Outdoor visual SLAM and Path Planning for Mobile-Robot

Jiwoong Choi
Kyung Hee University
jiwung22@gmail.com

Jueun Mun
Kyung Hee University
cindy4741@khu.ac.kr

Seongil Heo
Hankuk University of Foreign Studies
tjddlf101@hufs.ac.kr

Jiwon Park
Kyung Hee University
overflow21@khu.ac.kr

***Abstract*—This paper proposes robust visual SLAM and Path Planning algorithm for autonomous vehicles at the outdoor environment. The consideration of characteristics at the outdoor was essential in both SLAM and Path Planning. Visual SLAM system was developed considering the strong sunlight and utilized the GPS data which is robust at the open outdoor. GPS data compliments every three multi-thread of visual SLAM: 1) Camera Tracking; 2) Local Mapping; and 3) Loop Closing. It enhanced the accuracy of the map and prevented computational power by early stopping. Path Planning differed the priority of the path based on the stability of the road. With regard to determining the optimal path, the stability of the road as well time was considered, and the stable roads were marked by GPS sensor to assign more weight.**

***Index Terms*—Autonomous vehicles, SLAM, Path Planning, GPS, mobile robot**

I. Introduction

Autonomous vehicles (AVs) has been developed as the time goes by and shown promising performance. Modern automated vehicles have found application in numerous fields, including mobility, agriculture, and the space industry, where the capabilities of the driving situation have improved significantly. In Automated Driving Systems (ADS), there are various tasks linked and interacted with each other. In general, ADS can be divided by three stages: 1) Perception stage, which understands the location and pose of the vehicle, and map; 2) Decision stage, which generates an optimal action plan; and 3) Control stage, which produces an actual motion. The purpose of each stage is achieved by several tasks. Simultaneous Localization and Mapping (SLAM), Path Planing and Actuator [9] belong to perception, decision, and control stage respectively, and play significant roles in each of these phases. So far, several tasks have shown meaningful performance, especially at the *indoor*, which has a mild condition, or at the outdoor road where a car passes.

On the other hand, when it comes to mobile robot, several methods for the other purposes are not applicable to the *outdoor* SLAM and Path Planning due to the several reasons: 1) There is no traffic rule (lane markings); 2) There is an uncertainty whether the object is passable or obstacle; 3) environment is various and changes frequently. Considering these characteristics, this paper researched the autonomous driving of the mobile robot at the outdoor.

Visual SLAM requires choosing and matching specific points out of each camera frame for localizing and mapping. The algorithm chooses special points based on their distinguishing features, such as corners and edges. However, at the outdoor especially where the mobile robot moves, it is difficult to identify the specific points of the road due to the similarity of the environment. For instance, when a mobile robot passes a sidewalk beside the lawn, most pixels have similar features. Furthermore, non-stationary characteristic of the outdoor environment caused by nature power, human, and animals may disturb the process of creating coherent and accurate maps. Likewise, SLAM which considered these characteristics is necessary.

Meanwhile, traditional Path Planning is fundamentally based on the distance from the start point to the destination point. However, at the outdoor environment where mobile robot passes, the traditional method has some limitations since the optimal path is the one which considered distance, stability, and safety, not the one which considered only distance. For example, It is natural that the mobile robot should evade the bumpy and muddy road and drive on the concrete blocks.

There has been developed many robust methods of SLAM and Path Planning for the outdoor environment. SLAM methods can be divided by the sensor, such as LiDAR and Camera. This research selected Visual SLAM method using a camera, since it consumes low power and shows high performance, which is very important aspect for operating mobile robot. Out of Visual SLAM methods, monocular method is the primary methodm and It is convenient to install single camera at the mobile robot. However, if SLAM employs only single camera, due to the lack of various sensor data, SLAM may accumulates errors and fails tracking. So as to overcome this, GPS data equipped at the camera can do the good auxiliary role for correcting the error. Moreover, GPS data can make the process early stopped so as to prevent useless computing and react to the environment in real time quickly.

Out of Path Planning, the sampling based planners are usually applied at the outdoor environment owing to low computational complexity, in contrast to search based planners. In addition, the sampling based planners are more suitable for the large dynamic environment. Recently, there are some algorithms which added the advantage of the search based planners. For example, BIT* [20] and ABIT* [20] are the exemplars. Furthermore, this research made the existing algorithm optimal to express the nonlinear dynamics of a robot in terms of the robot's high-dimensional configuration space.

GPS data did the primary role for improving the algorithm.

Meanwhile, GPS sensor, which performs well at the outdoor, can specify the absolute position well, on the contrary to the SLAM which creates the relative position. Thus, GPS data can be good standard to lessen the stability of the monocular SLAM.

In conclusion, as Figure 6 shows this paper presents a Monocular SLAM that takes advantage of the GPS sensor to utilize additional information about the environment and a Path Planning approach that takes the samples into consideration in order to find a safe and optimal path based on the characteristics of mobile robot. In contrast to pure Monocular SLAM and path planning, these methods performed better in an outdoor environment.

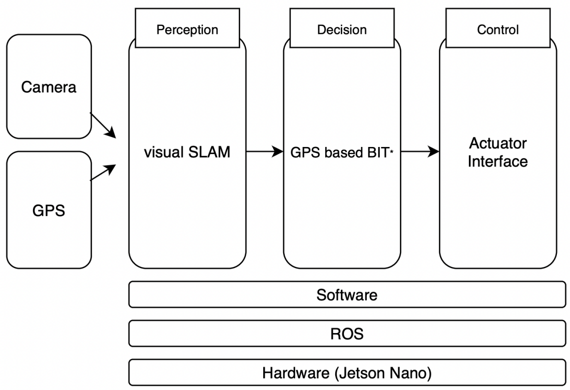

Fig. 1: System Overview

II. Related Works

A. Mobile Robot Driving

Mobile robot passes various kinds of environments, including the lawn, sidewalk, pavement, sand, and gravel road. Therefore, it is impossible to make the map absolutely in advance and the map should be made in real time reflecting the present state. In addition, there are also irregular characteristics as there are few clearly structured landmarks in open areas without lanes, signals and rules [7]. Therefore the big difference between various types of environments such as mountains and deserts should be checked.

In addition to above characteristics, there are additional problems about developing an autonomous driving system of a mobile robot at the outdoor environment: 1) The data set used in the other studies about outdoor autonomous driving of mobile robot are not formalized as there are few similarities between various types of outdoor environments. Therefore, It is necessary to collect data in person and make a model which is optimal for the near environment.

; 2) It should be able to respond to continuous change even in the same environment. When the mobile robot drives from the start point to the destination point, it faces various roads, such as a pavement, a sidewalk, a muddy or bumpy road, and the lawn. Therefore, system should be developed to react at the ever-changing environment and decide the optimal path for the current road. In order to embody this in real time, the calculation time of the algorithm should be short [21].

B. Visual SLAM

Visual SLAM is a SLAM that processes 2D images and occasionally uses depth or Inertial Measurement Unit (IMU) data. Visual SLAM utiliziang one camera is called Monocular type and the one using two cameras is called Stereo type. In addition, Visual SLAM which uses RGB-D camera is named RGB-D SLAM, and since there is no necessity to calculate the depth of the map points, it is likely to be seen as attractive, which is not true at the outdoor in fact.

The KinectFusion algorithm [16] is the first well-known RGB-D SLAM algorithm. It consists of four main steps: 1) the measurement; 2) pose estimation; 3) reconstruction update; and 4) surface prediction. On account of the lack of the loop closing process, which is proposed afterward, it has limit that the error accumulates as time passes.

Meanwhile, RGB-D SLAM is fundamentally not proper at the outdoor due to the two reasons. First of all, the infrared depth sensor is highly vulnerable under the sunlight. The following reason is that it is tough to operate the RGB-D SLAM at the mobile computer, since RGB-D SLAM requires large memory and high computing power. [14]

SLAM++ [18] is the object-oriented algorithm, which consists of four main steps: 1) camera pose estimation; 2) object insertion and pose update; 3) pose–graph optimization; and 4) surface rendering. This algorithm performs loop detection and increases efficiency by identifying repeatable objects. Since SLAM++ searches the object out of database to identify the object in the current frame, it is adept at the visited place, which means it is not proper at the vast outdoor environment.

Dense Visual Odometry is a key-frame based SLAM. [8] Each frame is created by a given interval and connected by graphs. *Keyframes*, which represent the posture of a car, become vertices of the graph, while edges of the graph represent the translation and rotation between vertices. Our SLAM is also follows this sort of key-frame based approach.

Likewise, visual SLAM algorithms have been developed in the form of multi-threads, and the recent state-of-the-art algorithms consist of three multi-threads: 1) camera tracking; 2) local mapping; and 3) loop closing. PTAM [10] started dividing visual SLAM into two categories: camera tracking and local mapping. Direct Sparse Odometry (DSO) [2] and ORB-SLAM2 [15] added loop closing and global BA. The system of this paper also follows the form of multi-threads and is built based on ORB-SLAM2. Hence, It is essential to understand each step of multi-threads of ORB-SLAM2.

Firts of all, camera tracking extracts the map points captured at the frames and estimates the poses of the vehicle. Comparing the map points between the current frame and the previous one, only meaningful frame that represents the certain location of the environment is decided as a keyframe. Next, local mapping thread builds a local map and optimizes the camera pose with the local map. This thread inserts new keyframes into a map, runs continuously, and extends the map. Lastly, loop closing thread adjusts the accumulated error. When the vehicle starts driving towards the unknown location, the error accumulates. Once the vehicle revisits a known location, the system detects it and adjusts the accumulated error. ORB-SLAM2 uses a covisibility graph and processes subgraph for loop closing, composed of the spanning tree. In this process, the loop is detected by the bag-of-words library DBoW2 [3], which makes the long-term data association possible. Bundle Adjustment is used three times at this system. Motion-only BA is conducted at the tracking step. Then, local Bundle Adjustment (local BA) is conducted, optimizing a local window of keyframes and points of the local map at the local mapping step. Lastly, A loop closure is followed by a full BA that optimizes all keyframes and points. The Levenberg– Marquardt method is used in this optimization process, implemented in g2o [12].

In this research, GPS data will be harnessed at the every multi-thread to adjust the error and lower the computing time in the outdoor for the mobile robot. As the GPS information is represent the actual location of the vehicle, this can help to improve the accuracy of the localization. Moreover, the price of GPS sensor is very cheap compared to other sensors, including 3D LiDAR and RGB-D camera.

C. Path planning

When the vehicles move the unknown environment, the strategy for trajectory is required. Path planning algorithm designs the path so that the vehicles are able to travel within the unknown environment with the least cost and without collision. In addition, when planning the movement of the vehicle, other information can be included, such as the movement of the obstacles or the movement of the other vehicles. Path planning algorithms can be classified into two types, Search based method and Sample based method.

1) Search based Method: Search based methods use the graph search method for finding the trajectory of the vehicle. In order to use the search based methods, the environment needs to be transformed into a discrete representation. The A* algorithm [6], which is similar with the Dijkastra's algorithm, guides users to the most optimistic states while saving significant amount of computation time compared to the Dijkstra's algorithm. The algorithm searches only those paths that succeed in connecting the nodes in the subgraph and selects the most efficient path based on the cost. Despite this, due to the discrete representation, A* only passes through the center of the gird. The accuracy depends on resolution of the grid [1]. Hybrid A* [1] provides a solution to this problem by ensuring that the path remains continuous state with each grid cell. This can make it possible to implement on the non-holonomic vehicle. Moreover, Hybrid A* predicts the motion of the vehicle depending on the various parameters of the vehicle such as the speed, steering angle and gear. D* algorithm [19] is proposed for finding the trajectory in the dynamic environment. D* generates the path considering the moving obstacles, which can easily re-plans using the cost due to the detected obstacles. However, D* has the high computational complexity, making it difficult to use in the large dynamic environment.

2) Sample based Method: For larger dynamic environments, sample-based methods are used by sampling the configuration without presenting a graph representation. While there are some sample-based methods capable of discretization, most methods search for the path in continuous space. RRT [13] creates a path in every region and expands to reach the desired destination. During this process, the path connects the nearest vertex selected according to the distance, and each vertex is generated at random. Also, the path expands more in the unexplored space, and this can make the path more optimal.The RRT Connect [11] combines two RRT method in which one starts from the starting point, and the other starts from the goal point, and they are connected by using heuristics. However, RRT Connect is not a asymptotically optimal planners due to the greedy tree connection [4]. BIT* [4] takes advantage of the search based methods by reducing the wasted computing effort from searching paths randomly in each process. BIT* searches for the path within each batch, and when the path is found, prunes the sample that are located outside of the path. In this way, the time complexity and the search space can be reduced. However, BIT* has some limitation to implement on the off-road vehicle since it only considers the distance, not the geographic feature of the road.

The proposed BIT* with GPS consider not only the distance of the path but also the geographic feature. In an off-road situation, our method facilitates the selection of an optimal path and a stable path for driving.

III. Methodology

A. System overview

The main parts of our off-road autonomous driving system operates sequentially; RGB-D SLAM of the perception stage preprocesses the RGB-D data and GPS data, and simultaneously localizes the posture of the vehicle and creates a map; When SLAM finishes, the final map is sent to the BIT* path planning algorithm of the decision stage; BIT* algorithm decides the trajectory from start point to goal point, taking into consideration not only time but also **stability** of the road. Meanwhile, GPS data performs the core role in the system, in that it makes it possible for SLAM to reduce the **computing time**, enhance the **accuracy**, and for Path Planning algorithm to differentiate the stable road from bumpy roads.

B. Monocular SLAM

Our SLAM was developed based on ORB-SLAM2 by adding new functions and conditions utilizing GPS data. Fig 00 shows the overall improvement points of every three multi-thread. In existing SLAM method, feature points are the only factor that processing the all stage. This is helpful to reduce the computational complexity, however, this might be unstable in certain environment. In our system, the GPS data is added to increase the stability.

1) Preventing bootless computing by GPS: Due to the rough ground at the off-road environment, there is a possibility that a vehicle can not moves or only moves a little even though it is given power input due to the environment such as the slope, mud, and bumpiness. Thus, it is necessary to prevent bootless computing for tracking when the vehicle moves a little. Using GPS data can solve this problem to calculate the distance of the vehicle's movement. When frames are created in certain interval, the GPS data of the vehicle is also measured and saved at each frame. Instead of tracking the vehicle at every frame, our system starts tracking only when the vehicle moves certain distance from the last frame and the distance is calculated using GPS data. If the shift in the position does not exceed certain threshold, tracking process is early stopped and system waits for next frame.

The Keyframe is also selected based on more strict condition. Useless Keyframes can disrupt the result of the SLAM, and they produce the additional computing. If the candidate Keyframe and the last Keyframe are placed in similar location, the candidate Keyframe should not be added in the map due to the similarity of the feature. If the candidate Keyframe is added, the system do calculate both of them even though they have huge similar features that are not helpful for making the map. For this reason, the system should extract only useful Keyframes. The distance between last Keyframe and candidate Keyframe is make possible to select the only useful Keyframe. If the distance is not big enough than the threshold distance, the candidate keyframe is not added in the map. For adding the condition based on the distance, the computing time can be reduced and the speed of the system can be faster than before.

2) KeyFrame Insertion and Covisibility Graph Update: Covisibility graph is the main structure in the system. For this graph, the calculating time is reduced but the result of the system is still qualitative. When new keyframe is inserted, the keyframes which is shared the same map points update their covisibility graph. In the covisibility graph, each keyframe is connected with keyframes, and the most correlated keyframe is connected closer than other keyframes. Using this graph, the system can assume the pose of the vehicle without comparing all keyframes. When building the covisibilty map, the feature points and distance are used to examine the correlation of keyframes. Only using feature data might be not enough to estimate the association. Even if they have largest similarity of features, it does not ensure they are the closet keyframes. Especailly in the off-road, the environment is not stable, and this characteristic can make the nearest keyframe does not have the largest similarity of features. For adding the distance between keyframes which is calculated using the GPS data, the system can consider not only the equivalence of features but also the distance between the keyframes. This can make the covisibility graph more accurately compared to existing covisibility map which uses only the similarity of the features.

3) Relocalization using GPS data: When the vehicle fail the tracking, it is important to find their location using the previous information. In this progress, the Keyframe database is used by comparing with the bag of words of current frame to get the Keyframe candidates which is similar with current frame. However, only using the Keyframe database might be not guarantee to find the location since the camera does not always extract the reliable data, especially in the outdoor. Moreover, a lot of computing time is needed as the system compares all of the keyframes. For this reason, the additional data should be used to make the vehicle can find the accurate location faster. GPS data can solve this problem as it represents the location of the actual world. First of all, the keyframe which is located far from current frame is excluded. This can reduce the computing time and also get only meaningful candidates to optimize the pose of current frame. After gathering the candidates, the pose of the candidate which has the most inlier points is optimize as current frame. In this stage, our algorithm is guided to search more inlier matches and more similar GPS data. Finally, the camera pose of current frame is optimized with the Keyframe which has sufficient inlier matches and similar GPS data.

C. Path Planning

This section is a explanation about the suggested Path Planning methods in detail. Our algorithm is based on Bath Informed Trees (BIT*) which is well-known Sampling based Path Planning algorithm [4]. Path Planning algorithm basically searches the shortest path. However, it is essential to understand the environmental factors that can disrupt the result of the algorithm. Moreover, to adjust the path planning in the off-road, considering the characteristic of the environment became more important. Therefore, Our algorithm can consider the distance with the obstacle to reduce the error which is produced by the environment. The stability of the road is also considered for planning the stable path. In this section, first, we explain our basic structure which is used in BIT* and introduce new algorithm which focus on the stability of the path and the clearance of the object.

1) BIT star: BIT* converges to the global optimum heuristically adding multiple batches of samples in the configuration space. It is consisted of implicit Random Geometric Graph (RGG) and explicit spanning tree, and finds the path by expanding the spanning tree as connecting the vertexes which is placed in certain range [17]. When vertex v and x, c(v,x) represents the cost between two vertexes. v,x $\in$ X About the explicit tree, τ, the sampled state x, *cost-to-go*, *cost-to-come* represents $\hat{g}(x)$, $\hat{h}(x)$ respectively, and the algorithm proceeds extending the τ to minimize the cost function $\hat{f}(x) := \hat{g}(x) + \hat{h}(x)$. In the current tree, the *cost-to-come* value represents $gt(x)$, and if $gt(v) + c(v, x) < gt(x)$ is true, vertexes are extended. When it comes to the path planning which computes the shortest path, $c(v, x)$ represents the distance between two vertexes. However, our algorithm computes the cost function $c(v, x)$, as a weighted aggregate of the path length, path clearance, and path stability.

$$c(v, x) = \alpha * dist(v, x) + \beta * clear(v, x) + \gamma * stable(v, x)$$

2) clearance path: To reduce the possibility of the crash with the obstacle, it is required to ensure the safety of the path. Moreover, as our path planning is more focus on the off-road, considering the condition of the road and sporadic obstacles is more important. Path clearance, which estimates the clearness of the space [5], is one of the possible option that can prevent the accident. By setting the threshold of the distance with the obstacle and optimizing it, our algorithm can guarantee the stability of the path. Determining the distance with the object in current state is processing with the condition that whether the obstacle is existed in curtain boundary. As Fig , The number of pixels is used to set the boundary.

3) stable path: To ensure the stable path, the stability function is added in the optimization criteria. Path stability, which is measured at the SLAM process, is applied for the weight of Stable path.

IV. Experimental Results

A. GPS SLAM evaluation

In the experiment, The Datasets which are video data and GPS data were collected using GoPro Hero 10, the frame was set as 240, and we tested the ORB-SLAM2 and GPS-SLAM in the same condition using ASUS VivoBook which is .

1) Overall result: First of all, we compared our SLAM result with ORB SLAM2 with the data that we extracted. Figure 2 shows the results of GPS Mobile SLAM and ORB SLAM2. Our testing data was hard to extract the feature, and this makes the ORB SLAM2 hard to make the accurate map. In contrast, even though the features are not extracted well, GPS Mobile SLAM use not only the features but also the GPS, and this can help the system can make more accurate map. Also, in the Table I, the specific aspects which are the number of tracking frames, keyframes, map points, relocalization in each SLAM method are compared. As you can see the ORB SLAM2 has more mappoint and the keyframe, however, it does not help to make a accurate map. This means the system failed to notify the same location, and they keeps add the keyframes in the map rather than culling the keyframes and the keyframes. As we mentioned before, in the outdoor, their are a lot of various circumstance and even if it is same place, the feature can be changed. For this reason, the ORB SLAM2 is disturbed when making the map. Moreover, GPS Mobile

TABLE I: The number of Keyframes and frames that is extracted by GPS Mobile SLAM

Aspects of Comparison	OURS	ORB SLAM2
Number of Keyframes	878	1687
Number of Map Points	27088	36880
Number of eliminated frames	470	432

SLAM can reduce the computing time than ORB SLAM2 even though the result of the system seem more clear.

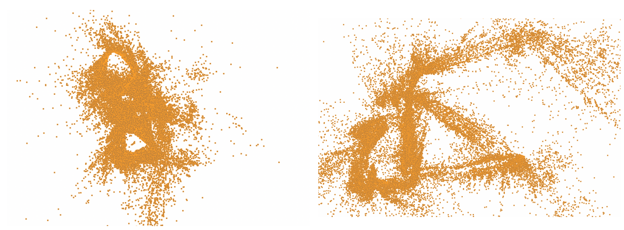

Fig. 2: The result of ORB-SLAM2 and GPS Mobile SLAM.

2) The influence usign GPS data when adding the keyframe: To compare the advantage of reducing computational complexity, we test the original GPS Mobile SLAM and GPS Mobile SLAM without additional condition when adding the new keyframe. The results of the system and the number of eliminated frames, map points, keyframes are compared. Figure 3 can show the results of the systems. Even though the Keyframe has less keyframe than the By including additional condition using the GPS data can help the system can add only meaningful Keyframes that can build the accurate map. Additionally, this can make it possible to prevent calculating the useless Keyframes. Table II shows the actual number of eliminated Keyframes. Even though the added keyframes are less in the system that is added the comparing location than the other, the result is more accurate. Since adding the condition that checking the distance with the last keyframe can make the system to add only the candidate keyframe which does not have huge similarity.

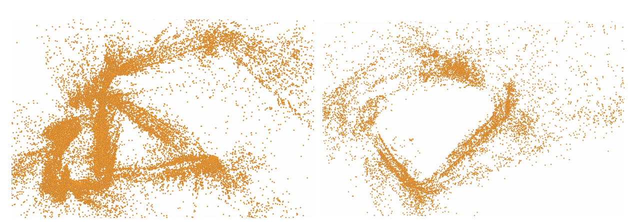

Fig. 3: The result of GPS Mobile SLAM and GPS SLAM without Adding Keyframe condition.

3) the influence of using GPS data when relocalization: When doing relocalization, the GPS data is used to calculate the possibility of the frame. The system compares the GPS

TABLE II: Comparing the Keyframes and eliminated frame that is extracted using GPS mobile SLAM and GPS mobile SLAM without the additional condition when adding the keyframe.

Aspects of Comparison	Ours	Ours w/o additional condition
Number of Keyframes	878	360
Number of Map Points	27088	12693
Number of tracking frames	15493	17210
Number of eliminated frames	470	387

data and eliminates the keyframes which are far from the frame. This can make the system search only the possible options and guarantee to find the location of the vehicle faster. Figure 7 shows the results of the map that which include the comparison the GPS data or not. As comparing the GPS data, the system can make more accurate map. Table III shows the comparison of the aspects. Using the GPS data, the system can find the location more frequently then the other. Even if the vehicle revisit the location, the frame is not always same and this difference disturb the system find the revisit place, eventually hard to relocate the vehicle. In contrast, the GPS data help to find the location, and finally the system does less Relocalization than the one without the additional condition.

Fig. 4: The result of GPS Mobile SLAM and GPS Mobile SLAM without Relocalization condition.

TABLE III: Comparing the Keyframes and eliminated frame that is extracted using GPS mobile SLAM and GPS mobile SLAM without the additional condition when Relocalization.

Aspects of Comparison	Ours	Ours w/o additional condition
Number of Keyframes	878	804
Number of Map Points	27088	25786
Number of Relocalization	8777	9317

B. BIT Path Planning evaluation*

Our path planning algorithm is evaluated as avoiding various obstacles depending on the state of the path in two different situations. The state of each state is calculated using the RGB data of the pixel on the map. The two different environments are shown in Fig. 5 and 7, respectively.

The first environment is a complex maze with many obstacles in 5. 5 (a) and (b) are the results of the BIT* and our algorithm, respectively. BIT* only considers the path length from the start point to the destination. However, our path planning is conducted while maintaining the distance from the obstacle while finding the paths. The distance to the obstacle is calculated according to the gray scale of the pixel color on the map. In Fig. 6, we shows the distance from the obstacle at each sampled point. It is shown that the distance to the obstacle is maintained in each direction finding step. Therefore, it reduces the risk of collision in an outdoor environment with complex obstacles.

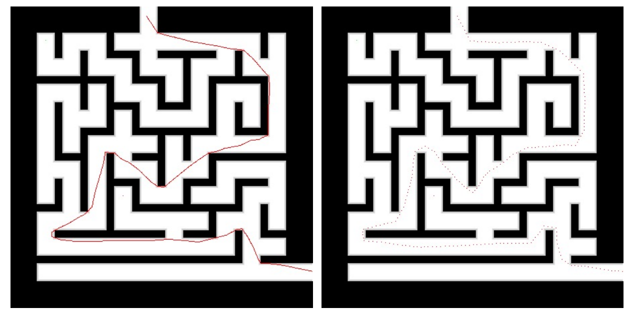

Fig. 5: The result of BIT* and our path planning in the maze environment

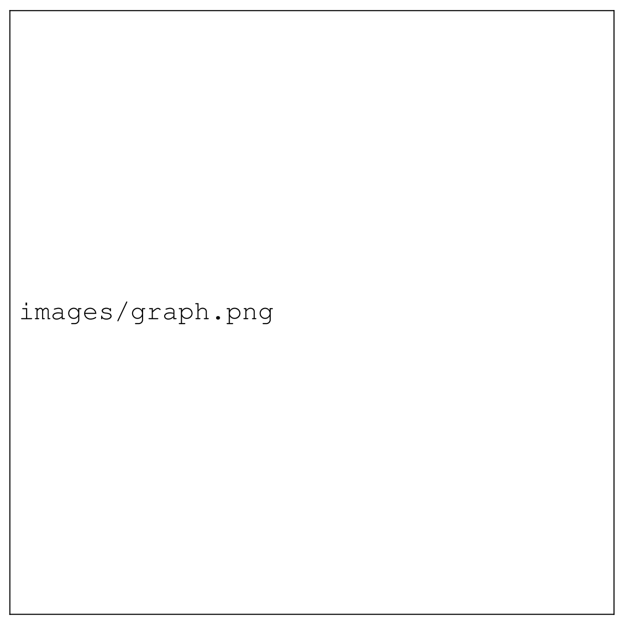

Fig. 6: Accumulated graph of distance from obstacle in each state

Fig. 7 shows the result of path planning in the outdoor environment where the slam was performed. In an environment with buildings, grass, trees, and roads, GPS was used to color the state of the stable path and use it for path planning.

Our path planning improves the path to a stable path by weighting the stable state among the sampled states. Table IV shows the state of the sample contained in the entire path to indicate how stable the entire path selects.

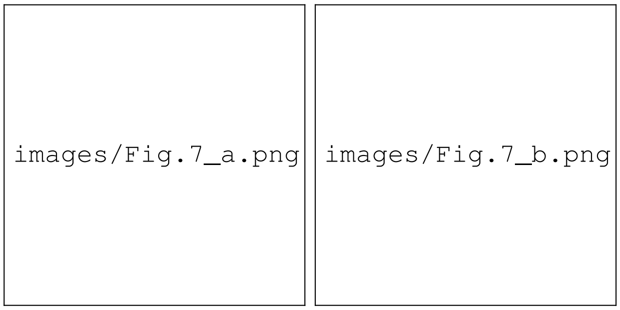

Fig. 7: The result of BIT* and our path planning in outdoor environemt

TABLE IV: The number of states sampled across the path.

Sampled State	BIT*	Ours
Free state	878	804
Stable state	1910	1782
Total state	2788	2586

V. Conclusion

In this study, we compared the performance of SLAM and path planning process with and without GPS in outdoor environments. Due to specific features of the environment, algorithms with GPS data showed more robust performance. Bootless computing during the SLAM process was prevented by adding GPS data into the Keyframe extracting algorithms. In previous studies, tracking process had to be conducted in every step. However, in our case, the system starts tracking only when the vehicle moves a certain distance. Moreover, by comparing the distance between a candidate Keyframe and the last Keyframe with GPS data, only useful Keyframes are added in the map. Therefore, the accuracy of the map increases while the amount of computation is reduced. In addition, the GPS data was added while calculating the *covisibility graph* and the data ensures that certain Keyframes are near each other. GPS data is used in the relocalization process, the process of finding the vehicle's position when the tracking fails. For the path finding algorithm to be suitable for outdoor environments, we improved it by incorporating two factors to the algorithm. *Clearance* factor is the distance between the vehicle ans the obstacles, which is presented as color difference in the map based on the GPS data. *Stability* factor contributes in determining whether a certain road is a stable path or not by counting the spread dots on the path. There is a higher probability of a path where there is more dots. In conclusion, these SLAM and path planning algorithm with GPS data showed higher performance in the outdoor environment than using only the vision algorithm which in this case is ORB-SLAM2. For this study uses camera and GPS sensors which is relatively easier to acquire than LiDAR sensor, this kind of approach will contribute to the popularization of SLAM.

Acknowledgment

References

[1] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Practical search techniques in path planning for autonomous driving. *Ann Arbor*, 1001(48105):18–80, 2008.

[2] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[3] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[4] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 3067–3074. IEEE, 2015.

[5] Roland Geraerts and Mark H Overmars. Clearance based path optimization for motion planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 2386–2392. IEEE, 2004.

[6] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[7] Kaijin Ji, Huiyan Chen, Huijun Di, Jianwei Gong, Guangming Xiong, Jianyong Qi, and Tao Yi. Cpfg-slam: A robust simultaneous localization and mapping based on lidar in off-road environment. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 650–655. IEEE, 2018.

[8] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013.

[9] Eungcheol Kim, J Kim, and Myoungho Sunwoo. Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics. *International Journal of Automotive Technology*, 15(7):1155–1164, 2014.

[10] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007.

[11] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[12] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.

[13] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[14] Andréa Macario Barros, Maugan Michel, Yoann Moline, Gwenolé Corre, and Frédérick Carrel. A comprehensive survey of visual slam algorithms. *Robotics*, 11(1):24, 2022.

[15] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[16] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. IEEE, 2011.

[17] Mathew Penrose. *Random geometric graphs*, volume 5. OUP Oxford, 2003.

[18] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[19] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3310–3317 vol.4, 1994.

[20] Marlin P Strub and Jonathan D Gammell. Advanced bit*(abit*): Sampling-based planning with advanced graph-search techniques. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 130–136. IEEE, 2020.

[21] Han Wang, Hongjun Zhang, Kun Wang, Chen Zhang, Chengxiang Yin, and Xingdang Kang. Off-road path planning based on improved ant colony algorithm. *Wireless Personal Communications*, 102(2):1705–1721, 2018.