Computer & Robotics

RGB-D SLAM Technique for an Indoor UAV Robot using the Levenberg-Marquardt Optimization Approach

Navid Dinarvand [a], Mohammad Norouzi [b,*], Mohamad Dosaranian-Moghadam [c]

[a] *Faculty of Electrical, Biomedical and Mechatronics Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

[b] *Faculty of Electrical, Biomedical and Mechatronics Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

[c] *Faculty of Electrical, Biomedical and Mechatronics Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

Received 06 June 2022; Accepted 11 November 2022

Abstract

Simultaneous localization and mapping (SLAM) technique is a practical approach for unmanned aerial vehicles (UAVs) to position themselves in unknown zones. In a structured arena with sufficient landmarks and enough lighting, the performance of the existing algorithms is satisfactory. But in a typical indoor field and in absence of GPS signal and poor texture and insufficient lighting, the SLAM would be unstable for navigation owing to the lack of features. In this article's suggested technique, the accuracy and resilience in many unknown situations (including dynamic and static ones) are enhanced by extracting edge and corner features instead of lone point features. A pre-processing block is intended to improve picture frames captured by the RGB-D sensor put on a robot with subpar characteristics. Using a predefined distance function, we filter out dynamic characteristics and solve dynamic issues in the same manner as static problems. Real-time use of our suggested strategy effectively reduces the influence of outliers and moving objects on the SLAM. This improves the accuracy of the procedure's computing output significantly. We validated our findings using data from the Technical University of Munich (TUM) to evaluate the proposed method. Additionally, our developed UAV is utilized for testing as well. The results of the trials indicate that the suggested approach is more precise and less susceptible to changes and system noise than the existing methods.

Keywords: Robot navigation, Rgb-d slam, Graph optimization, Indoor UAV, Outlier data reduction

1.Introduction and Related Works

Simultaneous Localization and Mapping (SLAM) [1] is a fundamental approach in many domains of autonomous navigation, including industry, medicine, agriculture, the military, mining, and search and rescue [2]. Initially, visual SLAM relied on affordable cameras that provided a plethora of information about the environment and enabled perfect localization [3]. During the last couple of decades, visual SLAM (V-SLAM) has made significant advancements. This has led to its extensive use in autonomous vehicles and drones [4-6].

The V-SLAM can be implemented as a real-time EKF SLAM problem. This solution is purely based on a single perspective camera and there is no information of laser scanner, odometry or GPS data. The problem is in fact similar to the Multiview Structure-from-Motion (SfM) practice where the system is employing just a single camera and tries to discover the camera motion and positions of features

* Corresponding Author. Email: mh.norouzi@gmail.com

in the arena. While in V-SLAM the features' positions are considered indefinite and a probabilistic framework would be considered to deal with this uncertainty. Consequently, the initial guess of the position of the features in commencement of system is quite essential. That makes the feature extraction a vital effort in success of the whole process. To implement EKF V-SLAM two functions of motion and measurement update should be formulated. The camera information would be expressed by 13 parameters where three for the position *r*, six for the translational and angular velocities (i.e. v and ω) and another four for the orientation-quaternion q i.e. $\mathbf{x}_t = [r_t, q_t, v_t, \omega_t]^T$. The dimension of the state vector in the EKF V-SLAM would be extended by 3*n* for n features' position as well. In the same way of the standard EKF SLAM, we form a state vector that includes all above mentioned information including camera pose, features position and camera velocity that can be state as $y_t = [x_t, m_0, m_1, ..., m_{n-1}]^T$. The Graph-Based SLAM formulation would be provided in section 2.1 for more problem statement of the work presented here.

As mentioned earlier, the odometry information is not available in V-SLAM for prediction of the next camera position. One solution in the literature is to consider a constant velocity model and computing the position of the camera at time *t* between two consecutive frames by integrating the motion starting at time *t-1*, assuming that the velocity between two consecutive frames remains constant. Note that still we consider uncertainties in translational and angular accelerations. These two parameters are modelled as zero mean Gaussian distributions. The reobservation of features is key solution in the camera pose correction phase i.e., the measurement update. The new features should be initialized and added to the map as the algorithm progresses. As it is apparent, the robust and reliable feature extraction is very significant for accomplishment of V-SLAM. The point features can be extracted directly from the pixels nevertheless more effective approaches employ feature descriptors. A measurement function is needed to compute the predicted observations as well as to predict the new position of features after the motion update. More details on these derivations could be find in [7]. The Graph-based SLAM is by means of the similar principles. Just as EKF SLAM, Graph SLAM must process incoming odometry data and observations. However, in Graph SLAM the correction of the map and the vehicle pose is moved into a separate step, the optimization. The computational complexity of the optimization is linear in the number of edges, so depending on the number of edges, the optimization may take a long time. Because of this, the implementation of the Graph SLAM algorithm is parallelized. The key idea is to move the optimization onto a second thread so that it can run in the background, while the algorithm is still collecting all incoming data.

Depending on the data employed by the system, visual SLAM might be characterized as either direct or indirect. In the direct technique, the system resolves the sensor's movement by minimizing the intensity and brightness differences between projected pixels and landmarks. In addition, direct techniques rely on the notion of photometric independence. In addition, such an algorithm is limited by the nonconvexity of gray levels. In contrast, direct approaches may be more accurate, but their processing complexity makes them less popular in real-world circumstances.

Feature-based techniques, in contrast, extract important points from an image and establish correspondences between landmarks and key points [7]. In this approach, the matched pairs and the calculation of the reprojection error terms are two crucial procedures that define the displacement result. The bulk of pairings is managed by utilizing feature matching methods. Therefore, they have strict standards for the possible compatibility of the qualities. Decrease the pixel gap between projected and recognized critical spots to improve camera motion. In the bundle adjustment (BA) [8] optimization, the pixel distance is referred to as the reprojection error.

Parallel Tracking and Mapping (PTAM) [9] is an example of early research using the feature point method. PTAM used two threads to perform motion estimate and mapping, enabling accurate real-time estimation. ORB-SLAM (Oriented FAST and Rotated BRIEF - SLAM) [10] has achieved a great deal of attention since it is free, open-source, accurate, and effective. This is one of the most often used SLAM system evaluation benchmarks. In addition, several studies have shown the limitations of feature points. Because a line gives substantially more geometrical and structural information about its surroundings, it has increased dependability in studies such as [11–15].

Feature extraction is a fundamental concept in machine vision that has not been used as the primary function in visual SLAM. With the visual SLAM system's introduction, several studies have emphasized feature extraction. Scale-Invariant Feature Transform (SIFT) [16], Speeded Up Robust Features (SURF) [17], and ORB [18] are some of the most often used feature detection and descriptors. This consists of invariants for the viewpoint, size, and rotation. Researchers have become more aware of the limitations inherent in the design of standard feature approaches. Therefore, researchers have devised a method for selecting the optimal localization characteristics during the SLAM estimation operation.

Similar to this, Zhang et al. [19] propose an additional filtering method based on prominent features to reduce mistakes successfully. They select components that contribute the most in terms of spatial and temporal factors to simplify computations during bundle adjustment at the expense of robustness. Yu et al. [20] introduce a unique RGB-D viewpoint invariant feature transform (PIFT). Researchers state that a single 2D image contains "false features" that cannot be identified or eliminated due to the lack of spatial data.

Vision-based SLAM systems are primarily focused on navigation utilizing a single camera in static circumstances. In contrast, the actual world has moving objects. While there are effective methods for recognizing and eliminating dynamic points using outliers, typical SLAM algorithms drift when the scene's objects undergo significant change. In addition, recreating the route of an item is an essential navigation job that is difficult to execute with a single RGB camera. We provide a remedy for overcoming these problems.

Our research focuses on RGB-D SLAM using an optimization strategy to improve performance in dynamic situations. Most contemporary research on feature-based SLAM disregards the impact of backdrop alteration on features. Our suggested solution utilizes a preprocessing step to mitigate the effect of outliers on SLAM performance. In general, two processes may be distinguished when visual characteristics are employed: Identifying topics of interest from various sources is the first step. Calculate the feature descriptors of the selected point, which are often derived from the surrounding environment. The "matching problem" is solved by robots using descriptors to determine if a landmark in their surroundings is one, they have seen previously or is new. These approaches assume that information about a point's surroundings may be utilized to reflect the point's qualities correctly. However, the local knowledge of the point may also be affected when the perspective moves due to a significant change in the background. Thus, the descriptor may not accurately describe the feature at the object's edge. In this situation, distinct points may not correspond to similar-looking backgrounds, and a feature point may not conform to its description.

In this research, we apply the following approaches to improve the performance of real-time applications:

- Errors are reduced by fusing depth frames for feature recognition and keyframe selection.
- By assigning a quality score to the input frames and selecting a threshold, we could boost inliers and reduce outliers.
- We employ weighted least squared (WLS) residuals to decrease the impact of poor

observations and boost the weight of excellent ones.

- Levenberg-Marquardt optimization is used for more stable optimization than Gauss-Newton optimization.
- Our suggested strategy was shown to be accurate and robust by testing with a restricted number of sources.

Experiments on the TUM RGB-D dataset [21] and demonstrations demonstrate that our proposed technique improves the precision of the system.

2. System Overview

Our proposed SLAM system based on feature point selection utilizing the distance function is shown in Figure 1. At each frame, the algorithm performs the steps as follows:

1. The histogram of frames with insufficient features has been equalized.
2. Equalized frames eliminate outliers, simplifying the cost function.
3. The quality of the filtered frames is then calculated using the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE), and some frames are removed based on the threshold.
4. The feature points are determined using Good Feature to Track (GFTT) and Binary Robust Independent Elementary Features (BRIEF) detectors.
5. The distance function is calculated for feature points.
6. Compare the distance values for the sequence to identify if a point is static or dynamic.
7. Calculate the camera 3D motion for each frame using the information contained in a static feature point.
8. We are detecting whether a feature has returned to a previously visited position.
9. The Levenberg-Marquardt algorithm is used to optimize graph structure.

The geometry of sequences captured by an RGB-D sensor could be estimated.

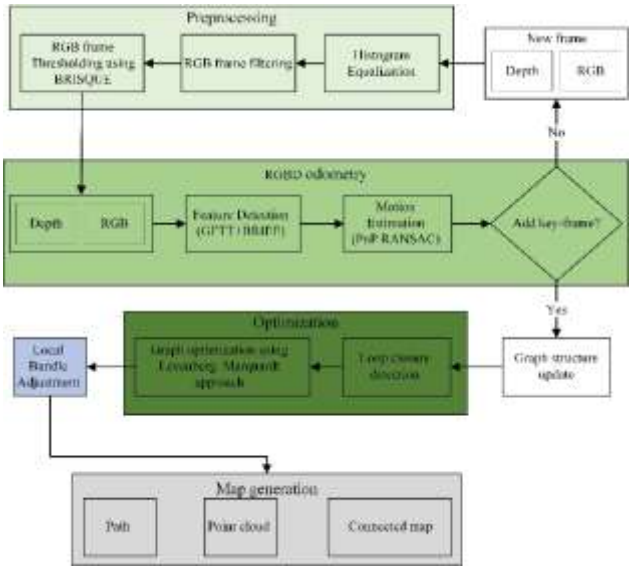

Fig. 1. The hybrid structure of visual and RGB-D SLAM techniques

2.1. Problem Statement

We assume a low dynamic environment and develop equations based on Gaussian noise model and non-rotational movement. Here, we will demonstrate the effect of dynamic foreground objects on motion estimation. Most visual SLAM algorithms assume a static environment and apply bundle adjustment.

2.2. Graph-Based SLAM Formulation

Graph-based SLAM is SLAM problem using graph approach which build the graph and find node configuration that minimized the overall error. Node configuration consists of nodes as robot poses or landmarks and edges as constraint between robot poses (nodes).

Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)^T$ denote robot position, where $\mathbf{x}_i$ describes the pose of node i. Moreover, z_{ij} denotes observation between node i, j and Ω_{ij} is the information matrix of observation between the node i, j It should be considered that observation is transformation that makes the observations acquired from i maximally overlap with the observation acquired from j. Based on above notations, $\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is the prediction of observation given a configuration

of the nodes $\mathbf{x}_i$, $\mathbf{x}_j$. The cost function γ_{ij} of a measurement z_{ij} is therefore

$$\gamma_{ij} \propto \left[\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right]^T \Omega_{ij} \left[\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right] \quad (1)$$

Furthermore, there are drifts between sensor measurement and motion estimation $\mathbf{e}\left(\mathbf{x}_i, \mathbf{x}_j, z_{ij}\right)$ which can be formulated as follows:

$$\mathbf{e}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) \quad (2)$$

Based on maximum likelihood approach, to achieve optimum node configuration $\mathbf{x}^*$ following equation should be solved:

$$\mathbf{x}^* = \underset{\mathbf{x}}{argmin}\, \gamma(\mathbf{x}) \quad (3)$$

Where $\gamma(\mathbf{x}) = \sum_{(i,j)} \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}$. If initial guess of the robot poses is good, (3) will be Solved by convergence numerically using Levenberg-Marquardt optimization approach.

The estimated robot and landmark postures are a subset of the graph SLAM structure and are calculated via bundle adjustment. Observing landmarks one at a time may be expressed as:

$$z_{ik} = \gamma\left(\mathbf{x}_{ik}\right) + \eta_{ik} \quad (4)$$

Where $\gamma(\mathrm{x}ik)$, η_{ik} are the measurement model, random Gaussian noise. To find the states X maximizing the probability of convergence, the Weighted Least Square approach is applied to the problem and could be written as

$$\begin{aligned} \mathrm{WLS}(\theta) &= (\hat{Z} - z(x,\theta))^T \cdot w \cdot (\hat{Z} - z(x,\theta)) \\ &= \sum_i^n W \cdot \left(\hat{Z}_i - z\left(x_i, \theta\right)\right)^2 = \sum_i^n \left(res_i(\theta)\right)^2 \end{aligned} \quad (5)$$

Where $w_i = \begin{cases} \frac{1}{\lambda_L^2} & \text{for } |\Delta_i| < \lambda_L \\ \frac{1}{\Delta_i^2} & \text{otherwise} \end{cases}$.

The equations above are written based on the static environment. In a dynamic world, object movement is expressed as a displacement function Δ_d according to

$$z_{dk} = \gamma\left(\mathbf{x}_{dk} + \Delta_d\right) + \eta_{dk} \quad (6)$$

Where $\Delta_d = \frac{|D_2 F D_1|}{\sqrt{\|v_1\|^2 + \|v_2\|^2}}$, F is the basic matrix, D_1 and D_2 are dynamic points, v_1 and v_2 are epipolar lines, respectively. (6) represents the measurement model. The displacement function generates additive noise and convergence compromise if (5) is applied to the dynamic landmark. Some resilient methods, like Random sample consensus (RANSAC), can only be used in low-dynamic environments to reduce movement's influence. These approaches fail in situations with poor texture and indoor lighting conditions. Therefore, the output of static-based estimating equations will include drift factors, resulting in an estimation failure. Moving feature points should be filtered to circumvent this failure issue. Using sensors, displacement may be monitored in real-time applications. Therefore, there is a trade-off between handling dynamic objects and real-time speed. In our suggested strategy, the weights in (5) are employed to eliminate feature points in motion.

Algorithm 1 Dynamic point determination

```
Input
    D   Detected Feature points
    Z   Observation vector
Output
    z_dk without dynamic feature points
Procedure:
Enhancing frames:
for d = 1 → I do
    for k = 1 → J do
        calculate z_dk = γ(x_dk + Δ_d) + η_dk
        calculate Δ_d = |D2' F D1| / sqrt(||v1||^2 + ||v2||^2)
        if Δ_d < Threshold then
            D is static.
        else if Δ_d > Threshold then
            eliminate D in z_dk.
        end if
    end for
end for
```

In the preprocessing stage, the histogram of RGB frames with inadequate characteristics found on our landmarks has been equalized. This distributes the image's most frequent pixel intensity values,

allowing locations with low local contrast to get a boost in contrast. Using BRISQUE, the input RGB frames' quality is then computed for processing in the thresholding block. As a result of outlier filtering, this phase minimizes the process's complexity.

Front-end operations include motion estimates between successive frames and keyframe insertion. While the system is being initialized, the following stages are executed. For each new frame, features are matched using the previous frame's detected features and described as a binary string using modified Binary Robust Independent Elementary Features (BRIEF) [22]. As shown in Figure 2, the average of neighboring pixels has been computed following the formula (2).

$$\begin{aligned} \bar{P}_1 &= \frac{p_1 + p_2 + p_3}{3} \\ \bar{P}_2 &= \frac{p_4 + p_6}{2} \\ \bar{P}_3 &= \frac{p_7 + p_8 + p_9}{3} \end{aligned} \tag{7}$$

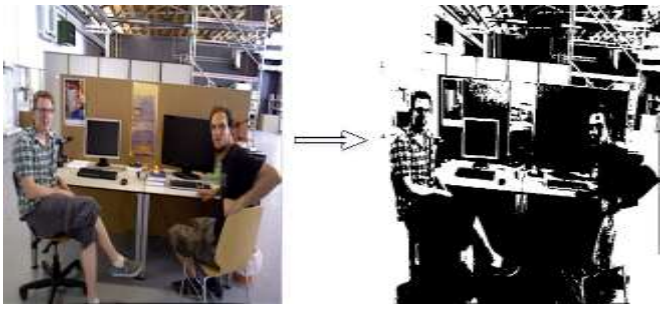

Fig. 2. The pixel value is substituted with the step function

We can convert the frame patch to the binary string by applying (5), as shown in Fig. 3:

Fig. 3. Average value calculated of pixel intensities on each row and substituting with pixel values

These augmented frames serve as inputs to the feature detector and matcher. After extracting GFTT and BRIEF features, the motion estimation process finds correspondences to model sensor data. Based on equation (4), the distance function is computed and compared to the previously measured value with moving identified landmarks. The feature will be deleted from the following step's interest points if the difference is not met. The Perspective-n-Point method determines the camera's posture based on modeled correspondences from the feature matching step between 3D reference points and their 2D projections. To develop a unique solution to the PnP issue, we use an iterative 5-point solution. The key benefits of this algorithm are its speed and the large percentage of correctly detected RGB frame characteristics. Using Bag-of-Words [23], we then attempt to locate a loop closure. If a new back-end closed-loop is discovered, the graph will be optimized using the Levenberg-Marquardt algorithm. The back-end attempts to minimize global cumulative drift and convergence, resulting in inadequate matches and diminished localization and mapping. This is the robustness of the technique we have presented.

3. Evaluation

This part is separated into UAV experiments on datasets and specific test scenarios. We first assess our proposed technique using TUM RGB-D indoor datasets to demonstrate it's precision and resilience. The TUM datasets include ground-truth pathways validated by a motion capture device in GPS-denied and indoor environments. These regions may be classified as static, low-dynamic, or dynamic. A tiny portion of the collected picture contains dynamic feature points in an environment with poor dynamic range. In a highly dynamic environment, most composed scenes include dynamic feature points. We did all operations on a computer with a 3.40GHz Intel Core i7 processor and 8 GB of RAM. Our suggested strategy is compared to those presented in [24–31].

3.1. RGB-D Datasets

Due to odometry loss and pose estimation problems, many difficult datasets were chosen to validate the robustness and accuracy of our proposed technique in indoor and GPS-denied scenarios [32]. Table 1 and Figure 4 represent the quantitative and qualitative outcomes, respectively. Our suggested approach increased accuracy and resilience to high angular velocities and large-scale trajectories by counting inliers and outliers. The red error between the black ground truth and the blue predicted trajectory is seen in Figure 5. As shown by the experimental findings, our suggested technique enhanced the precision and robustness of all chosen trajectories. It should be emphasized that the datasets we used comprised varying textures and brightness conditions. The suggested approach demonstrates a precise and reliable estimate. To compare the performance of our proposed method with state-of-the-art algorithms, we present table 3 which shows the simulation results.

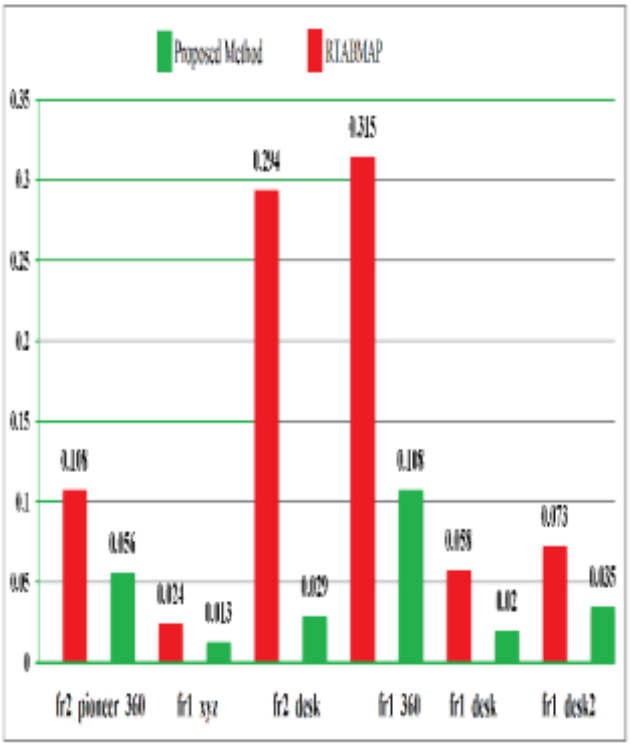

Fig. 4. The Proposed method performance improvement. The vertical axis indicates ATE value, and the horizontal axis indicates the dataset name.

Table 1
Calculated ATE [m] and compared the RTABMAP algorithm with our proposed method.

Dataset	**HS ng sitti FR3 ST**	**ng sitti FR3 XY**	**ng sitti HS king wal FR3 ST**	**king wal FR3 XY**	**king wal**
Pose pairs	180	36	171	1018	714	826
RMSE	0.0354	0.0125	0.0113	0.0255	0.0127	0.0154
mean	0.0293	0.0112	0.0101	0.0223	0.0110	0.0133
median	0.0230	0.0108	0.0089	0.0200	0.0102	0.0114
STD	0.0198	0.0055	0.0051	0.0123	0.0062	0.0076
min	0.0055	0.0027	0.0013	0.0015	0.0009	0.0007
max	0.1260	0.0290	0.0316	0.0683	0.0366	0.0585

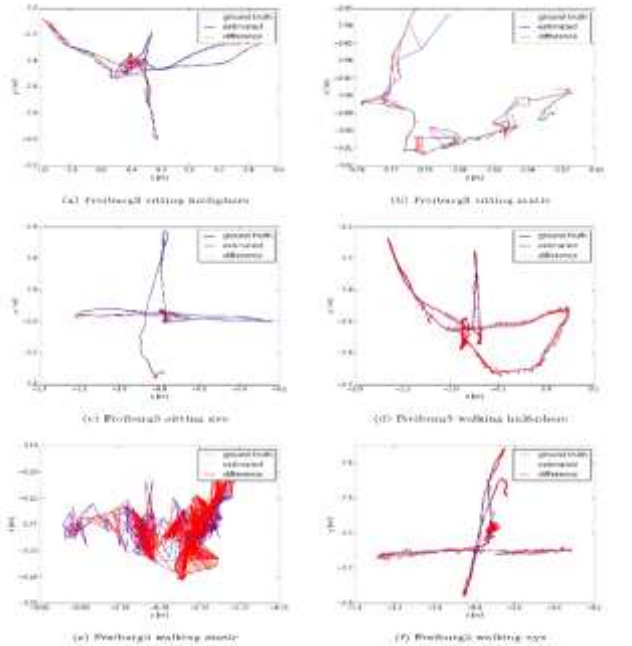

Fig. 5. The proposed method is applied to the TUM RGBD dynamic datasets.

3.2. Experiment Scenarios

The design of drones needs to be small and lightweight, with maximum load-carrying capacity and optimum power consumption. Consequently, battery and processing power are pretty limited for a flying robot. The UAV needs a small-sized processor with sufficient processing power to implement high-level control algorithms, such as SLAM. The UAV used in this study was designed

and assembled, as shown in Fig. 6. We have conducted three different experiments to evaluate our proposed method in our lab.

Fig. 6. Our designed UAV: (up) 3D CAD, (down) Assembled UAV equipped with RGBD sensor.

The first experiment is composed of a hallway and a circular furnished area with a diameter of 12 m, as shown in Fig. 7, including tall and texture-less walls, long corridors, and moving people. The localization information from the Gmapping [25] algorithm and 2D laser scanner is employed as the ground truth for our reference.

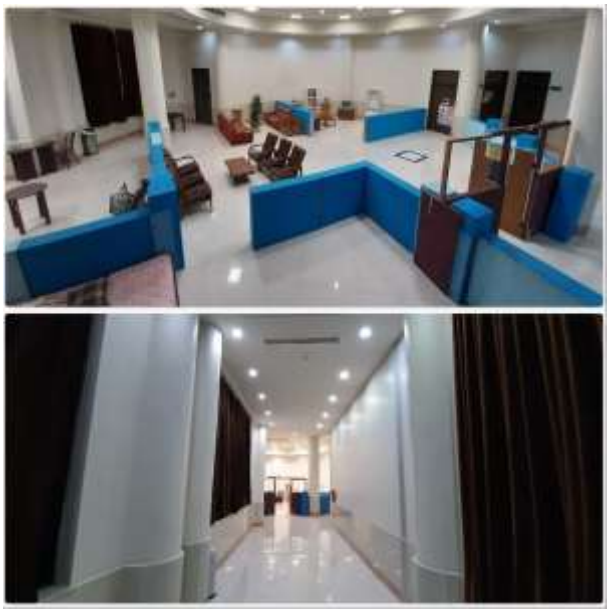

Fig. 7. The first experimental environment includes textureless walls and indoor lamps.

In addition, we used Octomap [26] to view our interior surroundings, as seen in Figure 8. The discrepancy between the projected and actual trajectory is modest and consistent, indicating that our suggested approach is accurate and vibration-resistant as the experimental results in Table 2 is shown. It should be noted that the second and third scenario experimental results is presentes to Table 2, as shown below.

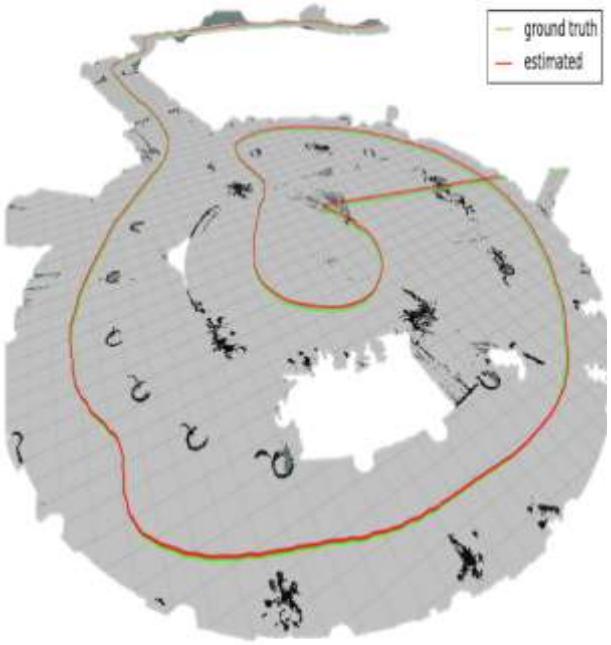

Fig. 8. Path generation of our proposed method alongside ground truth.

Table 2
ATE [m] criteria calculate the experimental result detail of our first scenario.

Test environment	First scenario	Second scenario	Third scenario
Pose pairs	279	411	411
ATE RMSE	0.156	0.191	0.191
ATE mean	0.119	0.153	0.153
ATE median	0.108	0.134	0.134
ATE STD	0.089	0.099	0.099
ATE min	0.015	0.011	0.011
ATE max	0.213	0.261	0.261

As seen in Fig. 9, The second experiment is composed of a corridor and a 50-meter-diameter circular furnished space. The brightness and texture vary from the first test environment. In the previous test, Gmapping and the 2D laser scanner served as the ground truth for the data. In Figure 10, the outcomes of the suggested technique are shown beside the ground truth. The discrepancy between the projected and actual trajectory is modest and consistent, indicating that our suggested approach is accurate and vibration-resistant. The experimental results is shown in Table 2. The Octomap was used to illustrate our interior environment map. Our scenario includes sunshine in our laboratory to demonstrate the camera's resistance to brightness circumstances, as the RGB-D camera emits infrared light and sunlight interferes. In addition, the structure included lofty columns, walls, and windows that might compromise feature detection.

Fig. 9. MRL main hall of indoor and GPS-denied arenal.

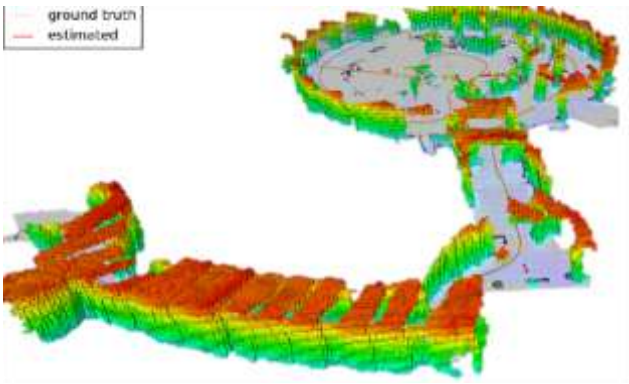

Fig. 10. Visualizes evaluation environment using Octomap.

The MRL main hall that the third test was conducted on was composed of a hallway and a circular furnished area with diameter of 50*m* that is shown in Fig 11. The luminance and texture are different from the first test environment. According to previous test condition, the localization information from Gmapping algorithm and 2D laser scanner are used as the ground-truth for our reference. The results of the proposed method alongside with the ground-truth are illustrated in Fig. 12. The error between estimated trajectory and ground- truth is uniform and small which means that our proposed method is accurate and robust to vibrations and noises. Our scenario included sun light in the MRL main hall to show robustness to luminance condition because of IR radiated from RGBD camera and sun light interference. In addition, it has tall columns, walls and windows which could corrupt feature detection. Test statistics are shown in Table 2.

Fig. 11. The second experimental environment includes textureless walls and indoor lamps.

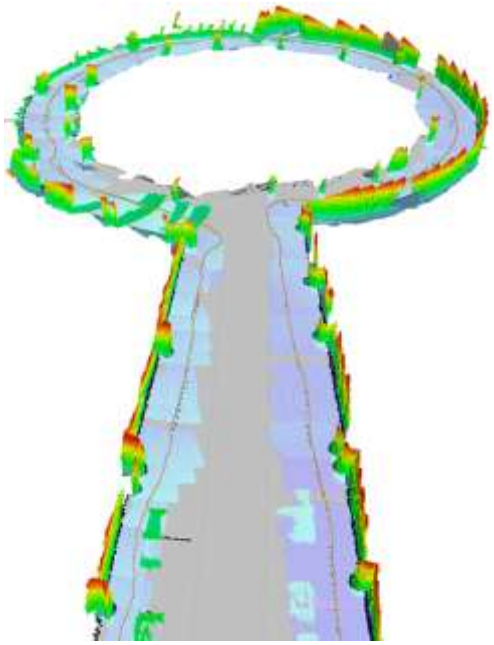

Fig. 12. Path generation of our proposed method alongside ground truth.

Table 3
ATE RMSE [m] criteria comparison between state-of-the-art methods.

Dataset	**Fr3 sitting HS**	**Fr3 sitting ST**	**Fr3 walking HS**	**Fr3 walking ST**	**Fr3 walking XYZ**
DVO [34]	0.1005	0.0157	0.2628	0.3818	0.4360
BaMVO [34]	0.0589	0.0248	0.1738	0.1339	0.2326
Depth edge+IAICP [34]	0.0624	0.0198	0.2016	0.1192	0.1802
Depth edge+RANSAC +IAICP [34]	0.0583	0.0210	0.0799	0.0496	0.1482
Static point Weighting [34]	0.0389	0.0231	0.0527	0.0327	0.0651
Our proposed method	**0.0354**	**0.0125**	**0.0255**	**0.0127**	**0.0154**

4. Conclusion and Future Works

In this paper, we present a vision-RGB-D SLAM based on the Levenberg-Marquardt algorithm as the backbone of our suggested strategy to minimize outliers and boost inliers in real-world applications. Combining depth information with vision data yielded precise and resilient graph optimization. The quality of these characteristics affects RGB-D SLAM's overall performance. The suggested technique enhanced these criteria and yield better results. The proposed approach was tested using public RGB-D datasets and our developed UAV in the lab's indoor and GPS-devoid main hall. The results revealed that our proposed strategy might be used in a demanding and texture-less application. They beat the majority of state-of-the-art RGB-D SLAM algorithms under circumstances of very high angular velocity and massive sequence datasets. It might potentially be used in large indoor environments for real-time applications. Implementation in a high dynamic environment with rotational movements and different noise models is left for future work. In addition, the distinctive sparsity of the Hessian structure underlying the point correlation formulation may be used to create more effective solutions. Finally, random sample procedures will be examined to improve the robustness of the estimate in future work.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," IEEE robotics & automation magazine, vol. 13, no. 2, pp. 99–110, 2006.

[2] Durrant-Whyte, H. and Bailey, T., 2006. Simultaneous localization and mapping: part I. IEEE Robotics & Automation Magazine, 13(2), pp.99-110.

[3] M. Norouzi, J. V. Miro, G. Dissanayake, and T. Vidal-Calleja, "Path planning with stability uncertainty for articulated mobile vehicles in challenging environments," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014, pp. 1748–1753.

[4] D. Sharafutdinov, M. Griguletskii, P. Kopanev, M. Kurenkov, G. Ferrer, A. Burkov, A. Gonnochenko, and D. Tsetserukou, "Comparison of modern open-source visual slam approaches," arXiv preprint arXiv:2108.01654, 2021.

[5] C.-Z. Sun, B. Zhang, J.-K. Wang, and C.-S. Zhang, "A review of visual slam based on unmanned systems," in 2021 2nd International Conference on Artificial Intelligence and Education (ICAIE). IEEE, 2021, pp. 226– 234.

[6] X. Gao and T. Zhang, Introduction to Visual SLAM: From Theory to Practice. Springer Nature, 2021.

[7] H. Kang, K. Ku, and J. Shim, "A rgb-d vision based indoor slam using 2.5 d map by multiple uavs," in

2021 21st International Conference on Control, Automation and Systems (ICCAS). IEEE, 2021, pp. 1624–1627.

[8] R. Azzam, T. Taha, S. Huang, and Y. Zweiri, "Feature-based visual simultaneous localization and mapping: A survey," SN Applied Sciences, vol. 2, no. 2, pp. 1–24, 2020.

[9] T. D. Barfoot, State estimation for robotics. Cambridge University Press, 2017.

[10] A. Harmat, I. Sharf, and M. Trentini, "Parallel tracking and mapping with multiple cameras on an unmanned aerial vehicle," in International Conference on Intelligent Robotics and Applications. Springer, 2012, pp. 421–432.

[11] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," IEEE transactions on robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[12] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "Pl-slam: A stereo slam system through the com- bination of points and line segments," IEEE Transactions on Robotics, vol. 35, no. 3, pp. 734–746, 2019.

[13] C. Wei, Y. Tang, L. Yang, and Z. Huang, "Slc-vio: a stereo visual- inertial odometry based on structural lines and points belonging to lines," Robotica, pp. 1–21, 2022.

[14] C. Qiao, T. Bai, Z. Xiang, Q. Qian, and Y. Bi, "Superline: A robust line segment feature for visual slam," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 5664– 5670.

[15] H. Lim, J. Jeon, and H. Myung, "Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping," IEEE Robotics and Automation Letters, 2022.

[16] G. Yang, Q. Wang, P. Liu, and C. Yan, "Pls-vins: Visual inertial state estimator with point-line features fusion and structural constraints," IEEE Sensors Journal, vol. 21, no. 24, pp. 27 967–27 981, 2021.

[17] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, and J.-P. Thiran, "Scale invariant feature transform on the sphere: Theory and applications," International journal of computer vision, vol. 98, no. 2, pp. 217–241, 2012.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," Computer vision and image understanding, vol. 110, no. 3, pp. 346–359, 2008.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in 2011 International conference on computer vision. Ieee, 2011, pp. 2564–2571.

[20] F. Zhang, S. Li, S. Yuan, E. Sun, and L. Zhao, "Algorithms analysis of mobile robot slam based on kalman and particle filter," in 2017 9th International Conference on Modelling, Identification and Control (ICMIC). IEEE, 2017, pp. 1050–1055.

[21] Q. Yu, J. Liang, J. Xiao, H. Lu, and Z. Zheng, "A novel perspective invariant feature transform for rgb-d images," Computer Vision and Image Understanding, vol. 167, pp. 109–120, 2018.

[22] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2018, pp. 10–20.

[23] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Rs-slam: Ransac sampling for visual fastslam," in 2011 IEEE/RSJ International Conference on Intel- ligent Robots and Systems. IEEE, 2011, pp. 1655–1660.

[24] F.-A. Georgescu, G. Cas, M. Datcu, and D. Raducanu, "Modified binary robust independent elementary features for fast earth observation data analysis," MTA Review, vol. 25, no. 2, pp. 189–194, 2015.

[25] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," International Journal of Machine Learning and Cybernetics, vol. 1, no. 1-4, pp. 43–52, 2010.

[26] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-time cam- era tracking and 3d reconstruction using signed distance functions," in Robotics: Science and systems (RSS) conference 2013.

[27] H. Dong, N. Figueroa, and A. El Saddik, "Towards consistent reconstruc- tions of indoor spaces based on 6d rgb-d odometry and kinectfusion," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014, pp. 1796–1803.

[28] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," IEEE transactions on robotics, vol. 30, no. 1, pp. 177–187, 2013.

[29] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in 2013 IEEE/RSJ International Conference on Intelligent Robots and

Systems. IEEE, 2013, pp. 2100–2106.

[30] M. Meilland and A. I. Comport, "Super-resolution 3d tracking and map- ping," in 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013, pp. 5717–5723.

[31] J. Stückler and S. Behnke, "Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras," in 2012 IEEE Inter- national Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). IEEE, 2012, pp. 162–167.

[32] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense rgb-d mapping," in 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013, pp. 5724–5731.

[33] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense rgb-d slam with volumetric fusion," The International Journal of Robotics Research, vol. 34, no. 4- 5, pp. 598–626, 2015.

[34] Li, S., & Lee, D. (2017). RGB-D SLAM in dynamic environments using static point weighting. IEEE Robotics and Automation Letters, 2(4), 2263-2270.