proposed method does not suffer from drift or error accumulation as the traveled distance increases.

IV. EXPERIMENTAL RESULTS

A. Platform and Software

Our UAV platform is used for data collection. It is composed of the DJI Matrice 210 drone, J130 with TX2, a forward-facing Hokuyo UST-10LX laser range finder, and a forward-facing Structure Sensor. The J130 communicates with the DJI Drone over a TTL cable and DJI OSDK.

For data collection, the drone is manually flown to the position of the interest on San Rafael Bridge, and the logging process is initiated by pressing of a button on the remote. Upon receiving the log signal, the logging is initiated and the data in the form of bag files is recorded. The 3D scan of the San Rafael Bridge is obtained with the same platform and a similar procedure.

We have collected a set of five different flight data for experimental results. Duration of flights ranges between 30 to 150 seconds and they contain a total of 3590 frames (point clouds) from all flights. UAV has flown in a range of 1 to 10 meters in each XYZ direction with different trajectories, seen in Fig. 5.

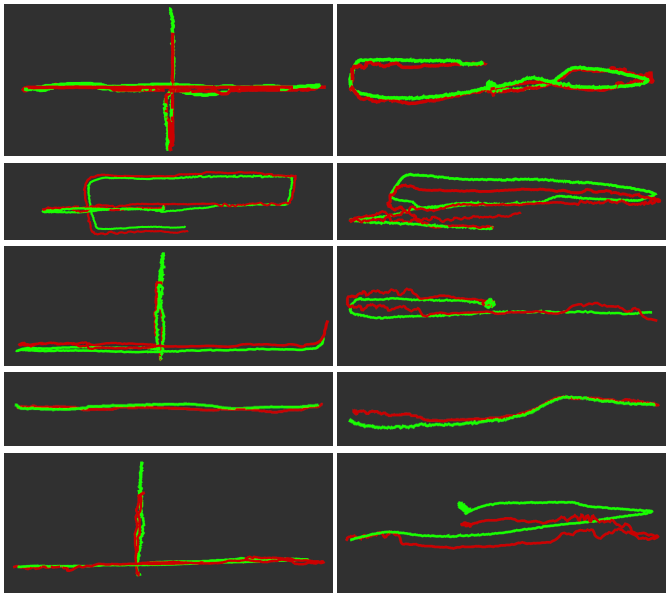
B. Obtaining Ground Truth

In order to obtain the ground truth information for the UAV position, we used Hokuyo 2D laser scanner sensor and laser altimeter onboard. The altitude (Z) is provided by the altimeter. To obtain the XY position, two vertical poles were placed on the bridge at both sides of the drone before takeoff. These poles are positioned sufficiently away from any object so that they can be easily identified and tracked. These poles can be seen in Fig. 1(a).

It should be noted that the proposed algorithm does not use and does not need poles for localization. They are only used to obtain the ground truth, and validate the proposed localization approach.

C. Data and Localization Module

Bag files in ROS are flashbacks of real-time experiments and provide the benefit of repeating the same experiment over and over again. In our experiments, we recorded our data as bag files, which contain series of point clouds from structure sensor as well as other necessary sensor data over a period of time. Using these recordings, we are able to replicate the experiments and tune the parameters without actually flying so many times.



Side view (XZ). +X direction is towards right and +Z direction is towards up on the paper.

Top-down view (XY). +X direction is towards right and +Y direction is towards up on the paper.

Fig. 5. Resulting drone flight paths of five different flight experiments from side (XZ) and top-down (XY) views. Green and red lines are ground truth and estimated paths, respectively. Figures span approximately 10 meters-wide length in the real world. The origin $(0, 0, 0)$ is defined as the initial position of the drone and the bridge is 1 to 2 meters away in +Y direction from the origin depending on the experiment.

We created a ROS package for this project. All necessary data stored in bag files are streamed into the package. After loading the bridge model, and creating its projection, the algorithm only depends on the point cloud data from structure sensor for localization. As mentioned above, the steps described in Section IV-B are only used for ground truth generation and evaluation purposes.

D. Discussion

What we have achieved in this project is to obtain a highly accurate 3D position estimation with a fixed heading assumption. Our experimental results show that the proposed algorithm is able to accurately estimate the position of the drone. Fig. 5 shows qualitative results of our approach for five different flights with different flight trajectories. Green line represents the ground truth path and red line represents the estimated positions. In each of the five subfigures, left part shows paths from side view whereas right part shows paths from top-down view. It can be observed from Fig. 5 that in some experiments, estimated path in the side-view does not reach the highest point where ground truth hits. It is because the structure sensor looses view of the bridge, and it provides an empty point cloud. In this case, the algorithm does not update the position and waits for sufficient view of the bridge. Once it gets a clear view, it continues the estimation process from where it left off.

Since the algorithm estimates the position for each frame independently from others, we did not carry out an error

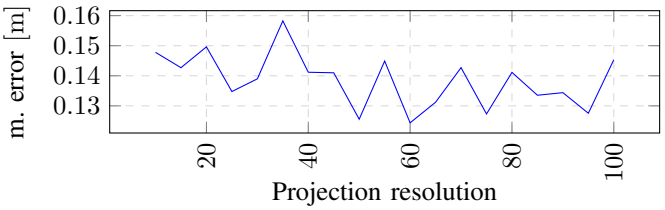


Fig. 6. Measuring the effect of increasing projection resolution on the mean position error (Averaged over all experiments).

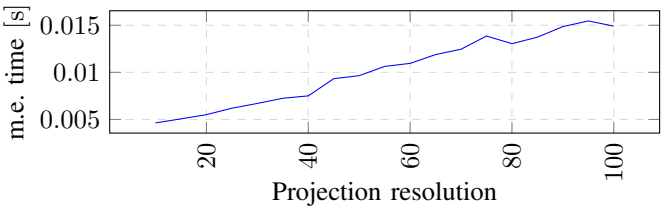


Fig. 7. Mean elapsed time linearly increases with projection resolution factor.

propagation analysis over time. Instead, we measured the mean and the standard deviation of distance error between the estimated position and ground truth for all frames. In some cases, algorithm fails and reports high errors. These cases only happen when the Structure Sensor completely loses the sight of the bridge. When these outliers are excluded, we measured a mean error of 13.4 cm with 8.4 cm standard deviation over the set of 3590 frames. It should be emphasized that these values are independent of the traveled distance, since our proposed approach treats each captured 3D scene individually and independent from the others. Furthermore, we analyzed the effect of projection resolution factor on the mean distance error. Fig. 6 shows the result.

We analyzed the robustness of our approach as it is one of the main criteria of our application. Average computation time of this approach with 20×20 projection resolution is as low as 5 ms, which means that the position can be updated at a rate of 200 Hz. 3.2 ms processing time is reported in [15] for 3D feature matching. However, they also report that the complete procedure of registration of two surfaces takes in the order of minutes on a high-end desktop with GPU. [16] is more on par with our work in terms of the processing time. They report an average rate of 10.5Hz localization performance. Yet, our procedure runs 20 times faster without requiring a GPU. We also made an analysis to see how projection resolution affects the computational complexity. It is observed from Fig. 7 that the projection resolution is inversely proportional to the overall computation time of the algorithm.

We have also observed that increasing the projection resolution does not significantly reduce the mean error. Besides, as discussed above, smaller resolution decreases the computational demand. Thus, we have found that 20×20 projection resolution is a sweet spot for our bridge inspection application as it is faster to compute and shows promising performance.

V. CONCLUSION

Accurate and reliable localization is crucial for autonomous operations of UAVs. In this work, we have presented an algorithm for 3D position estimation of UAVs from 3D sensor data without relying on GPS data. We only used 3D point cloud data from the UAV and 3D point cloud model of the structure, which is a bridge in our application, to estimate the position. Differently from the prior work, our method does not suffer from drift or error accumulation as traveled distance increases, and it can work in and around complex environments, such as bridges.

ACKNOWLEDGMENT

Authors would like to thank other members of Automodality team including Dan Hennege, Jimmy Halliday and Aaron Singer for their valuable contributions to this project.

REFERENCES

[1] G. Pajares, "Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs)," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–330, 2015.

[2] G. Cai, B. M. Chen, and T. H. Lee, "An overview on development of miniature unmanned rotorcraft systems," *Frontiers of Electrical and Electronic Engineering in China*, vol. 5, no. 1, pp. 1–14, 2010.

[3] A. L. Salih, M. Moghavvemi, H. A. Mohamed, and K. S. Gaeid, "Modelling and pid controller design for a quadrotor unmanned air vehicle," in *Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on*, vol. 1. IEEE, 2010, pp. 1–5.

[4] Y. Ham, K. K. Han, J. J. Lin, and M. Golparvar-Fard, "Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (uavs): a review of related works," *Visualization in Engineering*, vol. 4, no. 1, p. 1, 2016.

[5] B. Kakillioglu, S. Velipasalar, and T. Rakha, "Autonomous heat leakage detection from unmanned aerial vehicle-mounted thermal cameras," in *Proceedings of the 12th International Conference on Distributed Smart Cameras*. ACM, 2018, p. 11.

[6] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications," *J. Commun*, vol. 9, no. 9, pp. 687–692, 2014.

[7] S. Sankarasrinivasan, E. Balasubramanian, K. Karthik, U. Chandrasekar, and R. Gupta, "Health monitoring of civil structures with integrated uav and image processing system," *Procedia Computer Science*, vol. 54, pp. 508–515, 2015.

[8] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of imu and vision for absolute scale estimation in monocular slam," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[9] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," in *Robotics research*. Springer, 2010, pp. 201–212.

[10] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *null*. IEEE, 2003, p. 1403.

[11] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[12] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[13] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[14] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1271–1278.

[15] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017.

[16] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5266–5272.

[17] J. P. Lewis, "Fast template matching," in *Vision interface*, vol. 95, no. 120123, 1995, pp. 15–19.