A Real-Time Robust Global Localization for Autonomous Mobile Robots in Large Environments

Jianping Xie, Fawzi Nashashibi, Michel Null Parent, Olivier Garcia-Favrot

A Real-Time Robust Global Localization for Autonomous Mobile Robots in Large Environments

Jianping Xie, Fawzi Nashashibi, and Michel Parent
IMARA Team
INRIA Rocquencourt
Paris, France
First Name.Last Name@inria.fr

Olivier Garcia Favrot
Research and Development department
INDUCT
Paris, France
garciafavrot@hotmail.com

Abstract—**Global localization aims to estimate a robot's pose in a learned map without any prior knowledge of its initial pose. Achieving highly accurate global localization remains a challenge for autonomous mobile robots especially in large-scale unstructured outdoor environments. This paper introduces a real-time reliable global localization approach with the capability of addressing the kidnapped robot problem using only laser sensors. Our approach includes four steps: 1) local Simultaneous Localization and Mapping 2) map matching 3) position tracking and 4) localization quality evaluation. For sensor perception, we use occupancy grid method to represent robot environment. A novel pyramid grid-map based coarse-to-fine matching approach is proposed to improve the localization accuracy. Experimental results including an outdoor environment of 25,000 m^2 are presented to validate the feasibility and reliability of the proposed approach.**

Keywords—**global localization, SLAM, coarse-to-fine, mobile robots**

I. INTRODUCTION

Mobile robot localization deals with determining a robot's pose relative to its surrounding environment. It is considered as the fundamental problem in autonomous mobile robotics [1], [2]. Acquiring precise localization is of essential importance for accomplishing a given mission. The localization problems can be categorized into three levels: position tracking, global localization and the kidnapped robot problem [1], [3], [4].

Position tracking assumes that the initial robot pose is given and the robot's pose can be achieved by compensating pose errors and tracking robot over time in a known environment. Addressing the global localization problem is more challenging as it seeks to localize itself in a studied map with an unknown initial pose. The kidnapped robot problem is that well-localized robot gets kidnapped and teleported to some other location without being informed. It is even more difficult than the global localization, in that the robot might still naively believe that it is on the right track while it does not. Also, being capable of recovering from the localization failure is critical for truly autonomous robots since most state-of-the-art localization algorithms can not be guaranteed never to fail [3].

In this paper, we present a high-efficient robust global localization approach together with the ability of recovering from "kidnapping" in complex outdoor environments using

exclusively laser data. In our approach, the robot simultaneously determines its location and builds a SLAM map based on a proposed local SLAM (Simultaneous Localization and Mapping) algorithm. Meanwhile, a local map which differs from the SLAM map is also created for map matching purpose. Once the robot gains sufficient evidence for localization distinction, a map matching algorithm is carried out to search the initial pose by matching the local map to a global reference map. After that, we can update the robot pose based on robot initial pose and relative movement knowledge from local SLAM. Then, global localization is actually simplified as a position tracking problem. Besides, a novel pyramid grid-map based coarse-to-fine matching method is used to further improve localization accuracy. In order to handle the kidnapped robot problem, a localization quality evaluation algorithm is also integrated into the whole localization system.

The rest of the paper is organized as follows: In Section II, we review the localization problem formulation and briefly introduce the architecture of our localization system. Section III further discusses the localization and matching approaches in detail. Real experimental results in unstructured outdoor environments are given in Section IV. Finally, Section V presents the conclusion and future work.

II. LOCALIZATION PROBLEM AND SOLUTION

A. Localization Problem

In general probabilistic terms, the global localization problem is considered as a posterior probability density optimization problem conditioned on all available history data.

$$\hat{x}_t^{MAP} = \arg\max_x \{P(x_t \mid Y_t)\} \qquad (1)$$

where x_t is the robot pose at time t. All available data Y_t are divided into two types: perceptual data $z_{1:t} = \{z_1, z_2,..., z_t\}$ and control data $u_{1:t} = \{u_1, u_2,..., u_t\}$. x_0 stands for the initial pose. By applying the Markov assumption, (1) can be rewritten as

$$\hat{x}_t^{MAP} = \arg\max_x \prod_{i=1}^{t} P(z_i \mid x_i) \prod_{i=1}^{t} P(x_i \mid x_{i-1}, u_{i-1}) P(x_0) \quad (2)$$

B. Localization Solution

To address the global localization problem, we design and develop a novel architecture for our localization system. As
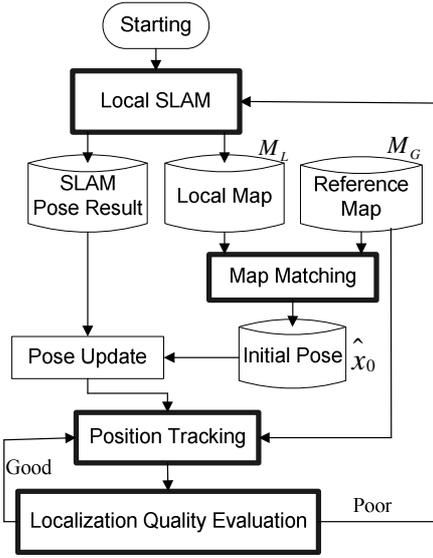
Figure 1.   Architecture of the Localization System

shown in Fig. 1, the localization system includes four main parts: local SLAM, map matching, position tracking and localization quality evaluation.

III.   ROBOT LOCALIZATION

A.   Local SLAM

Local SLAM algorithm enables a mobile robot to build an accurate SLAM map M of an unknown environment while concurrently using this map to recover its pose x_t. The SLAM algorithm calculates the joint posterior probability over all past observations and controls [3],[5] as follows:

$$P(x_t, M \mid z_{1:t}, u_{1:t}, x_0) \tag{3}$$

The joint posterior (3) can be obtained through a recursive procedure of (4) and (5).

$$
\begin{aligned}
&P(x_t, M \mid z_{1:t-1}, u_{1:t}, x_0) \\
&= \int P(x_t \mid x_{t-1}, u_t) \times P(x_{t-1}, M \mid z_{1:t-1}, u_{1:t-1}, x_0) dx_{t-1}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
&P(x_t, M \mid z_{1:t}, u_{1:t}, x_0) \\
&= \frac{P(z_t \mid x_t, M) P(x_t, M \mid z_{1:t-1}, u_{1:t}, x_0)}{P(z_t \mid z_{1:t-1}, u_{1:t})}
\end{aligned}
\tag{5}
$$

Where $P(x_t \mid x_{t-1}, u_t)$ is the motion model which denotes the transition of robot state. $P(z_t \mid x_t, M_L)$ is called the sensor model which describes the probability of observing z_t when x_t and M are already known. In this paper, a motion prediction model and a grid-map based maximum likelihood mapping framework are presented to tackle the local SLAM problem. Also, a pyramid grid-map based coarse-to-fine refining strategy is proposed to improve localization precision.

1) Motion Prediction: To model robot motion, we simply assume that the relative movement at time t equals the one at time t-1: $\Delta x_t = \Delta x_{t-1}$. This is because of the continuity property of the robot movements. As in most cases, the relative movements will not suddenly suffer from an enormous change, particularly when the system update rate is high. We also have $v_t = v_{t-1}$, $w_t = w_{t-1}$, where v_t is the translational velocity and w_t is the rotational velocity. The rough pose estimation is given by

$$x_t^* = x_{t-1} + \Delta x_t = x_{t-1} + \Delta x_{t-1} \tag{6}$$

In order to avoid confusion, the pose state x is denoted by vector $(xp, yp, \theta p)^{\mathrm{T}}$. Then the motion model is presented as

$$
\begin{aligned}
&P(x_t \mid x_{t-1}, u_t) \\
&= \eta' P(xp_t - xp_t^*) P(yp_t - yp_t^*) P(\theta p_t - \theta p_t^*)
\end{aligned}
\tag{7}
$$

Where η' is a normalization constant and

$$P(xp_t - xp_t^*) = \frac{1}{\sqrt{2\pi\delta_{xp}^2}} e^{-\frac{(xp_t - xp_t^*)^2}{2\delta_{xp}^2}} \tag{8}$$

$$P(yp_t - yp_t^*) = \frac{1}{\sqrt{2\pi\delta_{yp}^2}} e^{-\frac{(yp_t - yp_t^*)^2}{2\delta_{yp}^2}} \tag{9}$$

$$P(\theta p_t - \theta p_t^*) = \frac{1}{\sqrt{2\pi\delta_{\theta p}^2}} e^{-\frac{(\theta p_t - \theta p_t^*)^2}{2\delta_{\theta p}^2}} \tag{10}$$

The variances δ_{xp}, δ_{yp}, and $\delta_{\theta p}$ are given by $\delta_{xp} = \lambda_1 v$, $\delta_{yp} = \lambda_2 v$, and $\delta_{\theta p} = \lambda_3 w$. λ_i ($i = 1, 2, 3$) is just a parameter. Moreover, a candidate pose space $x_t' = \{x_{t,1}', ..., x_{t,k}'\}$ is generated by sampling $P(x_t \mid x_{t-1}, u_t)$ for later pose refining process. The correspondent relative movement Δx will finally be updated when the robot pose is refined.

2) Perception Representation: For the SLAM map, we use the occupancy grid method to represent the environment. In this representation, the robot environment is discretized into two-dimensional square cells and each cell is associated with a value in [0, 1] indicating the probability of a cell being occupied. The higher value of a grid cell is, the darker a grid cell is and more likely it is being occupied.

Assuming each grid cell is independent and poses $x_{0:t}$ are known. Given observations $z_{1:t}$, the posterior probability $P(m \mid x_{0:t}, z_{1:t})$ for a grid cell m is determined by using Bayes theorem:

$$P(m \mid x_{0:t}, z_{1:t}) = \frac{P(z_t \mid x_{0:t}, z_{1:t-1}, m) P(m \mid x_{0:t}, z_{1:t-1})}{P(z_t \mid x_{0:t}, z_{1:t-1})} \tag{11}$$

Assuming z_t is independent from $x_{0:t-1}$ and $z_{0:t-1}$ at given m:

$$P(m \mid x_{0:t}, z_{1:t}) = \frac{P(m \mid x_t, z_t) P(z_t \mid x_t) P(m \mid x_{0:t}, z_{1:t-1})}{P(m) P(z_t \mid x_{0:t}, z_{1:t-1})} \tag{12}$$

Also, for the opposite event:

$$P(\overline{m}\,|\,x_{0:t},z_{1:t}) = \frac{P(\overline{m}\,|\,x_t,z_t)P(z_t\,|\,x_t)P(\overline{m}\,|\,x_{0:t},z_{1:t-1})}{P(\overline{m})P(z_t\,|\,x_{0:t},z_{1:t-1})} \quad (13)$$

Dividing (12) by (13) leads to the cancellation of various difficult-to-calculate probabilities:

$$\frac{P(m\,|\,x_{0:t},z_{1:t})}{P(\overline{m}\,|\,x_{0:t},z_{1:t})} = \frac{P(m\,|\,x_t,z_t)P(\overline{m})P(m\,|\,x_{0:t},z_{1:t-1})}{P(\overline{m}\,|\,x_t,z_t)P(m)P(\overline{m}\,|\,x_{0:t},z_{1:t-1})} \quad (14)$$

Implementing log odds ratio for (14) brings an elegant recursive formula in log-odds term:

$$\begin{aligned}
&\log(Odds(m\,|\,x_{0:t},z_{1:t})) \\
&= \log(Odds(m\,|\,x_t,z_t)) - \log(Odds(m)) \quad (15)\\
&\quad + \log(Odds(m\,|\,x_{0:t-1},z_{1:t-1}))
\end{aligned}$$

Where,

$$\log(Odds(A)) = \log(\frac{P(A)}{P(\overline{A})}) = \log(\frac{P(A)}{1-P(A)}) \quad (16)$$

In (15), $P(m)$ is the prior occupancy probability of a grid cell. The remaining probability $P(m\,|\,x_t, z_t)$ stands for the inverse sensor model. We adopt a similar model according to [6]. Posterior probability $P(m\,|\,z_{1:t}, x_{0:t})$ can be recovered easily since the $Odds(m\,|\,z_{1:t}, x_{0:t})$ can be computed recursively.

3) Scan Matching: Building a consistent environment map requires knowing the robot's precise localization. However, the pose estimation from motion prediction is inaccurate. Therefore, we introduce a grid-based scan matching approach to refine the robot's pose in the maximum likelihood mapping framework. We correct the robot pose by comparing the current laser scan with the existing SLAM map. The robot's pose at time t is determined as follow [3], [6], [7]:

$$\hat{x}_t = \arg\max_{x_t}\{P(z_t\,|\,x_t, M_{t-1})P(x_t\,|\,\hat{x}_{t-1}, u_t)\} \quad (17)$$

In (17), $P(x_t\,|\,\hat{x}_{t-1}, u_t)$ is the motion model which is already discussed in the previous description. To calculate $P(z_t\,|\,x_t, M_{t-1})$, we apply a high-efficient method according to [6]. It only focuses on the grid cells which are hit by the laser end-points. We denote current laser scan by $z_t = \{z_t^1, z_t^2, ..., z_t^N\}$ which contains N individual measurements (N corresponding laser beams). Each $z_t^K (K = 1,..., N\}$ is projected into the existing SLAM map in global coordinate reference. The grid cell which is hit by the laser end-point z_t^K is denoted by $m_t^{hit(K)}$. If cell $m_t^{hit(K)}$ is being occupied, the occupancy value of this cell will be added into a voting score. The likelihood of the sensor measurement is represented by the final voting score:

$$P(z_t\,|\,x_t, M_{t-1}) \propto \sum_{k=1}^{N}\{P(m_{t-1}^{hit(K)}), m_{t-1}^{hit(K)} \text{ is occupied}\} \quad (18)$$

The maximizing operation in (17) is implemented by an extensive search over candidate pose space. The most likely robot pose is recovered when the current laser scan correctly aligns with the existing SLAM map. After that, we update SLAM map $M_t = M_{t-1} \cup \{z_t, x_t\}$ according to (15) and (16). Finally, the robot simultaneously improves its localization and SLAM map by alternating the pose refining and map updating.

4) Further Pose Refining: To further improve localization accuracy, we propose a novel pyramid grid-map based coarse-to-fine refining strategy: As Fig. 2 shows, Firstly, the scan matching process runs in the low-resolution grid map M_t^A (20×20cm²/pixel). The rough pose estimation $x_t^{*(1)}$ and candidate pose space $x_t^{'(1)}$ are obtained respectively from the motion model and the sampling algorithm. By performing the maximum likelihood mapping algorithm, $x_t^{(1)}$ is found as the refined pose among the candidate pose space. Furthermore, we enter a higher resolution map M_t^B (10×10cm²/pixel) and the same matching procedures are carried on in the second-level map starting as $x_t^{*(2)} = x_t^{(1)}$. It indicates that we regard the previous refined pose as the raw estimation in the current level map. Once the matching process is finished, again, we come into a higher resolution level to continue our matching process. Finally the best robot pose is recovered as $x_t^{(3)}$ in the map M_t^C (5×5cm²/pixel). Note that the pose candidate space area is reduced during the refining process. To avoid unexpected mismatching, we check the pose corrections and set the restriction as:

$$|\triangle x_t^{(j)}| > |\triangle x_t^{(k)}|\ (if\ j < k) \quad (19)$$

Where $\triangle x_t^{*(i)} = x_t^{(i)} - x_t^{*(i)}$ ($i = 1, 2, 3$) denotes the pose correction. The matching process will stop if correction relationship goes against (19), for instance: $|\triangle x_t^{(2)}| < |\triangle x_t^{(3)}|$, the matching process ends and $x_t^{(2)}$ returns as the final result.
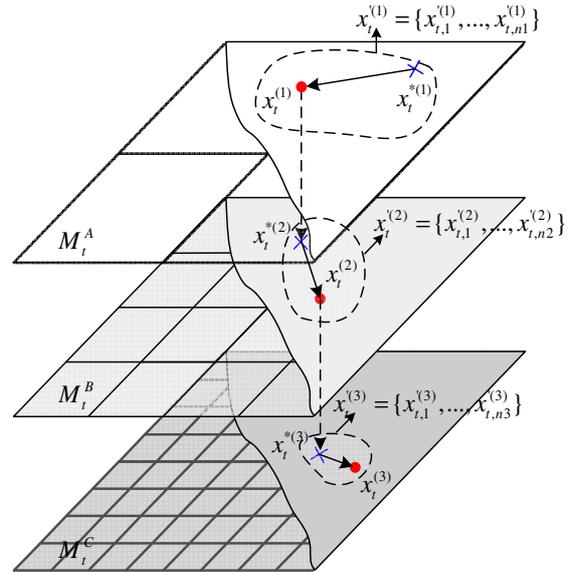

Figure 2. Pyramid grid-map based coarse-to-fine refining strategy

It's worth to notice that the initial pose x_0 is unknown for the moment. For implementing local SLAM algorithm, we assume the initial pose as $x_0 = (0, 0, 90°)^T$ and it will be corrected in the following discussions.

B. Map Matching

Laser-based global localization problem is similar to a recognition problem as the robot seeks to match the current local map with a global reference map. Before going into any detail, we first introduce the map presentation for map matching algorithm. Similar to the SLAM map, all maps are still presented using the occupancy grid approach and laser data are projected into two-dimensional square cells. However, instead of calculating the occupancy possibility, we regard a cell is empty until it is being hit more than twice.

In our method, we increasingly build the local map M_L based on laser data and localization results from local SLAM. The global reference map is built by a robust laser-based SLAM algorithm which is capable of creating a consistent map in large-scale outdoor environments [8]. In fact, we load all global reference data (reference map, reference laser scans and corresponding robot poses) into the localization system at the beginning. When the local map gains sufficient evidence, we carry out the map matching algorithm to search robot's initial pose. The map matching algorithm contains two steps:

Step1: We generate a set of local maps by rotating the local laser map M_L every $4°$ from $4°$ to $360°$ ($\theta = 4°,..., 360°$). The local map set is presented as

$$\{M_L^*\} = \{M_{L,1}^*, M_{L,2}^* ..., M_{L,90}^*\} \quad (20)$$

Assuming each local map $M_{L,k}^* (k = 1,...,90)$ contains n small grid cells:

$$M_{L,k}^* = \{m_{L,k}^{*(1)}, m_{L,k}^{*(2)},...m_{L,k}^{*(n)}\} \quad (21)$$

Where $m_{L,k}^{*(j)}$ is a small grid cell in $M_{L,k}^*$.

Step 2: We create position samples from robot reference poses. All the position samples are considered as the robot's possible initial position. As Fig. 3 shows, we denote all potential candidate positions (marked as circle) by the set $XC = \{XC_0, ..., XC_s\} = \{(XC_{x0}, XC_{y0}), ..., (XC_{xs}, XC_{ys})\}$. For map matching purpose, we create submap $M_{G,i}^*$ centered in XC_i based on global reference data:

$$M_{G,i}^* = \{m_{G,i}^{*(1)}, m_{G,i}^{*(2)},...m_{G,i}^{*(n)}\} \quad (i = 1,...s) \quad (22)$$

Where $m_{G,i}^{*(j)}$ is a grid cell in $M_{G,i}^*$. Apparently, both $M_{G,i}^*$ and $M_{L,i}^*$ have the same map size.

To quickly locate robot initial position area and speed up the whole matching process, the map matching firstly takes place in a low-resolution level ($50 \times 50 cm^2$/pixel): each local map $M_{L,k}^*$ seeks to match with submap $M_{G,i}^*$ ($i = 1,...,s$) one by one. The refined initial robot pose is obtained when $M_{L,k}^*$ and $M_{G,i}^*$ achieve the best matching:
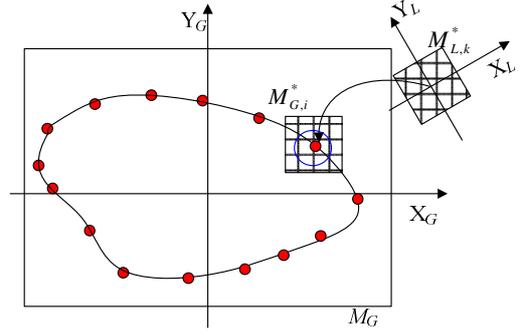

Figure 3. Matching process between local map and reference submap

$$\hat{x}_0 = \arg\max_{S,\theta} \sum_{i=1}^{s} \sum_{k=1}^{90} \sum_{j=1}^{n} ((P(m_{L,k}^{*(j)}), \begin{array}{c} m_{L,k}^{*(j)} is\ occupied \\ m_{G,i}^{*(j)} is\ occupied \end{array}) \quad (23)$$

Obviously, the recovered initial pose might not be very accurate at the moment. Similar to the SLAM refining process, we zoom into a higher resolution level and execute an extensive search around the previous recovered initial pose: First, we generate candidate pose space by sampling in the neighborhood area of the recovered initial pose (in both position and orientation); Next, we create a series of local maps based on candidate poses; Again, we implement similar matching process between new local maps and global reference map. Finally, an accurate initial robot pose is found by the progressive pose searching and map matching procedures.

C. Position Tracking

By now, we can update the robot pose according to both the localization results of SLAM (relative moments) and the precise initial pose. After that, the global localization problem turns out to be a position tracking problem. In this paper, we address the position tracking problem by matching the current laser scan to the existing global map using similar mapping framework discussed in the local SLAM section.

D. Localization Quality Evaluation

From previous discussion, the robot is able to precisely localize itself in the global map through three processes: local SLAM, map matching and position tracking. However, to cope with the localization failure or recover from "kidnapping", we need to address the kidnapped robot problem.

For solving the kidnapped robot problem, a grid-map based coverage ratio evaluation method is presented to assess the localization quality so as to detect localization failure or "kidnapping". Suppose there are N laser points in the current scan. First we project all laser points into global map. Then, the evaluation algorithm performs status-checking operation towards the cells which are hit by those transformed laser points. Cell status V_i and coverage ratio R are calculated by (24) and (25) respectively.

$$V_i = \begin{cases} 1, & grid\ cell\ m^{(i)}\ is\ occupied \\ 0, & grid\ cell\ m^{(i)}\ is\ empty \end{cases} \quad (24)$$

$$R = \frac{\sum_{i=1}^{N} V_i}{N} \qquad (25)$$

If the coverage ratio R is smaller than a certain threshold, the quality evaluation module informs system that the robot has been kidnapped or reached a poor localization. Consequently, our system automatically reset the global localization and switch to re-localization procedure (As illustrated in Fig. 1).

IV. EXPERIMENTS EVALUATION

A. Experiment Configuration

The proposed localization architecture has been implemented in two different vehicle platforms. As Fig. 4(a) shows, AGV is a fully autonomous vehicle equipped with two IBEO Alasca-XT laser scanners (left and right front corner). Cycab as depicted in Fig. 4(b) is a prototype of smart car mounted with a single IBEO-ML laser scanner in front.

B. Experiment Results

A number of experiments in different scenarios have been carried out to test and validate the feasibility and robustness of our localization approach. As depicted in Fig. 5(a), the experiment environment covers an area of 25,000 m^2 along with buildings, parking slots, trees, bushes and other clutter. The corresponding global map is shown as Fig. 5(b). During the experiments, AGV has been randomly placed in different starting spots (two examples, A and B, shown in Fig. 5(b)). Then, AGV seeks to localize itself from scratch without any localization notification: It incrementally builds the local map and determines itself locally based on SLAM. After the local map accumulates enough evidence, AGV is capable to recover itself in the global map by our matching algorithm.

Fig. 6(a) and Fig. 6(b) illustrate the local maps starting from spot A and B respectively (AGV marked as the rectangle in Fig. 6). Map matching and global localization results are shown in Fig. 6(c) and Fig. 6(d). As we can see, AGV obtains an accurate global localization (relative to the global map) when the local map achieves a satisfying alignment with reference map (as depicted in Fig. 6(c) and Fig. 6(d), local map data in small circles are projected to global reference map for showing the map alignment).

Another experiment was carried out during a one-week public demonstration in Formello, Italy, March 2010. The demo site is located in a highly unstructured and complicated outdoor environment (see Fig. 7(a)) with garden benches, tents, trees, bushes and even running kids. As Fig. 7(b) illustrated, Cycab starts its autonomous running from point A and B respectively. Results of local maps and final global localization from A and B are described in Fig (8) in detail.

In fact, the global maps (as shown in Fig. 5(b) and Fig. 7(b)) are not up-to-date. For example, parking cars appearing and disappearing (labeled by c1 in Fig.6 (c), c2, c3 in Fig. 6(d)), dynamic moving objects (labeled by P1 in Fig. 8(d)),

Figure 4. Experiment vehicle platforms: (a) AGV; (b) Cycab.
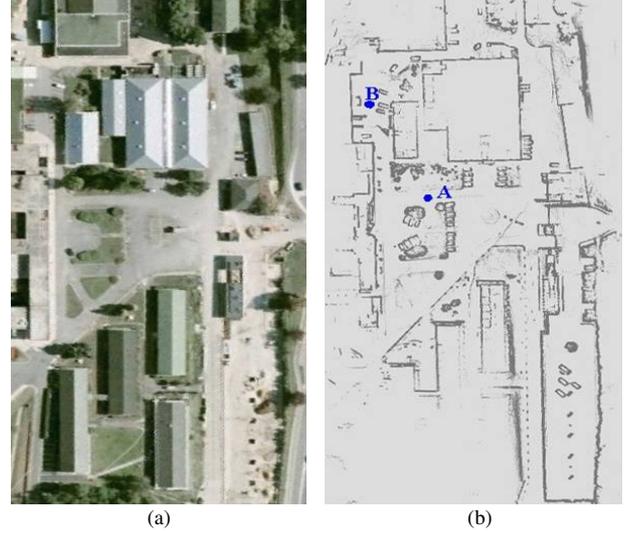

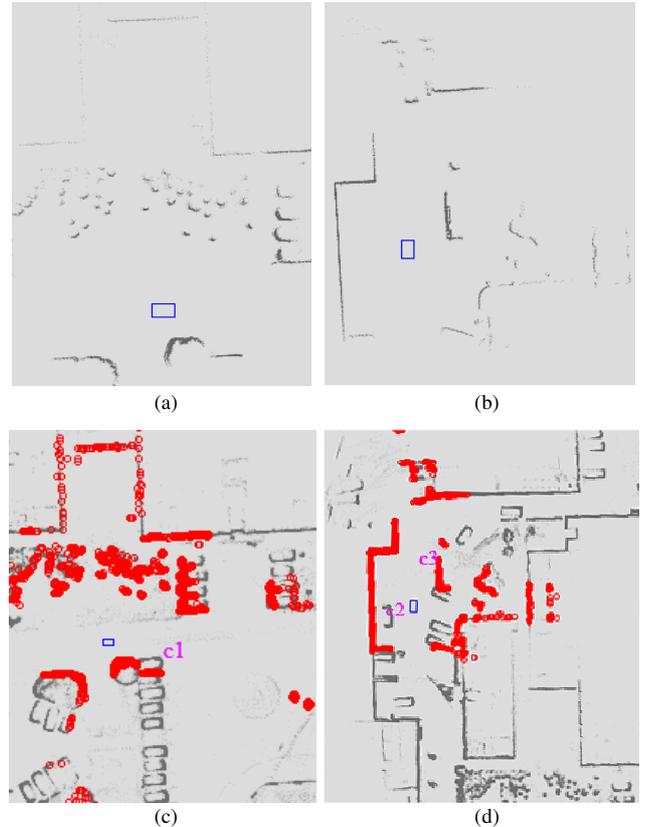Figure 5. Experiment scenario and map: (a) Scenario site; (b) Global map.

Figure 6. Experiment results of AGV: (a) Local map from A; (b) Local map from B; (c) localization result A; (d) localization result B.
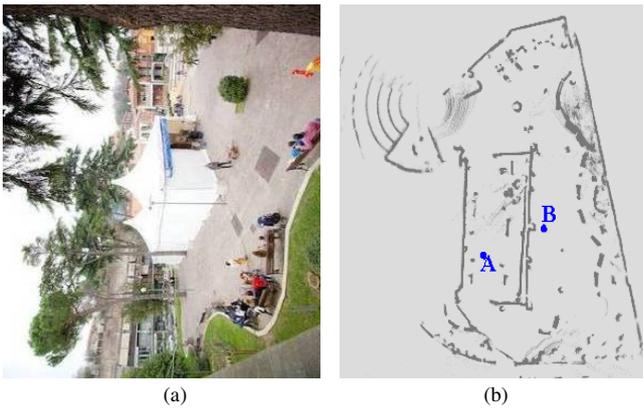
(a)   (b)

Figure 7.   Experiment scenario and map: (a) Scenario site; (b) Global map.

(a)   (b)

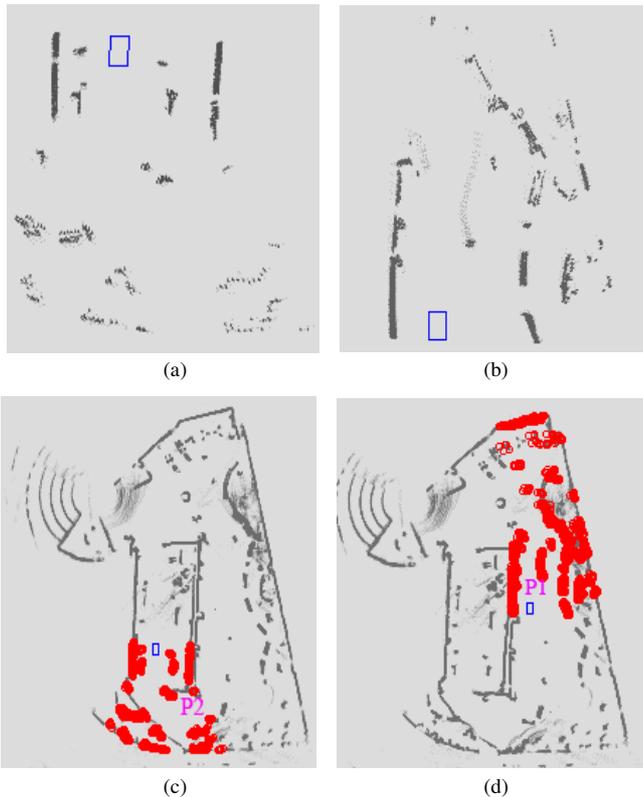(c)   (d)

Figure 8.   Experiment results of Cycab: (a) Local map from A; (b) Local
map from B; (c) localization result A; (d) localization result B.

scene structure change (labeled by P2 in Fig. 8(c)). However, as the real experiment results have shown, our localization approach is robust enough to those modifications in the scenes. This is because the map matching algorithm is mainly based on solid features and distinctive evidence which are unlikely to be modified in a short-term (i.e., city buildings, street infrastructures, trees, etc.). Overall, the proposed map matching algorithm is capable to determine robot localization despite of the minor environment changes. In a highly shape repetitive environment, the matching algorithm could be confused and forced to fail at the beginning. But, the robot can re-localize itself correctly in the global map after gaining sufficient evidence for localization distinction. Indeed, the

proposed localization approach could fail if the environments change significantly or the onboard laser sensors are "blocked" by the surrounding crowd.

To validate the ability of solving the kidnapped robot problem, we simulate the "kidnapping" situation by resetting a well-localized robot pose. More details as well as some experiment videos are available in [9]-[11].

V.   CONCLUSION

In this paper, we presented a high-efficient reliable global localization approach and demonstrated its good performance in challenging outdoor scenarios in real-time. The architecture of our localization system is mainly divided into four parts: local SLAM, map matching, position tracking and localization evaluation. Through local SLAM, map matching and position tracking, the robot can determine its localization relative to the global map. In addition, by incorporating localization quality evaluation algorithm, the robot is able to recover from the localization failure or "kidnapping" thus solving the kidnapped robot problem. In the future, for achieving truly "global" localization characteristic, we plan to associate GPS coordinate with the global map which enables our localization system to provide positioning information at the scale of the whole earth.

REFERENCES

[1]   I. J. Cox, and G. T. Wilfong, "Autonomous Robot Vehicle," Springer Verlag, 1990.

[2]   J. Leonard, and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp. 376-382, Jun. 1991.

[3]   S. Thrun, W. Burgard, and D. Fox. "Probabilistic Robotics," The MIT Press, Sept. 2005.

[4]   L. Zhang, and R. Zapata, "Probabilistic localization methods of a mobile robot using ultrasonic perception system," In Proc. of IEEE International Conference on Information and Automation, pp. 1062-1067, Jun. 2009.

[5]   H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," IEEE Robotics and Automation Magazine, vol. 13, no. 2, pp. 99-110, Jun. 2006

[6]   T.D. Vu, "Vehicle Perception: Localization, Mapping with Detection, Classification and Tracking of Moving Objects," PhD thesis, Grenoble Institute of Technology, 2009.

[7]   S. Thrun, W. Burgard, and D. Fox, "A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping," In Proc. of IEEE International Conference on Robotics & Automation, pp. 321-328, Apr. 2000.

[8]   J. Xie, O.G. Favrot, F. Nashashibi and M. Parent, "A Real-Time Robust SLAM for Large-scale Outdoor Environments," To appear In the Proc. of the 17th ITS World Congress, Oct. 2010. (accepted)

[9]   Localization Experiment using AGV. [Online]. Available: http://www.youtube.com/watch?v=elhI_3YQHIA.

[10] Localization Experiment using Cycab. [Online]. Available: http://www.youtube.com/watch?v=ZYBKZh0eOSU.

[11] CityNetMobil Demo in Formello, Italy. [Online]. Available: http://www.youtube.com/watch?v=QGwWaTT-WLA.