Proceeding of the IEEE
International Conference on Information and Automation
Yinchuan, China, August 2013

Mobile Robot 3D Map Building Based on RTM*

Ke Wang, Songmin Jia, Bing Guo and Yuchen Li
College of Electronic Information & Control Engineering
Beijing University of Technology
No. 100, Pingleyuan, Chaoyang District, Beijing, China
jsm@bjut.edu.cn

Abstract— **Under the structure of robot technology middleware(RTM),this paper presents a distributed method for mobile robot simultaneous localization and mapping(SLAM) to address the problem of 3D modeling in complex indoor environment.We integrate the image feature and depth information to establish the correspondence-based iterative closest point (ICP) algorithm for localizing the robot precisely. With the introduction of keyframe selection mechanism, a vision-based loop closure detect algorithm and tree-based network optimizer(TORO) are used to efficiently achieve globally consistent and accuracy maps during the map building. Experimental results verify the feasibility and effectiveness of the proposed algorithm in the indoor environment.**

Index Terms— **mobile robot; 3D map building; SLAM; keyframe; loop closure; RTM**

I. Introduction

Recently, the environment 3D reconstruction is becoming a hot area of research in the filed of mobile robotic, and has aroused great concern. G. Klein *et al.* recently presented a parallel tracking and mapping(PTAM) system for live modeling of the desktop scene[1]. In PTAM, it relied on a keyframe-based pose estimation with thousands of scene feature points, interleaved with repeated bundle adjustment in an optimised parallel implementation. It achieves highly accurate estimation for the camera localization with thousands of scene feature points, and the map is set up by using this sparse feature points. But it is hardly to present an intuition 3D describe of the environment. Combined with a high performance SLAM algorithm, Rachird A. Newcombe *et al.* used a state of the art multi-scale compactly supported radial basis function(MSCSRBF) technique for 3D initial map, and achieved real-time reconstruction in the desktop environment by using the high accuracy optical flow algorithm. However, two high performance GPUs were required in the algorithm, and it is difficult to meet the robot 3D map building in a complex indoor environment[2]. Andreas Nüchter proposed a linear solution method of Iterative Closest Point(ICP) algorithm for the mobile robot 3D map building by using a 3D laser[3]. Although the method could generate a accurate and adaptable map in the outdoor environment, the high development cost of the 3D laser sensor and textureless map can not suit the needs of colorful environment. V. Castaneda *et al.* constructed a 3D measurement model with a ToF(Time of Flight) camera and an RGB camera and an improve EKF SLAM for 3D map building[4]. In their method, they creatively combined the depth with RGB data , but the depth camera is too expensive to develop this hardware system.

In this paper, we present a fast method for mobile robot 3D SLAM to address the problem of 3D map building in indoor complex environment. In this method, the image features are detected and matched for data association, and a correspondence-based ICP algorithm is used for robot localization. With the keyframe selection mechanism, the keyframe is captured and current depth data is projected into the world coordinate for map building. To improve the precision of robot localization, the RANSAC(Random Sample Consensus) algorithm is used in the pose estimation. A vision-based loop detect method is used to observe the loop closure, and the loop error is minimized to achieve the global consistency and accuracy during the process of 3D modeling by using tree-based network optimizer(TORO)[5-6]algorithm. Finally, under the framework of Robot Technology Middileware(RTM)[7-9], a distributed mobile robot 3D SLAM system is developed to improve the efficiency of the system development.

The rest of paper consists of 4 section. Section 2 describes the structure of algorithm. Section 3 details the RTM and the developed RT components. The experimental results are given in section 4. And section 5 concludes the paper.

II. The Architecture of Algorithm

A. Preliminaries

In this paper, Kinect is used to capture the depth data and texture data in the environment. The calibration of the Kinect is an important step in this system, and we calibrate the Kinect by using the method which is presented in [10]. The calibration parameters include K_d, K_c and T. K_d and K_c are the intrinsic parameters matrix of the depth camera and color camera, and T is the extrinsic parameters matrix

* The research work is financially supported by Key Program of Beijing Natural Science Foundation (B), National Natural Science Foundation #61175087, National Natural Science Foundation #61105033 and Scientific Research Staring Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China.

between the depth camera and color camera.

$$K_d = \begin{bmatrix} f_{ud} & 0 & u_d \\ 0 & f_{vd} & v_d \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$K_c = \begin{bmatrix} f_u & 0 & u \\ 0 & f_v & v \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Where $(f_u, f_v), (f_{ud}, f_{vd})$ are the focal of RGB and depth camera, $(u, v), (u_d, v_d)$ are the center coordinate of these cameras.

According to the intrinsic parameters of depth camera, the corresponds between pixel coordinate x_d in the depth image and the 3D point p is established,which is also used for 3D map building. It is described as:

$$x_d = \pi(p) \tag{3}$$

Under the extrinsic parameters of depth camera and RGB camera, the corresponds between space point p and x in the RGB image also can be established, and which is described as

$$x = \Pi(p) \tag{4}$$

B. Feature Detection and Matching

FAST[11,12] corner algorithm is a cheap and fast feature point detect method. The feature point is decided by using the different value of pixel intensity between the candidate point and its neighborhood. it is described as:

$$N = \sum_{s \in circle(x)} |I(s) - I(x)| \tag{5}$$

where x is the candidate feature point, s is a point in the neighborhood of candidate feature, $I(x)$ is the pixel intensity value at point x. In this paper, we used FAST-10 corner detect method to get the image feature data.

To solve the problem of data association, we extracte the BRIEF[13] feature descriptor in the feature points to match the feature points. BRIEF is an efficient feature point descriptor which is using relatively few bits. The BRIEF feature descriptor is extracted by test on the patch of size $S \times S$, it is described as

$$f_{n_d} = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(x, x_i, y_i) \tag{6}$$

where $n_d = S \times S$ is the dimensional of BRIEF descriptor, τ is the different between point x_i and y_i, and $\tau(u, x_i, y_i) = \begin{cases} 1 & I(x_i) \leq I(y_i) \\ 0 & others \end{cases}$.

The matching of BRIEF descriptor can be evaluated using the Hamming distance. In this paper, a k-d tree-based nearest neighbor search algorithm is used for the BRIEF feature matching.

C. Pose Estimation

In this part, the RGB camera coordinate is used as the Kinect local coordinate, and the world coordinate is as the same as the fisrt frame of Kinect data. At the time i, robot's pose is described as

$$T^i_{cw} = \begin{bmatrix} R^i_{cw} & t^i_{cw} \end{bmatrix} \tag{7}$$

where "cw" is described as "from world coordinate to current", $T^i_{cw} \in SE(3)$, $SE(3)$ is a $Lie\ group$, and $SE(3) := \{[R, t] | R \in SO(3), t \in R^3\}$. According to the Lie algebra, T^i_{cw} can be expressed as

$$\begin{bmatrix} R^i_{cw} & t^i_{cw} \\ 0 & 1 \end{bmatrix} = exp(\hat{\zeta}) \tag{8}$$

where $\zeta = (\alpha, \beta, \gamma, t_x, t_y, t_z)$, $\alpha, \beta, \gamma, t_x, t_y, t_z$ are the rotation and transform in the direction of x, y, z-axis. $\hat{\zeta}$ is defined as

$$\hat{\zeta} = \begin{bmatrix} 0 & \gamma & -\beta & t_x \\ -\gamma & 0 & \alpha & t_y \\ \beta & -\alpha & 0 & t_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

On the basis of the robot's pose T^i_{cw}, the relationship of 3D point between world coordinate and local coordinate is established, and it is written as

$$p_c = T^i_{cw} p_w \tag{10}$$

In the pose estimation algorithm, $M(x)$ is defined to handle the invalid measurement data in the depth image, when the value is invalid in point x, $M(x) = 0$, otherwise $M(x) = 1$.

To effectively achieve the current robot pose, ICP algorithm model[14] is established by using the relationship of current and world coordinate, it is described as

$$E = \min_{T^k_{cw}} \sum_{p^k \in \omega} \left\| T^k_{cw} p_w - p^k \right\| \tag{11}$$
$$\omega = \{p^k | x = \pi(p^k), M(x) = 1\}$$

where p_w is the point in the world coordinate, p^k is the new captured point.ω is the valid data set in the local coordinate.

During the interval of the time $k-1$ and time k, the increment of robot's position is assumed as T^k_{inc}, the increment of the rotation and the transformation are $r_{inc} = (\triangle\alpha, \triangle\beta, \triangle\gamma)$ and $t_{inc} = (\triangle t_x, \triangle t_y, \triangle t_z)$ respectively. The increment formula of robot's pose can be described as

$$T^k_{cw} = T^k_{inc} T^{k-1}_{cw} \tag{12}$$

$$T^k_{cw} = \exp(\hat{\zeta}) \tag{13}$$

$$\zeta = (r_{inc}, t_{inc}) \tag{14}$$

When the increment of rotation is satisfied to the hypothesis of small angle assumption, T_{inc}^{k} is expressed as

$$\begin{aligned} T_{inc}^{k} &= \left[\begin{array}{ccc|c} 1 & \Delta\gamma & -\Delta\beta & \Delta t_x \\ -\Delta\gamma & 1 & \Delta\alpha & \Delta t_y \\ \Delta\beta & -\Delta\alpha & 1 & \Delta t_z \end{array} \right] \\ &= \left[\begin{array}{c|c} R_{inc} & t_{inc} \end{array} \right] \\ &= \left[\begin{array}{c|c} [r_{inc}]_{\times} & t_{inc} \end{array} \right] \end{aligned} \tag{15}$$

where $[r_{inc}]_{\times}$ is the skew-symmetric matrix.

By using mathematical transform, the ICP model can be written as

$$\begin{aligned} &\min_{\xi \in se(3)} \sum_{\omega} \left\| F(p_w^{k-1})\xi + p_w^{k-1} - p^k \right\| \\ &\omega = \{p^k | x = \pi(p^k), M(x) = 1\} \end{aligned} \tag{16}$$

where p_w^{k-1} is the point in the local coordinate of the time k-1.

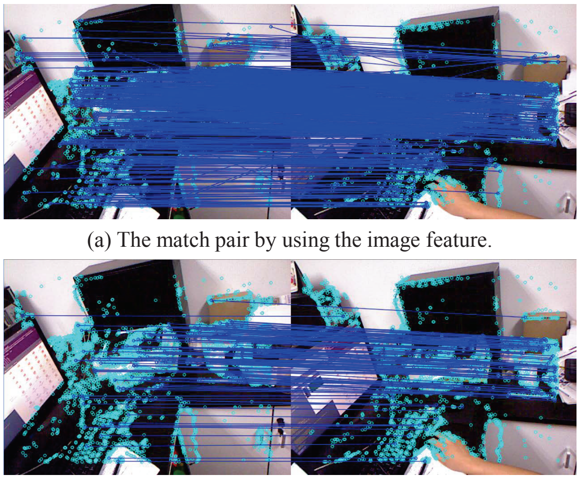
In this paper, a method combining the image feature matching and stand ICP algorithm are used to achieve the robot precision pose estimation. The image feature is used to establish the correspondence of the 3D point, and then the correspondence-based ICP algorithm model is solved to achieve the current robot's pose. Finally, the ICP algorithm is described as

$$\begin{aligned} &\min_{x \in se(3)} \sum_{\Omega} \left\| F(p_w^{k-1})\xi + p_w^{k-1} - p^k \right\| \\ &\Omega = \{(p_w^{k-1}, p^k) | x^k = \Pi(p^k), x^{k-1} = \Pi(p_w^{k-1}) \\ &\qquad x^k \ and \ x^{k-1} is \ matched \ pair\} \end{aligned} \tag{17}$$

Algorithm 1 $Pose\_estimation(I_1, I_2)$

```
$F = Extract\_RGB\_feature(I1)$
$F_t = Extract\_RGB\_feature(I2)$
for $i = 0$ $to$ $max\_loop$ do
    $A_f = RANSAC\_SAMPLE(F, F_t, n)$
    $x = \arg\min_{x} \sum_{A_f} \|F(p_w^{k-1}x + p_w^{k-1} - p^k\|$
    if $E(x) \leq max\_error$ then
        $A_{inner} = Inner(x, F, F_t)$
        $x_{inner} = \arg\min_{x} \sum_{A_f} \|F(p_w^{k-1}x + p_w^{k-1} - p^k\|$
        if $E(x_{inner}) \leq max\_inner\_error$ then
            return true
        end if
    end if
end for
return false
```

To effectively achieve the pose estimation, the RANSAC algorithm is used to eliminate the mismatched features and improve the robustness and accuracy of the pose estimation. At beginning of the pose estimation, n matching pairs are arbitrarily selected as the input data of our ICP algorithm, and the best estimation is solved by using Levenberg-Marquardt algorithm. And when the value of (17) is satisfied to the threshold, the total matching pairs are tested to distinct the interior points and outer points. Then a new estimation of ζ is solved by using all the interior points, and when the value of (17) is satisfied to the interior threshold, the pose estimation is succeeded. The overall execution process is shown in Algorithm 1. Fig.1 (a) shows the match pairs by using the image feature, and Fig. 1(b) shows the match pairs after the pose estimation. Fig. 1 shows that Algorithm 1 can effectively eliminate the mismatch pairs and also verifies the efficiency of Algorithm 1.



(a) The match pair by using the image feature.

(b) The match pair after the pose estimation

Fig. 1. The elimination of mismatch pairs.

D. Keyframe Selection

With the current pose and depth data, an accurate 3D Map can be constructed. Considering that there are about 300,000 points in every frame, it is necessary to create a more concise representation of the 3D map. In this system, a keyframe selection mechanism is introduced, and the map building procedure will start only when a new keyframe is captured. However, the accuracy of pose estimation is depended on the overlap between the input data. Therefore, a distance threshold parameter should take into count in our system for determining the keyframes.

The overall execution of keyframe selection is shown in Algorithm 2. In the Algorithm 2, I_k^{kf} is the kth keyframe, I_i is the frame in the sequence between the kth and $k+1$th keyframe. In the procedure of keyframe selection, if the pose estimation between current frame I_i and current keyframe I_k^{kf} is failed or the distance is exceeded to the threshold, the frame I_{i-1} will be capture as keyframe. Obviously, There is a dense association between keyframes is effectively established to ensured the accuracy of pose estimation by using this keyframe selection mechanism.

Algorithm 2 $Keyframe\_selection(I_{i-1}, I_i, I_k^{kf})$

```
if $distance(I_{i-1}, I_{kf} \leq max\_distance$ then
```

```
$pre_est = Pose\_estimation(I_{i-1}, I_k^{kf})$
if $pre\_est$ is true then
    if $distance(I_{i-1}, I_{kf} \leq max\_distance$ then
        $curr\_est = Pose\_estimation(I_i, I_k^{kf})$
        if $curr\_set$ is false then
            $add\_keyframe(I_{i-1})$
        end if
    else
        $add\_keyframe(I_{i-1})$
    end if
end if
```

Algorithm 3 $loop\_detection(I_k^{kf}, I^{kf})$

```
for $I_j^{kf}$ in $I^{kf}$ do
    $d = distance(I_k^{kf}, I_i^{kf})$
    $c = angle(I_k^{kf}, I_i^{kf})$
    if $d \leq max\_distance$ and $c \leq min\_angle$ and $abs(i-k) > min\_frame\_count$ then
        if $Pose\_estimation(I_i^{kf}, I_k^{kf})$ then
            $Optimize\ the\ map$
        end if
    end if
end for
```

E. Loop Detection and Optimization

During the map building, the quality of global map is largely dependent on the accuracy of the pose estimation. However, with the cumulation of the pose estimation error, a drift is arisen in the 3D map. To effectively achieve the global optimization of the 3D map, a vision-based loop detect method is used in the system to determine the closed-loop data when a new keyframe is captured. During the loop detect, a relative pose between the current keyframe and the previous keyframes is estimated by using the pose estimation algorithm. When the relative pose estimation is succeeded, it means that close loop is detected and the TORO optimization algorithm should start immediately. The overall execution of loop detection is shown in Algorithm 3. In Algorithm 3, the function $angle(I_k^{kf}, I_i^{kf})$ is used to get the orientation angle of camera between the kth and ith keyframe. In the TORO algorithm, the robot's poses are considered as the end points, the relative positions are the edges, and all of those end points and edges constitute a relation tree of robot's pose. In this tree structure, the minimization of the error function of the observed and calculated relative pose is used to achieve the optimization of current tree structure. The error function is described as

$$T_{cw} = \arg\min_{T_{cw}} F(T_{cw}) \tag{18}$$

$$F(T_{cw} = \sum_{(i,j)\in G} e(T_{cw}^i, T_{cw}^j, T_{ij})\Lambda e(T_{cw}^i, T_{cw}^j, T_{ij}) \tag{19}$$

where $T_{cw} = \{T_{cw}^1, T_{cw}^2, ..., T_{cw}^n\}$ is the vector descriptor of robot's poses, T_{ij} is the observed relative pose between time i and j, Λ is the relative matrix among the robot's poses, G is the relative tree. $e(T_{cw}^i, T_{cw}^j, T_{ij})$ is the error function measures how well the poses T_{cw}^i and T_{cw}^j satisfy the observe constraint T_{ij}. It is 0 when T_{cw}^i and T_{cw}^j perfectly matches the constraint.

III. Distributed Processing

RTM[7-9] was developed by Agency of Industrial Science and Technology of Japan (AIST) to promote application of Robot Technology (RT) in various fields. RTM aims to modularize robotic functional elements as RT software components which enables the system to be extended and integrated easier for new applications. In order enable system to be language independence, operating system independence, RTM has been developed based on CORBA (Common Object Request Broker Architecture). CORBA uses an Object Request Broker as the middleware that establishes a client-server relationship between objects, and it is an object-oriented extension of Remote Procedure Calls (RPCs). CORBA uses GIOPs (General Inter-ORB Protocols), ESIOPs (Environment Specific Inter-ORB Protocols) and IIOP (Internet Inter-ORB Protocols) to implement a truly heterogeneous distributed system. This heterogeneity enables CORBA to inter-operate ORBs purchased from different vendors and supported on different platforms. In the distributed system, the fundamental principles conclude the distributed module

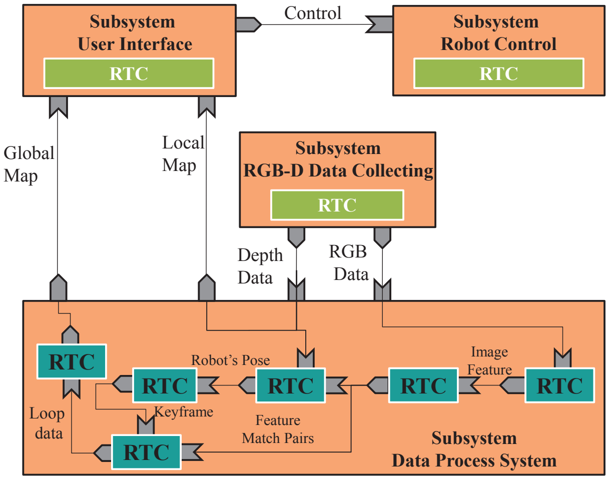
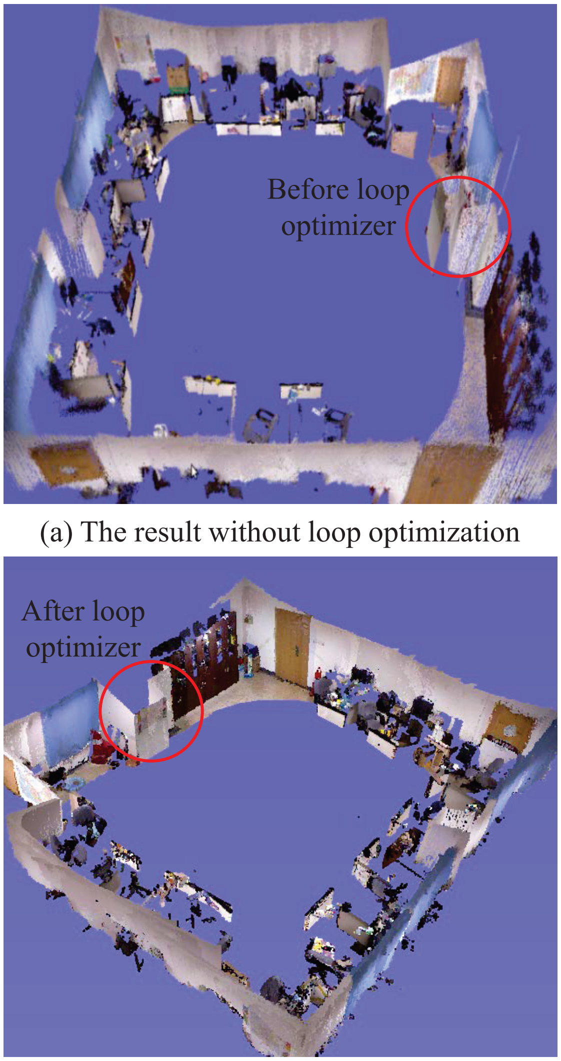


Fig. 2. The distributed structure of map building system.

has the smallest functional, strong independence, and the minimum consumption between the modules. Consider that principles, our system is divided to 9 modules, and there are robot control, data acquisition, image feature extraction, image feature matching, pose estimation, keyframe selection, loop detect, loop optimization and GUI. The distributed system structure shown in Fig. 2. It includes:

A. Mobile robot control component: control function classes include:
   - Inport <Vector2D>: control command for mobile robot.
B. Kinect data gettiong component: get raw data from the Kinect and algin the depth and image:
   - Outport <TimedLongSeq>: output the depth data.
   - Outport <TimedOcterSeq>: output the RGB image data.
C. Image feature extraction component: get raw data from the Kinect and algin the depth and image:
   - Inport <TimedOcterSeq>: input the RGB image data.
   - Outport <TimedfloatSeq>: output the image feature.
D. Image feature matching component: match the features:
   - Inport <TimedfloatSeq>: input the image feature.
   - Outport <TimedLongSeq>: output the match pairs.
E. pose estimation component: estimate the robot's pose:
   - Inport <TimedLongSeq>: input the image match pairs.
   - Inport <TimedLongSeq>: input the depth data.
   - Outport <TimedfloatSeq>: output the robot's pose.
F. loop detect component: detect the loop close:
   - Inport <TimedfloatSeq>: input the robot's pose.
   - Inport <TimedfloatSeq>: input the image match pairs.
   - Outport <TimedfloatSeq>: output the relative pose.
G. loop optimization component: optimization the :
   - Inport <TimedfloatSeq>: Input the relative pose.
   - Outport <TimedfloatSeq>: output the optimization of robot pose.
H. Map building component: optimization the :
   - Inport <TimedfloatSeq>: Input the robot pose.
   - Outport <TimedfloatSeq>: output the local map.
I. GUI component: show the 3D map:
   - Inport <TimedfloatSeq>: Input the local map.
   - Outport <Vector2D>: control command for mobile robot.



(a) The result without loop optimization

(b) The result with loop optimization

Fig. 3. The result of map building.

IV. Experimental Result

To verify the effectiveness of the algorithm, the experiment is taken in the indoor environment. The experimental platform consists of an American Mobile Robots Inc. Pioneer3-DX mobile robot and a Mircosoft Kinect. The P3-DX is available with an embedded PC, and can be used as an intelligent mobile platform. We use Kinect for obtaining environment data. This fantastic sensor offers USB interface to provide the depth and color information. The maximum field of view for this detector is 52° and the image resolution is 640 × 480 with a refresh frequency of up to 30Hz.

During the experiment, the robot is instructed to traverse around the work table centered in our laboratory which covers a planar area of 16×12 m^2. Translational speed is about 0.2 m/s. While the depth and color data is captured by a Kinect which is mount on the robot, the robot pose is estimated by using the proposed algorithm. When a keyframe is captured, the procedure of mapping will start to update the global map.

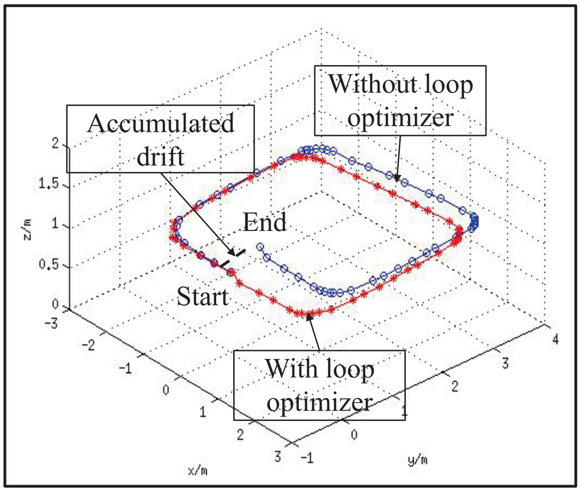


Fig. 4. The result of robot pose curve.

Because of the accumulated error of pose estimation, a drift is arisen in this map which is hardly to be closed, as red circle which is shown in the Fig. 3(a). In the Fig. 4, the accumulated drift also can be obviously seen in the blue pose curve. To efficiently eliminate the drift, the loop detect and optimized procedure is introduced in this system when a new keyframe is captured. With this procedure, we can achieve a global consistency map and improve the quality of generated map efficiently, as it is verified in the Fig. 3(b). In the Fig. 4, the red pose curve is shown the effect of optimization. All results demonstrate that the generated 3D map is largely reflected the real environment scene, and also verify the feasibility and the efficiency of the proposed algorithm.

V. Conclusion

This paper systematically studied the problem of robot 3D SLAM in indoor complex environment. Under the structure of robot technology middleware, a distributed system is developed to achieve the 3D mapping. In the system, the image feature match is used to established correspondence of 3D points, and then a correspondence-based ICP model is solved to localize the robot precisely. On this basis, the keyframe is chosen by using a keyframe selection mechanism, and the drift error of 3D map is optimized by using the TORO algorithm when a loop closure is detected by using a vision-based loop detect method. The indoor environment experiment results verify the feasibility and effectiveness of the proposed method. In the future, a improvement algorithm of pose estimation and a application of multi-robot map will be the research priorities.

References

[1] G. Klein, D. Murray. ”Parallel tracking and mapping for small AR workspaces,” Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality , 2007: 225-234.

[2] R. A. Newcombe, A. J. Davison. ”Live dense reconstruction with a single moving camera,”In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2010

[3] Andreas Ňuchter , Jan Elseberg, Peter Schneider, Dietrich Paulus.”Study of parameterizations for the rigid body transformations of the scan registration problem,” Computer Vision and Image Understanding,2010.

[4] V. Castaneda, D. Mateus, N. Navab. ”SLAM combining ToF and high-resolution cameras,” In IEEE Workshop on Motion and Video Computing, 2011.

[5] Grisetti Giorgio, Slawomir Grzonka, Cyrill Stachniss, Patrick Pfaff, Wolfram Burgard. ”Efficient Estimation of Accurate Maximum Likelihood Maps in 3D,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2007

[6] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, Wolfram Burgard. ”A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent,” Robotics: Science and Systems (RSS), 2007

[7] Songmin Jia, Murakami T, Chugo D, et al. ”Interactive robot system for supporting object acquisition based on Robot Technology Middleware,”Proceedings of the 2008 IEEE International Conference on Information and Automation. Piscataway, NJ, USA: IEEE, 2008: 966-971.

[8] N Ando, T Suehiro, K Kitagaki, et al. ”Implementation of RT composite components and a component manager”The 22nd Annual conference of the Robotic Society of Japan. IC26, 2004.

[9] Songmin Jia, Ke Wang, Xiuzhi Li, et al. ”Map building for mobile robot based on distributed control technology,”Proceedings 2011 International Conference on Information and Automation (ICIA 2011). Piscataway, NJ, USA: IEEE, 2011: 279-284.

[10] Jan Smisek, Michal Jancosek and Tomas Pajdla. ”3D with Kinect,” 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). Prague, 2011, 1154-1160.

[11] Edward Rosten, Tom Drummond. ”Machine learning for high-speed corner detection,” Proceedings of European Conference on Computer Vision, 2006: 430-443.

[12] Edward Rosten, Reid Porter, Tom Drummond. ”FASTER and better: A machine learning approach to corner detection,”IEEE Trans. Pattern Analysis and Machine Intelligence. 2010: 105-119.

[13] M. Calonder, V. Lepetit, C. Strecha, P. Fua. ”BRIEF: Binary Robust Independent Elementary Features,”Proceedings of the European Conference on Computer Vision. 2010: 322-328.

[14] P. J. Besl, H. D. McKay. ”A method for registration of 3-d shapes,” Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011: 127-136.