2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

# Real-Time 6D Lidar SLAM in Large Scale Natural Terrains for UGV

Zhongze Liu, Huiyan Chen, Huijun Di, Yi Tao, Jianwei Gong*, Guangming Xiong, Jianyong Qi

***Abstract*— Simultaneous Localization And Mapping (SLAM) plays a more and more important role in the environment perception system of Unmanned Ground Vehicle (UGV), most SLAM technologies used to be applied indoor or in urban scenarios, we present a real-time 6D SLAM approach suitable for large scale natural terrain with the help of an Inertial Measurement Unit(IMU) and two 3D Lidars. Besides dividing the entire map into many submaps which consists of large numbers of tree structure based voxels, we use probabilistic methods to represent the possibility of one voxel being occupied/null. A Sparse Pose Adjustment (SPA) method has been used to solve 6D global pose optimization with some relative poses as pose constraints and relative motions computed from IMU data as kinetics constraints. A place recognition method integrated a method named Rotation Histogram Matching (RHM) and a Branch and Bound Search (BBS) based Iterative Closest Points (ICP) algorithm is applied to realize a real-time loop closure detection. We complete global pose optimization with the help of Ceres. Experimental results obtained from a real large scale natural environment shows an effective reduction for Lidar odometry pose accumulative error and a good performance for 3D mapping.**

## I. Introduction

The recent literature in UGV shows a growing interest in SLAM approaches. With the development of UGV technology, traditional 2.5D elevation map's disadvantages appear such as existence of blind zone, absence of information in z orientation, sensitive to noises, etc. In addition, because of the existence of thick forests and tunnels in natural terrains, UGV can't get accurate location information. A 6D SLAM system suitable for large scale natural terrains is aspired. Plenty of vision-based 6D SLAMs performance such as [1], [2], can be easily influenced by the change of illumination and weather. Little SLAM approaches are designed for UGVs working in natural rough terrains due to the huge amount of environment data, high requirements of processing efficiency and the complexity of the scenarios. With the efficiently computation relative to computer vision-based method, We proposed a 6D Lidar SLAM system with two 3D Lidars and an IMU as sensors, which can meet the demands of UGV's real time autonomous driving in large scale natural terrains, and two 3D Lidars can obtain more environment information from once observation and smaller blind area than one. This system is divided into two subsystems: 3D mapping and 6D global pose optimization. Through 3D mapping, a global hybrid map combining the advantages of 3D grid map and topological map is implemented for UGV to deal with real-time traversability analysis in natural rough terrains. The computation efficiency is promoted a lot by dividing an entire global map into many submaps. The integrality of the global map, meanwhile, isn't damaged through organizing all submaps with the help of topology. In addition, inspired by the octree map in [6], voxels maintained in submaps are organized with tree-structure to meet the requirements for UGV's real time environment perception. A SPA algorithm presented in [7] is applied to reduce the accumulated error of the Lidar odometry to solve the 6D global pose optimization problem.

The global poses of all Lidar scans and all submaps are optimized as nodes by constructing a pose graph [8], and three kinds constraints called inter-pose constraints, intro-pose constraints and kinetics constraints, are built as edges in the pose graph with the aid of an IMU and two 3D Lidar. Inter-pose constraints are obtained through loop closure detection which apply a place recognition method called GLARE [9] and an ICP algorithm improved by BBS to obtain a real time detection.

## II. Related Work

Lots of SLAM systems are performed with computer vision technologies, such as [1],which presented an ORB feature-based monocular SLAM system operating in real time, in small and large indoor and outdoor environments. Visual and inertial measurements are combined in [2] to realize an accurate visual-inertial odometry. But the performance of vision-based methods can be easily influenced by the illumination and weather. Some Lidar-based SLAM systems are also presented such as [3], [4], [9]. The ICP algorithm [5] is extended in [6] to obtain a method for continuous-time trajectory estimation suitable for SLAM which takes into account inter-sample pose errors. Nchter et al. [10] presented a robotic outdoor SLAM system based on consistent 3D laser range scans, which need a heuristic for closed loop detection. 3D mapping is necessary for UGV to digitalize environment. To satisfy the requirements for a real-time process, the map data must be well organized with a good digital scenario model as a prerequisite. Popular three-dimensional mapping models include point clouds, surfels, octrees, and voxel grids. The point cloud map describes the environment with a high metric accuracy [10], [11], [12], but it can not process map data timely to express the changes in the environment. The

*This work was supported by the National Natural Science Foundation of China under Grant 61703041, Beijing Institute of Technology Research Fund Program for Young Scholars and Key Laboratory of Biomimetic Robots and Systems, Beijing Institute of Technology, Ministry of Education (Project Grant No. 2017CX02005).

Zhongze Liu, Huiyan Chen, Huijun Di, *Jianwei Gong, Guangming Xiong, Jianyong Qi are with Beijing Institute of Technology, Beijing 100081, China (e-mail: 2120160359@bit.edu.cn, chen_h_y@263.net, ajon@bit.edu.cn, corresponding author, gongjianwei@bit.edu.cn, xiongguangming@bit.edu.cn, qijianyong@bit.edu.cn).

Yi Tao is with Beijing Special Vehicle Academy, Beijing 100081, China (e-mail: taotaoyiyi@163.com).

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

Surfel map will work well in scenarios reach in surface information [13], but natural outdoor environment is lack of surfaces. Douillard et al. [14] combine object voxel maps with a coarse elevation map to process tree-dimensional segmentation, they do not organize submaps in a hierarchy which causes the low efficiency when dealing with the map data updating. Kai et al. [6] presented a novel multi-resolution approach to efficiently mapping 3d environments with a octree-based probabilistic 3D map, which uses an octomap [15] algorithm as reference.

The poses obtained from the Lidar or vision odometry contains accumulated error, which must be reduced to achieve more accurate global poses. Mainstream global pose optimization methods include particle filter, Extended Kalman Filter (EKF) and graph-based SLAM. Grisetti et al. use a particle filter to solve the SLAM problem taking into account not only the movement of the robot but also the most recent observation. But particle filter must represent the entire system state in each particle, which will be resource consumable in large natural terrains. The EKF SLAM in [17] lies strongly in the assumptions on both the UGV motion and the sensor noise, which is hard to achieve for a natural environment. Most graph-based SLAM approaches deal with a set of nodes which represent UGV poses and world features, and edges connecting nodes as constraints. A g2o optimization framework has been presented based on the graph theory [18]. An Efficient Sparse Pose Adjustment method in [20] can work out the real-time online optimization for a large scale SLAM.

## III. System Overview

This paper aims at achieving a practical real-time SLAM in large scale natural terrains. as is shown in Figure 1, the SLAM scheme is splitted into two parts: 3D mapping and 6D global pose optimization. The angular velocities, linear accelerations from an IMU and the point clouds from two 3D 32-E Lidar are grabbed as the input of this SLAM system.

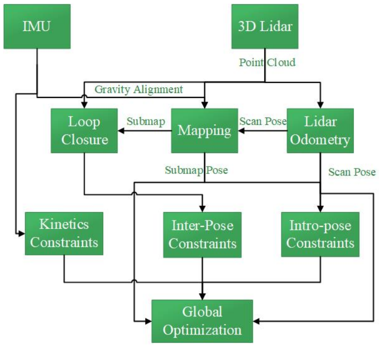


Fig. 1. The real-time 6D Lidar SLAM system framework

To achieve a timely updating, the global map is splitted into many submaps organized with the topology inspired by [23], so that each submap can be updated and transformed individually. Each submap is regarded as a map-node in the global topological structure. Actually, the connectivity of submaps forms the map-edges in the topological map, ultimately an entire hybrid global map is obtained which combines the advantages of voxel grid map and topological map (Figure 2).

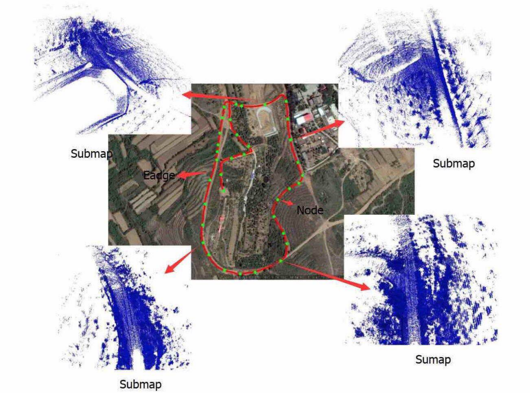


Fig. 2. Global hybrid map. Each node in the map represents a submap node, which include the submap as data, and the edges between nodes represents relations between submaps.

A SPA optimization method based on the graph theory is adopted to solve the 6D global pose optimization problem. The sparse pose graph has both Lidar poses and submap poses as nodes, with measurements connecting them as constraints. All the constraints can be classified into tree categories which are inter-pose constraints, intro-pose constraints and kinetics constraints. The relative poses between the submap pose and the Lidar poses in the submap are regarded as intro-pose constraints. Inter-pose constraints are computed from the loop closure detection which combines RHM method and BBS-based ICP algorithm together to realize a real-time loop closure detection. The velocity change and angular change obtained from the acceleration and angular velocity which are measured by an IMU are used as a kinetics constraint between two adjacent nodes. A Ceres solver [21] is adopted to computer the optimization function.

## IV. 3D Mapping

The global hybrid map is constructed by creating submaps one by one and connecting every submap with the help of the topology theory. The 3D mapping approach includes three parts which are mapping initialization, mapping updating and mapping ending.

### A. Mapping initialization

When we create the first submap of the global hybrid map, the pose of the initial submap is set to be $\xi_0^S(0,0,0,g_0)$, and the frame of point cloud pose is set to be $\xi_0^C(0,0,0,0,0,0)$. $g_0$ is measured by the IMU when the UVG is still. The following submap poses are set to be

$$\xi_i^S = (\xi_x^i, \xi_y^i, \xi_z^i, g_i)$$

and the point cloud poses are set to be

$$\xi_j^C = (\xi_x^j, \xi_y^j, \xi_z^j, \xi_p^j, \xi_r^j, \xi_{yawl}^j)$$

which are computed from the Lidar odometry algorithm, and the $(\xi_x^i, \xi_y^i, \xi_z^i)$ is the location of the first point cloud which is going to be inserted into the submap. $g_i$ is updated by

$$\begin{cases} \Delta t = t_k - t_{k-1} & k = 1,2,3\dots \\ \alpha = 1 - \frac{1}{e^{\frac{\Delta t}{gc}}} \\ g_k = \alpha g_{k-1} + (1-\alpha) g_{imu} \end{cases} \tag{1}$$

where $t_k$ and $t_{k-1}$ are two neighboring sample times of the IMU data. $g_c$ is the local gravity constant, $g_{imu}$ is the current acceleration measured by IMU, $g_i$ consists with the closest $g_k$ in time.

### *B. Mapping updating*

Map updating is a iterative process of repeatedly aligning points and submap coordinate frames. We can represent one frame of point clouds as $C = \{c_k\}_{k=1,\dots,K}, c_k \in \mathbb{R}^3$ in the Lidar coordinate frame, which can be transformed into submap frame through the transformation

$$T_{\xi_j} = \xi_i^{-1S} \cdot \xi_j^C$$

with the help of

$$\begin{cases} T_{\xi_j} h_k = R_{\xi_j} h_k + t_{\xi_j} \\ R_{\xi_j} = \begin{pmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{pmatrix} \\ t_{\xi_j} = \left(\xi_x^i, \xi_y^i, \xi_z^i\right)^T \\ a11 = \cos r \cos yawl - \sin r \sin p \sin yawl \\ a12 = -\cos r \sin yawl - \sin r \sin p \cos yawl \\ a13 = -\sin r \cos p \\ a21 = \cos p \sin yawl \\ a22 = \cos p \cos yawl \\ a23 = -\sin p \\ a31 = \sin r \cos yawl + \cos r \sin p \sin yawl \\ a32 = -\sin r \sin yawl + \cos r \sin p \cos yawl \\ a33 = \cos r \cos p \end{cases} \tag{2}$$

when the relative motion between two poind cloud frames exceed a certain motion filtering threshold or a certain period of time has passed.

The submap is constructed as one 3D grid voxel map, where all the grid voxels are managed with tree structure to balance the efficiency of data reading and writing such as Figure 3.The meta-nodes in the dynamic extension layer are managed with the chain structure and the middle-nodes and voxels in meta-nodes are organized by the tree structure. Through this way, the advantages of the high indexing speed for the chain structure and the high modification efficiency for the tree structure are combined together. The number of meta-nodes grow with the extension of environment observation. For example, when the resolution of the submap is $0.1m$, which is the length of the voxel, a menta-node can represent a scale of $6.4m \times 6.4m \times 6.4m$ of the current environment. A few consecutive poind clouds are obtained to construct a submap. As is shown in Figure 4, whenever a point cloud is to be inserted into the submap, the state of each voxel can be classified into three types: hit, traverse and irrelevance. A probabilistic method is adopted to update the probability in the voxel, which is set to be zero as the initial value. If one voxel is observed for the first time, a probability $p_{hit}$ of $p_{traverse}$ will be assigned. The bayes' theorem is adopted to update the probabilities in the formerly observed voxels.

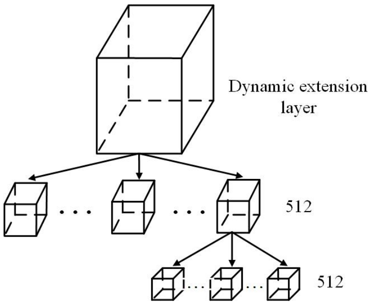


Fig. 3. submap structure schematic diagram. The structure of the entire submap can be divided into three layers - dynamic extension layer, middle layer and voxel layer. the number of meta-nodes in dynamic extension layer will increase with the extension of the environment observation. each dynamic extension node consists of 512 middle-nodes in the middle layer and each nodes consists of 512 voxels in the voxel layer.

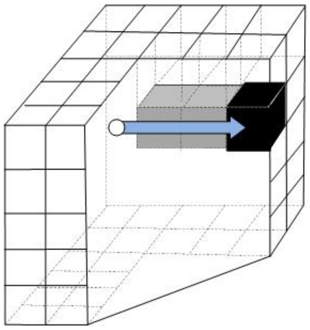


Fig. 4. A laser and voxels associated with hit (black), traverse (shaded) and irrelevance (transparent).

### *C. Mapping ending*

Every a few seconds, a submap construction will be finished and a new one will start to be created. A part of the data between two adjacent submaps overlaps so as to ensure the smoothness during the submap switching. As a node in the global hybrid map, once a submap construction is finished, a series of attributes such as submap ID, original submap pose, optimized submap pose, terrains and relationships with other nodes will be assigned. Actually the relationships between submaps form the edges in the topological structure. Eventually a complete global hybrid map (Figure 2) is formed which combines the advantages of the absolute grid voxel map and the topological map.

## V. Global 6D Pose Optimization

Recognizing formerly observed places is necessary to detect a loop closure, which is an efficient method to eliminate accumulated errors. Integrated with Branch and Bound Search and Rotation Histogram Matching algorithms, the Iterative Closest Points algorithm is adopted to find a most accurate relative pose $\xi^\star$ between the current point cloud frame and the formerly finished submaps, define as

$$\xi^\star = \arg\min \sum_{i=1}^{N_s} \sum_{k=1}^{N_k} S^i_{nearest}(T_\xi c_k), \xi \in W_i \tag{3}$$

where $S^i_{nearest}$ is the submap $S_i$ extended to all of $\mathbb{R}^3$ by rounding its arguments to the nearest grid voxel, whose probability is updated in the meantime. $S_i$ is obtained by distance filtering, and $W_i$ is the search window corresponding to $S_i$. The search window only includes information in four dimensions in this step with respect to the fact that the movement of UGV in large scale terrains mainly changes in x, y, z and heading angle. The step size of heading angle $\delta_{yawl}$ is carefully computed to achieve an accurate matching as

$$\begin{cases} \delta_{yawl} = \arccos(1 - \frac{r^2}{2d^2_{\max}}) \\ d^2_{\max} = \max_{k=1,\ldots,K} ||c_k|| \end{cases} \tag{4}$$

where $r$ is the resolution of the submap. After the linear and heading angle search window sizes has been set, we obtain the search window

$$\begin{cases} w_x = \lceil \frac{W_x}{r} \rceil, w_y = \lceil \frac{W_y}{r} \rceil, w_z = \lceil \frac{W_z}{r} \rceil, w_{yawl} = \lceil \frac{W_{yawl}}{\delta_{yawl}} \rceil \\ \bar{W} = \{-w_x, \ldots, w_x\} \times \{-w_y, \ldots, w_y\} \times \\ \qquad \{-w_z, \ldots, w_z\} \times \{-w_{yawl}, \ldots, w_{yawl}\} \\ W = \{\xi_0 + (rh_x, rh_y, rh_z, rh_{yawl}) : \\ \qquad (h_x, h_y, h_z, h_{yawl}) \in \bar{W}\} \end{cases} \tag{5}$$

To efficiently compute $\xi^\star$ over so large search windows, the computation is divided into two steps. In the first step, a Rotation Histogram Matching method is adopted inspired by the GLARE [9] to obtain a best $h^\star_{yawl}$ in the search window.At first, the current poind cloud is rotated by $h_{yawl}$ and sliced into several slices according to the z information, and then every point in the cloud is sorted according to the angle. If a point $p_i$ is surrounded by some points $\{p_m\}_{m=1,\ldots,M}, p_m \in \mathbb{R}^2$ which are within the Euclidean distance threshold, the point will be selected as a landmark and a value will be calculated corresponding to the angle of $p_i$ just as the Figure 5 shows, define as

$$v_{\theta i} = \sum_{l=1}^{L} \sum_{m=1}^{M} (1 - \frac{(p_m - p_i) \cdot (p_m - p_0)}{|p_m - p_i||p_m - p_0|}) \tag{6}$$

We add all the values with the same angle in all slices, and map the angle from $0 - \pi$ to $0 - 255$. Eventually the histogram of the current point cloud is obtained corresponding to the angle of $p_n$ just as the Figure 5 shows, define as $H_c = (v_0, \ldots, v_{255})$. The histograms $H_{ij}, j = 1, \ldots, J$ of all point clouds in the submap $S_i$ are also calculated. The $h^\star_{yawl}$ will be obtained by

$$h^\star_{yawl} = \arg \max_{h_{yawl} \in W_{yawl}} H_{ij} \cdot H_c \qquad j = 1, \ldots, J \tag{7}$$

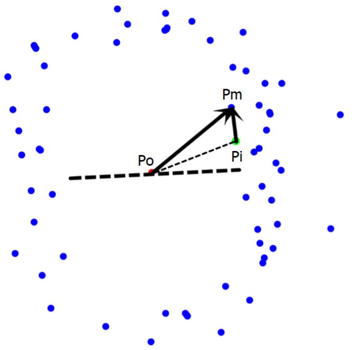


Fig. 5. Rotation Histogram Matching sketch graph. $p_0$ is the center of this slice, and $p_i$ is the selected landmark. $p_m$ is the points around the landmark.

If the $h^\star_{yawl}$ surpassed the rotation score threshold , a BBS based ICP algorithm applied in [24] will be computed to obtain the $\xi^\star$. Eventually, the best relative pose $\xi^*$ is computed with the help of Ceres as

$$\min_{\xi^*}(\sum_{k=1}^{N_k} \rho_1(1 - S^i_{nearest}(T_\xi c_k)) + \ \rho_2(t_\xi - t^\star) + \rho_3(R^{-1\star} R^k_\xi)) \tag{8}$$

where the $\xi^\star$ is the relative pose calculated before between current point cloud frame and the submap $S^i_{nearest}$.

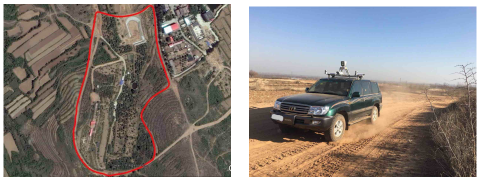
### A. Pose optimization

The Sparse Pose Adjustment algorithm is adopted to optimize the poses generated during the movement of the UGV with the help of Levenberg-Marquardt (LM) method. The poses are calculated by the Lidar odometry without any optimization, which include all the submap poses $\Xi^S = \{\xi^S_i\}_{i=1,\ldots,S}$ and point cloud poses $\Xi^C = \{\xi^C_j\}_{j=1,\ldots,C}$. All the constraints can be classified into three types which are inter-pose constraints, intro-pose constraints and kinetics constrains. The inter-pose constraints are generated by loop closure detection and the intro-pose constraints are relative poses between each submap pose and the point cloud poses in the submap. The kinetics constraints are calculated on the basis of the IMU data, which include the velocity change $\delta v_{imu}$ and $\delta R_{imu}$. We use Ceres to solute the optimization problem as

$$\arg \min_{\Xi^S, \Xi^C} \frac{1}{2} \sum_{ij} \rho(E^2(\xi^S_i, \xi^C_j, \Sigma_{ij}, \xi_{ij})) \tag{9}$$

where the constraints $\xi_{ij}$ contain both inter-pose constraints and intro-pose constraints. The covariance matrices $\Sigma_{ij}$ are estimated by Ceres and the residual $E$ is computed by

$$\begin{cases} E^2(\xi^S_i, \xi^C_j, \Sigma_{ij}, \xi_{ij}) = \ e(\xi^S_i, \xi^C_j, \xi_{ij})^T \Sigma^{-1}_{ij} e(\xi^S_i, \xi^C_j, \xi_{ij}) \\ e(\xi^S_i, \xi^C_j, \xi_{ij}) = \xi_{ij} - \begin{pmatrix} R^{-1}_{\xi^S_i}(t_{\xi^S_i} - t_{\xi^C_j}) \\ \xi^S_{i;p} - \xi^C_{j;p} \\ \xi^S_{i;r} - \xi^C_{j;r} \\ \xi^S_{i;yawl} - \xi^C_{j;yawl} \end{pmatrix} \end{cases} \tag{10}$$



(a) Experimental Scene (b) Experimental Platform

Fig. 6. Experimental scene and platform. The red route is drawn according to the data obtained by the RTK-GPS mounted on the UGV.

Another problem is computed to consider the kinetics constraints as

$$\arg\min_{\Xi^C} \frac{1}{2}\sum_j \rho_K(E_K^2(\xi_j^C, \Sigma_j^K, C_j^K)) \qquad j = 2, \ldots, C-1 \quad (11)$$

where $C_j^K$ are the kinetics constraints. The covariance matrices $\Sigma_j^K$ are estimated by Ceres and the residual $E_K$ is computed by

$$\begin{cases} E_K^2(\xi_j^C, \Sigma_j^K, C_j^K) = & e(\xi_j^C, C_j^K)^T {\Sigma_j^K}^{-1} e(\xi_j^C, C_j^K) \\ e(\xi_j^C, C_j^K) = C_j^K - & \begin{pmatrix} \frac{t_{\xi_{j+1}^C} - t_{\xi_j^C}}{\delta time_1} - \frac{t_{\xi_j^C} - t_{\xi_{j-1}^C}}{\delta time_0} \\ \xi_{j;p}^C - \xi_{j-1;p}^C \\ \xi_{j;r}^C - \xi_{j-1;r}^C \\ \xi_{j;yawl}^C - \xi_{j-1;yawl}^C \end{pmatrix} \end{cases} \quad (12)$$

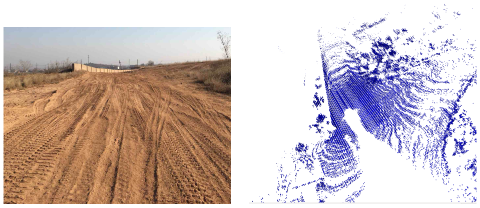
where $\delta time_1$ is the time period from the point cloud pose $\xi_j^C$ to the point cloud pose $\xi_{j+1}^C$, and $\delta time_0$ is the time period from the point cloud pose $\xi_{j-1}^C$ to the point cloud pose $\xi_j^C$.

## VI. Experimental Results

Many experiments have been made at large scale natural terrains. Fig.6 shows one of our experimental scenes (left) and our experimental platform (right). Many terrains, including bumpy roads, rampways, forks, concrete roads and bridges, are included in the experimental route. Two 32E velodyne Lidar are mounted at the top of the vehicle to obtain point clouds corresponding to the experimental, and an IMU is installed at the center of the vehicle's mass, which can measure the linear accelerations and angular velocities of the UGV in real time. Besides, a RTK based GPS is mounted to get the platform's exact positions and another high accurate IMU is adopted to obtain the reference postures so that we can evaluate our pose optimization performance. All the computation runs on an Intel(R) Core(TM) i7-7700 HQ CPU at 2.82 Hz.

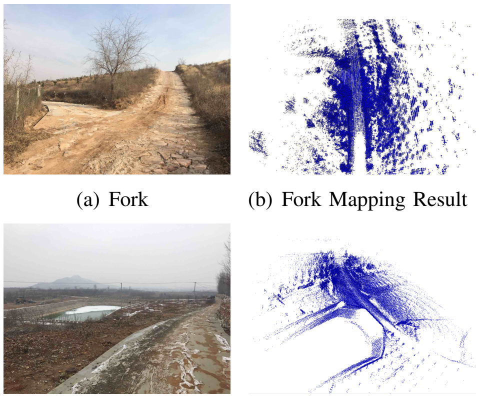
### A. 3D mapping

The environment is reconstructed in the 3D mapping process with the UGV running at a speed of approximately 20Km/h. Figure 7 shows a wall on the bumpy road (left) and the mapping result (right) shows a good performance in so bumpy a road. Figure 8 shows the mapping results of the fork and the bridge.



(a) Bumpy Terrain (b) Mapping Result

Fig. 7. Experimental scene and platform.



(a) Fork (b) Fork Mapping Result

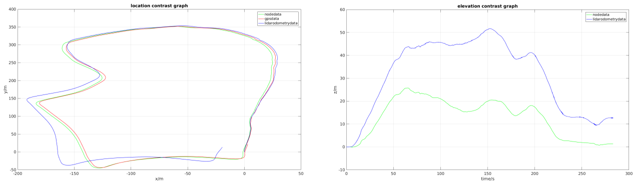
(c) Bridge (d) Bridge Mapping Result

Fig. 8. Natural terrains and mapping results.

### B. Global pose optimization

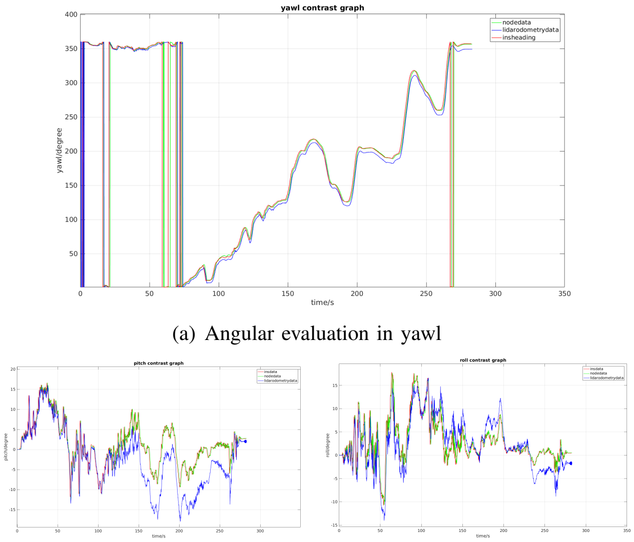
The performance of the pose optimization is evaluated in a real large scale natural terrain which is shown in Figure 6. Figure 9 shows the translation results in the pose optimization. With a route about 1.5 km, the accumulated x-y error reaches 21.5 m when the UGV returns to the start position after a loop driving, and the accumulated error of z comes to 9.5 m without optimization. The offsets of both x-y and z are optimized within 1 m with a loop closure obtained successfully. As we can see from the figure, the optimized location is further away from the reference position at around $(25, 250)$, which is due to that the local location accuracy is sacrificed a little to achieve the optimization in the global location.

The angular optimization performance is also evaluated in Figure 10. We can see that the pose estimation results from the Lidar odometry are accurate in the short period,



(a) Location in x-y (b) Location in z

Fig. 9. Translation contrast graph. The red curve is drawn according to the RTK-GPS data, which is regarded as the reference of the accuracy here. The green curve is drawn according to the optimized pose data and The blue one is drawn according to the Lidar odometry data.



(a) Angular evaluation in yawl

(b) Angular evaluation in pitch

(c) Angular evaluation in roll

Fig. 10. Angular contrast graph.The red curve is drawn according to the high accuracy IMU data, which is regarded as the reference of the accuracy here. The green curve is drawn according to the optimized pose data and The blue one is drawn according to the Lidar odometry data.

and the accumulated errors grow with time going by. When the UGV returns to the start point, the accumulated angular errors comes to 9 degrees for the yawl, 0.7 degrees for the pitch and 2.2 degrees for the roll. the optimized angles are almost consistent with the measurement of a high accuracy IMU with a little delay in time.

## VII. Conclusions

In this paper, a real time SLAM system suitable for the large scale natural terrains has been applied. Abundant of information of the real driving situation is obtained due to the good performance of 3D mapping, which can provide a good basis for UGV's environment perception. Besides, the computation efficiency is guaranteed by organizing map data with the combination structure of the chain and the tree-based structures. The real time loop closure detection is carried out using both the Rotation Histogram Matching method and the BBS-based ICP algorithm. The accumulative errors are optimized on the basis of SPA where an IMU is used to add kinetics constraints. Through the 6D pose optimization, the accurate poses can be obtained when GPS signal is weak or even lost.

## References

[1] R. Mur-Artal, J. M. M. Montiel and J. D. Tards,"ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in *IEEE Transactions on Robotics,* vol. 31, no. 5, pp. 1147-1163, Oct. 2015.

[2] Stefan Leutenegger, Simon Lynen, Michael Bosse, *et al.*,"Keyframe-based visual-inertial odometry using nonlinear optimization," in *International Journal of Robotics Research,* vol. 34, no. 3, pp. 314-334, 2015.

[3] T. Gee, J. James, W. Van Der Mark, P. Delmas and G. Gimel'farb, "Lidar guided stereo simultaneous localization and mapping (SLAM) for UAV outdoor 3-D scene reconstruction," in *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1-6, 2016.

[4] R. A. Hewitt and J. A. Marshall,"Towards intensity-augmented SLAM with LiDAR and ToF sensors," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 1956-1961, 2015.

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, no. 2, pp. 239-256, Feb. 1992.

[6] K. M. Wurm *et al.*, "Hierarchies of octrees for efficient 3D mapping," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 4249-4255, 2011.

[7] K. Konolige, G. Grisetti, R. Kmmerle, W. Burgard, B. Limketkai and R. Vincent, "Efficient Sparse Pose Adjustment for 2D mapping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 22-29, 2010.

[8] F. Lu, and E. Milios, "Globally Consistent Range Scan Alignment for Environment Mapping." in *Autonomous Robots,* vol. 4, no. 4, pp. 333-349, 1997.

[9] M. Himstedt, J. Frost, S. Hellbach, H. J. Bhme and E. Maehle, "Large scale place recognition in 2D LIDAR scans using Geometrical Landmark Relations," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 5030-5035, 2014.

[10] A. Nchter, L. Kai, J. Hertzberg, *et al.*, "6D SLAM3D mapping outdoor environments," in *Journal of Field Robotics,* vol. 24, no. 8-9, pp. 699-722, 2007.

[11] D. M. Cole and P. M. Newman, "Using laser range data for 3D SLAM in outdoor environments," in *Proceedings 2006 IEEE International Conference on Robotics and Automation,* pp. 1556-1563, 2006.

[12] Z. Yi, C. Qi, and L. Yuan, "Building and Optimization of Kinect-based 3D Point Cloud Map," in *Semiconductor Optoelectronics*, 2016.

[13] M. Krainin, P. Henry, X. Ren, *et al.*, "Manipulator and Object Tracking for In Hand Model Acquisition", 2010.

[14] B. Douillard *et al.*, "Hybrid elevation maps: 3D surface models for segmentation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 1532-1538, 2010.

[15] K. M. Wurm, A. Hornung, M. Bennewitz, *et al.*, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. of the ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[16] G. Grisettiyz, C. Stachniss and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation,* pp. 2432-2437, 2005.

[17] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot, "Consistency of the EKF-SLAM Algorithm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3562-3568, 2006.

[18] R. Kmmerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation, Shanghai,* pp. 3607-3613, 2011.

[19] J. Strom and E. Olson, "Occupancy grid rasterization in large environments for teams of robots," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 4271-4276, 2011.

[20] K. Konolige, G. Grisetti, R. Kmmerle, W. Burgard, B. Limketkai and R. Vincent, "Efficient Sparse Pose Adjustment for 2D mapping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 22-29, 2010.

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

[22] J. Clausen, "Branch and bound algorithms-principles and examples," *Department of Computer Science, University of Copenhagen,* pp. 130, 1999.

[23] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2278-2283, 2005.

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