Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Harsh-Environment Visual Odometry for Field Robots Using Data Fusion of Gyroscope & Magnetometer

Chul-hong Kim*, Jee-seong Kim*, and Dong-il "Dan" Cho*, **

**Department of Electrical and Computer Engineering and Automation and Systems Research Institute (ASRI), Seoul National University, Seoul, Korea (chkim89@snu.ac.kr, jsknov@snu.ac.kr)*
***Inter-university Semiconductor Research Center (ISRC), Seoul National University, Seoul Korea (dicho@snu.ac.kr)*

Abstract: This paper presents a harsh-environment visual odometry method that is robust to the robot orientation as well as the camera movement in an outdoor environment. The accuracy of visual odometry in robots can be enhanced by using additional sensor measurements such as an encoder, gyroscope, and/or magnetometer. This strategy can even reduce the computational time. However, in an outdoor environment, the moving robot can experience vibration, which causes unsynchronized data fusion between the camera and additional sensors. This unsynchronized data fusion causes errors in the robot orientation, which can lead to unwanted large drift errors in localization. To overcome this problem, firstly two distinctively different characteristics of the gyroscope and the magnetometer are combined to estimate the robot orientation. The initial robot orientation is estimated by integration of the gyroscope input, and this initial robot orientation is corrected using the magnetometer data in bundle adjustment. Secondly, the poses of the robot and the camera are estimated separately, and these separately estimated poses of the robot and the camera are used in feature matching and bundle adjustment to reduce drift errors in localization in the outdoor environment. The performance of the proposed method is demonstrated using dataset-based experiments.

Keywords: Visual odometry; Mobile robot; Field robot; Outdoor environment.

1. INTRODUCTION

Recently, mobile robot technologies such as autonomous vehicle, indoor robot and field robot have been actively researched. Simultaneous localization and mapping (SLAM) is an important prerequisite for autonomous navigation of these robots. In SLAM, vision sensors are frequently used among various sensors because they provide abundant information (Lee *et al*., 2015, 2016). Visual SLAM is usually composed of two parts, visual odometry and loop closing.

In visual SLAM, researchers mainly estimated the location of the robot by estimating the camera movement, translation and orientation. The camera movement is mainly estimated by using feature points as a landmark. The parallel tracking and mapping (PTAM) which is proposed by Klein *et al*. (2007) simultaneously tracks the camera pose and maps an environment by using two threads. Lee *et al*. (2018) proposed line feature-based monocular SLAM with three threads which are tracking, bundle adjustment and loop closing. By using supplementary sensors such as inertial sensor and encoder, this method reduces the computational complexities of SLAM, which applies to a low-cost embedded system. The current location of the robot is initially estimated by using the inertial sensor and the encoder to the previously estimated location of the robot. So, errors in the robot orientation can cause large drift errors in localization.

The robot orientation can be estimated by various sensors including the gyroscope and the magnetometer. However, the gyroscope and the magnetometer have distinctively different characteristics. The gyroscope-based orientation estimation method, which is an integration of angular velocity in the time domain, suffers from integration errors with time. The magnetometer-based orientation estimation method, which calculates the robot orientation with respect to the north, suffers from environment-dependent errors. These two distinctively different characteristics can be combined to reduce errors in the robot orientation. (Kim *et al*., 2019)

From the perspective of visual odometry, conditions such as a low textured area (Park *et al*., 2017), slippery floor, a low-luminance condition (Yi *et al*., 2018) and an outdoor environment, can be considered as a harsh environment. Among these conditions, the outdoor environment causes unsynchronized data fusion between vision and sensors. In an outdoor environment, the robot will experience vibration while moving, which causes the camera orientation to change at a high rate. To use sensor measurements, the robot orientation needs to point at the next frame. As the camera orientation indicates the camera direction at the moment, the robot orientation is not equal to the camera orientation in the outdoor environment. So when the robot pose is equal to the camera pose, using sensor measurements can cause large drift errors in localization. Hence the pose of the robot and the

camera at the same frame needs to be separately estimated in orientation to use sensor measurements.

The main contribution of this paper is harsh-environment visual odometry that is robust to the robot orientation as well as the camera movement in an outdoor environment. In visual odometry, the image process takes the biggest portion of computational time, and the easiest way to reduce the computational time is decreasing the number of input images. To prevent the degradation of visual odometry performance due to reduced input images, the supplementary sensors are used. These supplementary sensors are a robot wheel encoder, gyroscope and magnetometer. The robot wheel encoders and the gyroscope are used to estimate the initial pose of the robot, and the magnetometer is used to reduce the orientation error of the robot. The pose of the robot is estimated in the 2D plane with 3 degrees of freedom (DoF) (x, y, pan), while the camera pose is estimated with 5 DoF (x, y, pan, roll, tilt). These two separately estimated poses of robot and camera are used in feature matching and bundle adjustment to reduce the drift error in the outdoor environment. Consequently, the performance of the visual odometry can be improved.

The rest of the paper is organized as follows. In Section 2, we present the visual odometry algorithm. In Section 3, we present our experimental results. Finally, Section 4 presents our conclusion.

2. VISUAL ODOMETRY

2.1 Overall Structure

In this work, we use a camera as a primary sensor, and a gyroscope, an encoder and a magnetometer as supplementary sensors. The data of the gyroscope and the encoder are integrated to estimate the initial poses of the robot which are stated as raw odometry. The magnetometer is used to measure the robot orientation at each instance. Images from the camera are used to estimate the robot pose and landmark position. The overall structure of visual odometry is shown in Fig. 1. Inspired by the algorithm of Lee *et al.* (2018), visual odometry comprises two threads which are tracking and optimization. The tracking thread extracts feature and uses random sample consensus (RANSAC) to remove the outlier. These feature matchings are used to estimate landmark position by triangulation and to calculate essential matrix which is used to estimate the tilt and roll of the camera. The optimization thread optimizes the poses of an active window of selected frames and the position of corresponding landmarks by using local bundle adjustment.

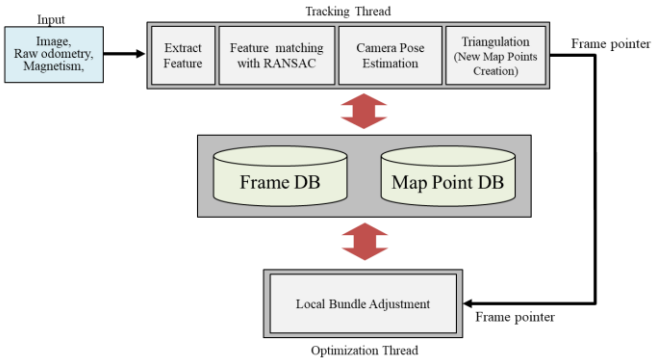

Fig. 1. Overall structure of the proposed visual odometry.

2.2 Camera Pose Estimation in Tracking Thread

To separate the poses of robot and camera, the camera poses need to be estimated as the initial robot poses are estimated by supplementary sensors. The pose of the robot is estimated in the 2D plane with 3 degrees of freedom (DoF) (x, y, pan), while the camera pose is estimated with 5 DoF (x, y, pan, roll, tilt). As the poses of the robot and the camera share x, y and pan, roll and tilt of camera poses should be estimated additionally.

The essential matrix between frames is used to estimate these additional parameters, roll and tilt. Firstly, the rotation matrix is calculated from an essential matrix as follows:

$$\mathbf{R} = \det\left(\mathbf{U}\mathbf{W}\mathbf{V}^T\right)\mathbf{U}\mathbf{W}\mathbf{V}^T \text{ or } \det\left(\mathbf{U}\mathbf{W}^T\mathbf{V}^T\right)\mathbf{U}\mathbf{W}^T\mathbf{V}^T, \tag{1}$$

where **U** and **V** are left-singular vectors and right-singular vectors of the essential matrix, respectively. **W** is defined as follows:

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

By using the Euler angle, we can find the tilt and roll of the frame with respect to the other frame.

$$\mathbf{R}_{\theta_1\theta_2\theta_3} = \begin{bmatrix} c_1c_2 & c_2s_1 & -s_2 \\ c_1s_2s_3 - c_3s_1 & c_1c_3 + s_1s_2s_3 & c_2s_3 \\ s_1s_3 + c_1c_3s_2 & c_3s_1s_2 - c_1s_3 & c_2c_3 \end{bmatrix}, \tag{3}$$

where θ_1, θ_2 and θ_3 are pan, tilt and roll, respectively, c_i and s_i are cosine and sine of i^{th} angle. For every matched frame, tilt and roll of the frame are estimated and RANSAC is used to finally estimate the tilt and roll of the frame with respect to the world frame.

2.3 Tracking Thread

Tracking thread deals with the pose of the current frame and position of the corresponding landmarks before registering them on the map. ORB feature (Rublee *et al.*, 2011) which is extracted from a current frame is used in feature matching, and 8 points RANSAC is used to remove outliers. With feature matching results, we generate an essential matrix between the current frame and the matched frame. This essential matrix is used to calculate the tilt and roll of the current frame through camera pose estimation.

After estimating tilt and roll of the current frame, the feature matching result is used to estimate the landmark position with triangulation. Before the triangulation process, landmarks are classified into two cases, either the new

landmark or the registered landmark on the map. If landmarks are already registered in the map, the observation of the current frame will be added to the landmark of the map. If the landmark is new, the triangulation of the observations will be used to calculate the initial position of the landmark, then register it to the map.

2.4 Robot Orientation Estimation using Gyroscope and Magnetometer in Optimization Thread

Gyroscope-based orientation estimation and magnetometer-based orientation estimation have distinctively different characteristics. The gyroscope-based orientation estimation is an integration of angular velocity in the time domain, so it has an integration error with time. The magnetometer-based orientation estimation calculates the robot orientation with respect to the north, so it has environment-dependent errors. The gyroscope-based orientation estimation is more accurate than magnetometer-based orientation estimation in a short period. However, the magnetometer-based orientation estimation has better performance than gyroscope-based orientation estimation in a long period.

The proposed method uses graph optimization to combine these two distinctively different characteristics to reduce the robot orientation estimation. Gyroscope is used to estimate initial robot orientation because the accuracy of gyroscope-based orientation estimation is high in a short period. In the graph optimization process, raw magnetometer data is used to correct the initial orientation estimation errors which are accumulated with time in a long period. The following energy function is used in the optimization thread:

$$\begin{aligned} &E(\theta_s, \theta_{s+1}, \cdots) = \\ &\sum_i R_g \cdot \text{normAngle}(z_{\text{g},i} - (\theta_i - \theta_{i-1}))^2 + \\ &\sum_j R_m \cdot \text{normAngle}(z_{m,\text{j}} - \theta_j)^2, \end{aligned} \tag{4}$$

where θ_i is the i$^{\text{th}}$ frame robot orientation, $z_{\text{g},i}$ is the i$^{\text{th}}$ gyroscope-based initial robot orientation estimation result, $z_{\text{m},i}$ is the i$^{\text{th}}$ raw magnetometer data. R_{g}, R_m is the information values of the gyroscope and the magnetometer, respectively. In bundle adjustment, robot orientation will be expressed inside 2D robot poses. The Levenberg-Marquardt algorithm is used to minimize the energy function (4). A g2o framework (Kummerle *et al*., 2011) is used for the Levenberg-Marquardt algorithm.

2.5 Optimization Thread

Optimization thread uses local bundle adjustment to optimize the poses of selected frames and the positions of corresponding landmarks. Consecutive frames before the current frame are selected for optimization, and the number of consecutive frames is determined depending on the measurements of these frames. The measurement of the magnetometer is used to estimate the orientation of each frame as it shows the orientation of the robot with respect to the earth's magnetic field. The following energy function is used in optimization:

$$\begin{aligned} &E(\mathbf{x}_s^w, \mathbf{x}_{s+1}^w, \cdots, \mathbf{p}_t^w, \mathbf{p}_{t+1}^w, \cdots) = \\ &\sum_i \| (\mathbf{z}_{o,i} - (\mathbf{x}_i \ominus \mathbf{x}_{i-1})) \|_{\Sigma_o}^2 \\ &+ \sum_i R_i (z_{\text{mag},s,i} - (\theta_i - \theta_s))^2 \\ &+ \sum_i \sum_j \| (\mathbf{z}_{uv,i,j} - \mathbf{K}(\mathbf{R}_i \mathbf{p}_j^w + \mathbf{t}_i)) \|_{\Sigma_{w_i}}^2, \end{aligned} \tag{5}$$

where $\mathbf{x}_i$ is the i$^{\text{th}}$ frame pose, $\mathbf{p}_j$ is the jth landmark position, $\mathbf{z}_{o,i}$ is the i$^{\text{th}}$ raw odometry data, $\mathbf{z}_{uv,i,j}$ is the observation of the i$^{\text{th}}$ frame of the j$^{\text{th}}$ landmark from the cameras, $\mathbf{K}$ is intrinsic parameters of the camera, $\mathbf{R}_i$ is the rotation matrix of the i$^{\text{th}}$ frame with fixed tilt and roll, $\mathbf{t}_i$ is the translation matrix of the i$^{\text{th}}$ frame, $z_{\text{mag},s,i}$ is the measurement of magnetometer at i$^{\text{th}}$ frame, θ_i is the i$^{\text{th}}$ frame orientation, Σ_o and Σ_{w_i} are the covariance matrix of raw odometry and features respectively, and $\ominus$ is a relative transformation between two sequential frame poses $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$, defined as follows:

$$\mathbf{x}_i \ominus \mathbf{x}_j = \begin{bmatrix} (x_i - x_j)\cos\theta_j + (y_i - y_j)\sin\theta_j \\ -(x_i - x_j)\sin\theta_j + (y_i - y_j)\cos\theta_j \\ \theta_i - \theta_j \end{bmatrix}. \tag{6}$$

The Levenberg-Marquardt algorithm is used to minimize the energy function (4). A g2o framework (Kummerle *et al*., 2011) is used for the Levenberg-Marquardt algorithm.

3. EXPERIMENTS

3.1 Experimental Setup

For evaluation, datasets-based experiments are executed on a desktop personal computer. For the robot platform, we use the Pioneer 3-DX (Espinosa *et al*., 2010) shown in Fig. 2. The robot has a built-in encoder. MPU-9265 is used for the gyroscope and the magnetometer and placed on the top of the robot. The camera is placed in front of the robot. Figure 3 is the consecutive images of the camera while the robot moves forward. As shown in the images, tilting and rolling of the camera change through the images.

The experimental environment is illustrated in Fig. 4. As the robot moves on the grass, the camera vibrates frequently. The camera took the images at 1 Hz, and the robot moved at 0.4 m/s. The robot moved in 76.4 m, 76.1 m, 344.2 m, 323.0 m, and 357.8 m trajectories and captured 312, 300, 1284, 1185, and 1210 images respectively, for five outdoor datasets. At the end of the datasets, we control the robot to place back to the beginning position. The distance between the beginning and end position of the trajectory is used to evaluate visual odometry performance. This distance is referred as the closed-loop error for the rest of the paper.

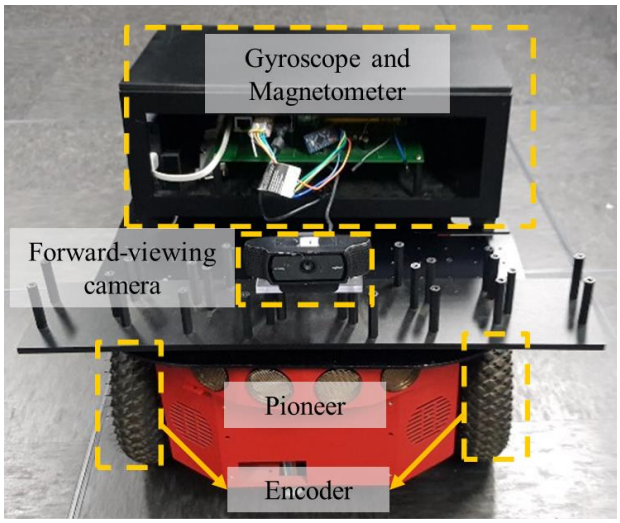

Fig. 2. Pioneer 3-DX used for experiments.

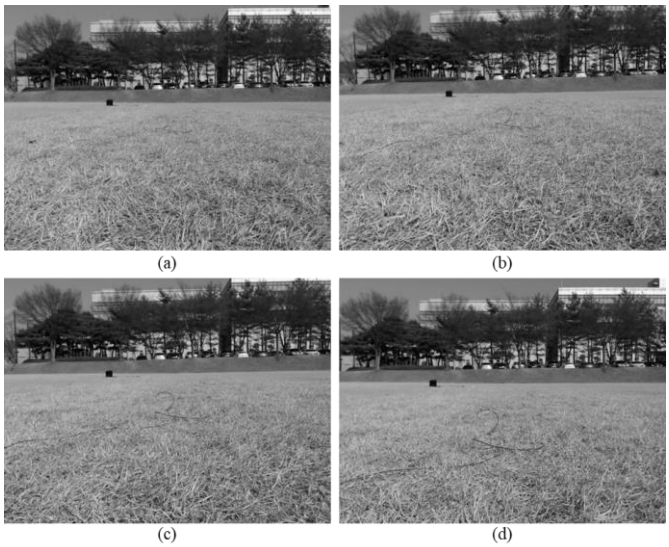

Fig. 3. Consecutive images of camera while robot moving forward.

3.2 Experiment Result

The closed-loop error is measured for all experimental datasets. In Fig. 5, the robot trajectory of the raw odometry data and the proposed method for dataset 1 and 5 are shown. The robot moved in a square pattern for dataset 1 and moved in a zigzag pattern for dataset 5. The closed-loop errors for the datasets-based experiment are shown in Table 1. The closed-loop error of the raw odometry is larger than that of the proposed method. The average closed-loop errors for the raw odometry and the proposed method are 7.89 m and 0.93 m respectively. These results show that the closed-loop errors of datasets with long moving distance are greater than that of datasets with short moving distance. Therefore, this shows that the visual odometry builds localization error as the robot moves.

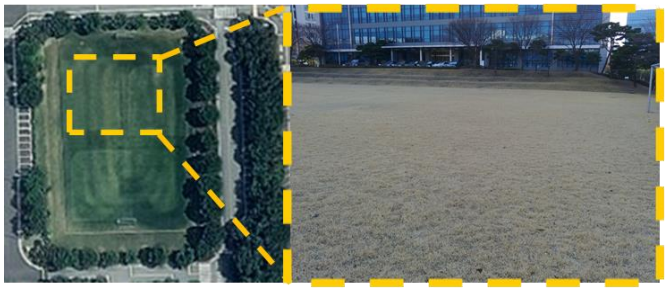

Fig. 4. Outdoor environment for experimental datasets.

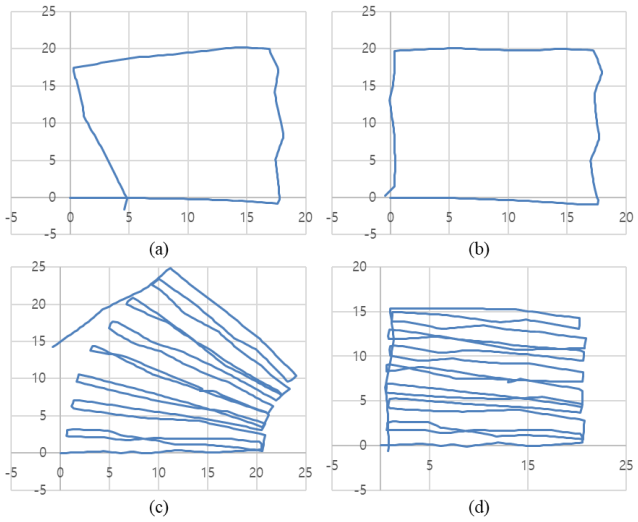

Fig. 5. Trajectory of raw odometry and the proposed method for dataset 1 and 5. (a, c) the trajectories of the raw odometry for dataset 1 and 5, respectively. (b, d) the trajectories of the proposed method for dataset 1 and 5, respectively.

Table 1. Closed-loop errors of raw odometry and the proposed method in 5 datasets.

Dataset	Total moving distance (m)	Closed-loop errors of raw odometry (m)	Closed-loop errors of the proposed method (m)
1	76.4	4.84	0.51
2	76.1	1.28	0.59
3	344.2	14.3	1.16
4	323.0	5.93	1.19
5	357.8	13.1	1.20
Avg.	235.5	7.89	0.93
Std.		5.59	0.35

4. CONCLUSIONS

In this paper, we proposed harsh-environment visual odometry which is robust to camera movement in an outdoor

environment by using different poses of the robot and the camera. The robot wheel encoders and the gyroscope were used to generate raw odometry, and the magnetometer was used to reduce the orientation error of the robot. The pose of the robot was estimated in the 2D plane with 3 DoF (x, y, pan), while the camera pose was estimated with 5 DoF (x, y, pan, roll, tilt). These separately estimated poses of robot and camera were used in feature matching and bundle adjustment to reduce the drift error in localization. Dataset based experiments were performed, and the results showed that the closed-loop error of the proposed method is 88% lower than that of the raw odometry.

ACKNOWLEDGEMENT

This work was supported by LG Electronics Inc., Seoul, Korea.

REFERENCES

Coughlan, J. M. and Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by Bayesian inference. *In Proc. IEEE International Conference on Computer Vision*, 943–947. Kerkyra, Greece.

Espinosa, F., Salazar, M., Pizarro, D., and Valdes, F. (2010). Electronics proposal for telerobotics operation of P3-DX units, *Remote and Telerobotics*, IntechOpen.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g2o: A General Framework for Graph Optimization. *In Proc. IEEE International Conference on Robotics and Automation*, 3607–3613. Shanghai, China.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. *In Proc. IEEE ACM Int. Symp. Mixed Augmented Reality*. 1-10. Nara, Japan.

Kim, C., Kim, J., and Cho, D. (2019). Graph Optimization-based Orientation Estimation of Mobile Robot Using Gryroscope and Magnetometer. *In Proc. The 34th ICROS Conference*, TP-32, Gyeongju, Korea.

Lee, T., Jang, B., and Cho, D. (2015). A Noverl Method for Estimating the Heading Angle for a Home Service Robot Using a Forward-Viewing Mono-Camera and Motion Sensors. *International Journal of Control, Automation and Systems*, 13(3). 709-717.

Lee, T., Yi, D., and Cho, D. (2016). A monocular vision sensor-based obstacle detection algorithm for autonomous robots. *Sensors*, 16(3). 1-19.

Lee, T., Kim, C. and Cho, D. (2019). A Monocular Vision Sensor-Based Efficient SLAM Method for Indoor Service Robots. *IEEE Transactions on Industrial Electronics*, 66(1). 318-328.

Park, J., Lee, T., Kim, C., and Cho, D. (2017). Convolutional Neural Network-Based Pose-Graph SLAM System Using a Monocular Vision Sensors. *In Proc. The 17th International Conference on Control, Automation and Systems*, FP-C, Jeju, Korea.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *In Proc. IEEE International Conference on Computer Vision*, 2564–2571. Barcelona, Spain.

Yi, D., Lee, T., and Cho, D. (2018). A New Localization System for Indoor Service Robots in Low Luminance and Slippery Indoor Environment Using Afocal Optical Flow Sensor Based Sensor Fusion. *Sensors*, 18(171). 1-14.

Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., and Yu, W. (2015). StructSLAM: Visual SLAM with building structure lines. *IEEE Transactions on Vehicular Technology*, 64(4). 1364-1375.