Robust Submap-Based Probabilistic Inconsistency Detection for Multi-Robot Mapping

Yufeng Yue[1], Danwei Wang[1], P.G.C.N. Senarathne[2] and Chule Yang[1]

***Abstract*— The primary goal of employing multiple robots in active mapping tasks is to generate a globally consistent map efficiently. However, detecting the inconsistency of the generated global map is still an open problem. In this paper, a novel multi-level approach is introduced to measure the full 3D map inconsistency in which submap-based tests are performed at both single robot and multi-robot level. The conformance test based on submaps is done by modeling the histogram of the misalignment error metric into a truncated Gaussian distribution. Besides, the detected inconsistency is further validated through a 3D map registration process. The accuracy of the proposed method is evaluated using submaps from challenging environments in both indoor and outdoor, which illustrates its usefulness and robustness for multi-robot mapping tasks.**

I. Introduction

With the maturity of the single robot technology, the use of a group of coordinated robots [1] in the past decade has been gradually taken seriously. The use of multiple robots significantly improve the efficiency and robustness in search and rescue tasks [2] and collaborative mapping [3]. One of the key challenges is to generate a globally consistent map of the environment. Global map generation is generally achieved by fusing the local maps generated by individual robots [4]. However, this is based on the assumption that the created local maps are consistent with no significant errors. Fusing erroneous local maps results in an inconsistent global map, which renders the accurate execution of autonomous tasks infeasible. Therefore detecting inconsistency in both local maps and fused global maps is vital for multi-robot missions.

The development of map inconsistency detection strategies has not received much attention, where the majority of the research is focused on generating accurate maps and post-processed optimization. However, map generated does not guaranteed to be foolproof and may generate inconsistent maps, especially in large environments and over long operating times. In addition, post-processed optimization is computationally expensive, where a proper triggering time and quantitative inconsistency measurements are required. Hence, there is a gap between the map generation and the map optimization, which is the inconsistency detection of the map. The capability to detect inconsistency in the generated maps allows robots either to perform an efficient optimization algorithm in order to continue the mission, or to return to a safe location for recovery without getting trapped.

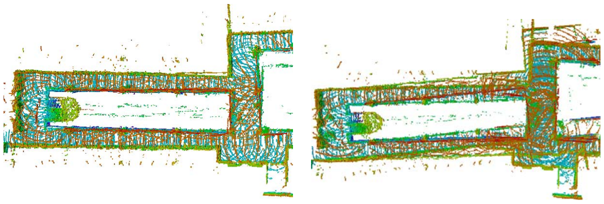

(a) Consistent map (b) Inconsistent map

Fig. 1. An example of consistent and inconsistent map

This paper presents a novel strategy that detects the inconsistency of map generated from two levels, i.e., single robot and multi-robot level. The inconsistency of the system is modeled as a probabilistic distribution, where submap-based inconsistency testing is performed on three types of inconsistency defined on both single robot and multi-robot level. In both levels, map inconsistency measurements are computed by modeling the misalignment errors between submap pairs into a truncated Gaussian distribution, and the resultant mean and variance are used to detect the inconsistency. Besides, the detected inconsistency is further validated through a 3D map registration process. The fused global map is deemed consistent only if all submap pairs of all robots pass the inconsistency checks.

Main contributions of this work are listed below:

- A novel multi-level topological framework is proposed to model the inconsistency of full 3D maps on both single robot and multi-robot level.
- A submap-based probabilistic algorithm is developed to efficiently detect and verify the three types of inconsistency (i.e., sequential, local and global).

The rest of this paper is organized as follows: Section II reviews the related literature. Section III presents the topological probabilistic structure. Section IV details submap-based inconsistency modeling and detection. Section V presents the experiments. Section VI concludes the paper.

II. Related Work

Various mapping algorithms have been proposed in the past decades [5] [6], however, these methods can't guarantee to produce globally consistent maps all the time. To increase the convergence of mapping, graph SLAM back-end has been proposed to minimize the accumulated error [7]. Based on that work, variants using dynamic covariance scaling [8] and switchable constraints [9] have been addressed.

*The research was partially supported by the ST Engineering-NTU Corporate Lab through the NRF corporate lab@university scheme.

[1] Yufeng Yue, Danwei Wang and Chule Yang are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

[2] P.G.C.N. Senarathne is with ST Engineering-NTU Corporate Laboratory, Nanyang Technological University, Singapore

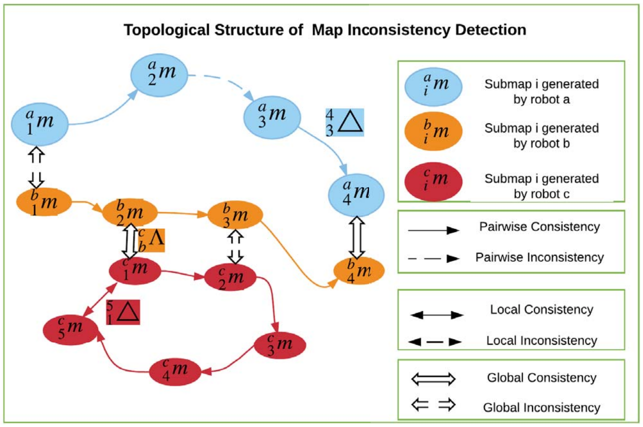


Fig. 2. Topological structure of map inconsistency detection. The nodes with different colors represents the submaps of different robots. The three types of edges show the inconsistency at different levels.

Some researchers present vision based place recognition to trigger back-end [10], which is sensor-dependent and needs to process huge amount of raw sensor data. Under the condition of limited communication and memory of multi-robot system, a compact map generated by compressing raw sensor data is an alternative choice. However, very few research has been conducted on detecting the inconsistency of generated maps, except the works proposed in [11] [12] [13]. In [11], a Bayesian representation of map posterior is used to detect the wrong data associations. Then it is extended to a multi-scan scenario with a cascaded map inconsistency statistical test in [12]. Based on that work, [13] developed a robot homing strategy whenever the map is detected to be inconsistent. These works pioneer the map inconsistency detection problem, and the use of this knowledge to improve autonomous mapping missions. However, these methods focus on sensor data level in a 2D environment with a single robot, which is infeasible to be extended to a multi-robot scenario in full 3D mapping missions.

In large environment, submap-based approach provides incremental [14] and efficient [15] approach for robot mapping. For multiple robots coordination [16], submap is even more suitable under the constraints of limited communication bandwidth and coverage. As submaps are generated over a short motion window of the robot, it is reasonable to assume that they are locally consistent [17]. To the extent of our knowledge, this is the first application of robust submap-based inconsistency detection for multi-robot mapping.

III. Topological Structure of Map Inconsistency Detection

This section presents the structure of submap-based inconsistency measurement, which is shown in Fig. 2. The submap-based inconsistency measurement is defined as a topological structure, where the node is defined as the submap and the edge is defined as the submap-wise inconsistency. Then we further explain the three types of inconsistency on single robot and multi-robot level.

A. Definition of Node

The node is defined as a submap and notation is denoted in Eq.(1)

$$^{x}_{i}V = \{^{x}_{i}m\}_{i\in n_x, x\in\mathcal{R}} \tag{1}$$

where n_x stands for the set of known submaps generated by robot x, and $\mathcal{R}$ represents the set of robots $\{a,b,c,\cdots\}$. Submap $\{^{x}_{i}m\}$ is generated when certain conditions have met and is added to the topology as a new node. In this paper, a submap is represented as a 3D probabilistic occupancy voxel map [5]. Considering the limited size of submap will lead to less drift, while still providing sufficient submap size for map registration. The criterion that triggers the creation of a new submap is based on the condition that the robot moved for a certain distance or rotated for a certain angle.

Based on submaps generated, the local map and global map on a higher level are defined below.

1) Local map generated by a single robot: Local map ^{x}m generated by a single robot x consists of submaps $\{^{x}_{i}m\}$ with $i \in n_x$, which is shown as the chain of node with the same color in Fig. 2 and is denoted as:

$$^{x}m = \{^{x}_{i}m, i \in n_x\} \tag{2}$$

2) Global map generated by multi-robot: Global map M that integrates 3D occupancy local maps ^{x}m with $x \in \mathcal{R}$ is defined as:

$$M = \{^{x}m, x \in \mathcal{R}\} \tag{3}$$

B. Definition of Consistency

Map inconsistency is defined as a measurement of alignment between the submaps. On that basis, consistency indicates that submaps have been registered properly, while inconsistency implies a large offset between two submaps. The inconsistency between two submaps is represented by the edge, which defines the inconsistency between two submaps on single robot level and multi-robot level.

1) single robot level: The submaps i, j within the local map ^{a}m connected by an edge $^{i}_{j}\triangle$ is denoted as single robot level inconsistency, which has two types: sequential inconsistency and local inconsistency.

a) Sequential inconsistency: Sequential inconsistency is defined as the probability between consecutive submaps $^{a}_{i}m$ and $^{a}_{i-1}m$ and is defined in Eq.(4). Sequential inconsistency always happens when two consecutive submaps didn't align properly due to SLAM error like a sharp turn occurs or a sudden drift in odometry sensor.

$$^{i}_{i-1}\triangle = p(^{a}_{i}m|^{a}_{i-1}m)_{\{i=2:n_a\}} \tag{4}$$

b) Local inconsistency: local inconsistency is defined in Eq.(5), which calculates the probability of $^{a}_{i}m$ over the past submaps $^{a}_{j}m$ generated by robot a.

$$^{i}_{j}\triangle = \; p(^{a}_{i}m|^{a}_{j}m)_{\{i=k+1:n_a, j=1:i-k\}} \tag{5}$$

The calculation is not performed from $i-k$ to $i-1$ submaps because $^{a}_{i}m$ always has small overlapping with those submaps. Local inconsistency always happens when the robot in current submap $^{a}_{i}m$ didn't recognize the previously traveled submap $^{a}_{j}m$.

2) Multi-robot level: The local maps ${}^{a}m$, ${}^{b}m$ connected by an edge ${}^{a}_{b}\Lambda$ is denoted as multi-robot level inconsistency. Global inconsistency is defined as probability of submap ${}^{b}_{j}m$ in robot b over all the submaps ${}^{a}_{i}m$ in robot a, which is defined in Eq.(6). Global inconsistency always happens when two robots can not recognize the same place they visited.

$$ {}^{a}_{b}\Lambda = p({}^{b}m|{}^{a}m) \propto \prod_{i=1:n_a, j=1:n_b} p({}^{b}_{j}m|{}^{a}_{i}m) \tag{6}$$

3) Overall system: With the three types of inconsistency defined above, we can write the inconsistency measurement for the whole system. Here, we denote the inconsistency of the system in Eq.(7).

$$\begin{aligned} &\prod_{a} \underbrace{p({}^{a}_{i}m|{}^{a}_{j}m)}_{\text{single robot level}} \cdot \prod_{a,b} \underbrace{p({}^{b}_{j}m|{}^{a}_{i}m)}_{\text{multi-robot level}} \\ =&\prod_{a} \left(\underbrace{p({}^{a}_{i}m|{}^{a}_{i-1}m)}_{\text{pairwise consistency}} \underbrace{p({}^{a}_{i}m|{}^{a}_{j}m)}_{\text{local consistency}} \right) \cdot \prod_{a,b} \underbrace{p({}^{b}_{j}m|{}^{a}_{i}m)}_{\text{multi-robot level}} \end{aligned} \tag{7}$$

Since the three types of inconsistency defined can be represented at submap-level, the inconsistency measurement can be modeled as a Gaussian distribution $f(x;\mu,\sigma^2)$. The modeling will be detailed in Sec.(IV).

IV. Submap-Based Inconsistency Detection

In this section, the submap-based inconsistency measurement is calculated by modeling the histogram of inconsistency distance into a truncated Gaussian distribution. Then submap-based inconsistency testing is performed and verified by applying 3D map registration.

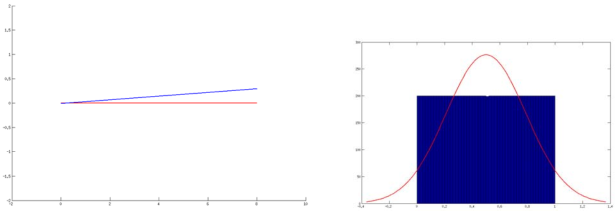
A. Submap-wise Inconsistency Distance

A proper inconsistency distance should be defined to describe the discrepancy between submaps. Here, we assume m and n as two submaps, where m_i and n_j are two voxels in submap m and n, respectively. The edge connects m and n can be any of the three types of inconsistency defined in Sec.III.

The distance applied here is the Occupancy Iterative Closet Point(OICP) distance defined in [4]. The OICP distance is an error metric specially designed for 3D occupancy grid map that combines the Euclidean distance $d_e(m_i,n_j)$ and occupancy probability distance $d_f(m_i,n_j)$. The inconsistency distance between a pair-wise matched voxels is defined in Eq.(8). The details of OICP distance can be found in [4].

$$d(m_i,n_j) = d_e(m_i,n_j) + d_f(m_i,n_j) \tag{8}$$

Where $d(m_i,n_j)$ is the OICP distance between voxel m_i in submap m to the closest voxel n_j in submap n. The histogram of OICP distance over all the matched voxels (i,j) is computed to show the distribution of the inconsistency distances. Since inconsistency is defined as a probability distribution, the histogram is further modeled into probability distribution to describe the discrepancy between submaps.



(a) The pair-wise submaps are misaligned with an error

(b) Uniform histogram is modeled into guassian distribution

Fig. 3. An example of misaligned submaps and the histogram of inconsistency distance

B. Modeling of Probability Distribution

For simplification, we assume the submaps m and n to be straight lines and the probability of each voxel equals to 1. Histograms of perfect alignment and misalignment are modeled into Gaussian distributions $f(x;\mu,\sigma^2)$.

1) Modeling of Perfect Alignment : Assuming two submaps are noise free and perfectly aligned, which means the distance metric $d(m_i,n_j)$ between all corresponding voxels equals to zero. Then the histogram is subject to a Dirac delta distribution, which is the limit the normal distribution when $\sigma^2 \to 0$.

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}}; \ \mu = 0, \ \sigma^2 \to 0, \tag{9}$$

However, in real environment the map will be noisy and alignment will have a distance residual, the σ^2 will always greater than zero.

2) Modeling of Misalignment: Considering the inconsistency situation for straight lines, the two lines will intersect at a point. Here, we assume the two lines intersect at the start point, as shown in Fig. 3a. The parallel lines are not considered since submap overlapping is a basic requirement for inconsistency detection.

The histogram of distance distribution complies a uniform distribution and is shown in Fig. 3b.

$$f(x) = \begin{cases} \frac{1}{b-a} & for\ a \leq x \leq b \\ 0 & for\ a \leq 0\ or\ x \geq b \end{cases} \tag{10}$$

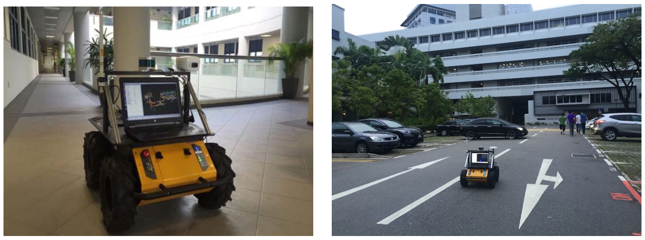
The uniform histogram can also be modeled by a Gaussian distribution with mean μ and variance σ^2 as shown in Fig. 3b.

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{-(x-\mu)^2}{2\sigma^2}}; \ \mu = \frac{a+b}{2}, \ \sigma^2 = \frac{(b-a)^2}{12} \tag{11}$$

Based on the observations, the simplification reveals that μ and σ^2 can be used to describe the discrepancy. In general, large μ and σ^2 indicate a large misalignment error.

C. Truncated Gaussian Distribution

As shown above, the histogram can be modeled into Gaussian distribution $X \sim N(\mu,\sigma^2)$. Since the distance is absolute value, the Gaussian distribution can be modified to truncated Gaussian distribution in the interval of $[a,b]$. Given



(a) Indoor experiment

(b) Outdoor experiment

Fig. 4. The environment of the experiment

a normal distribution $X \sim N(\mu, \sigma^2)$ with the probability density function(pdf) $\Phi(x)$ and cumulative density function(cdf) $\Psi(x)$. Then, $X \sim N(\mu, \sigma^2)$ conditioned on interval $a \le x \le b$ is derived:

$$f(x;\mu,\sigma^2;a,b) = f_t(x;\mu_t,\sigma_t^2) = \frac{\Phi(x)}{\Psi(b)-\Psi(a)} \tag{12}$$

Let $\alpha = \frac{a-\mu}{\sigma}$, $\beta = \frac{b-\mu}{\sigma}$where truncated mean μ_t and truncated variance σ_t^2 are calculated as follows:

$$\mu_t = \mu + \sigma\frac{\Phi(\alpha)-\Phi(\beta)}{\Psi(\beta)-\Psi(\alpha)} \tag{13}$$

$$\sigma_t^2 = \sigma^2[1 + \frac{\alpha\Phi(\alpha)-\beta\Phi(\beta)}{\Psi(\beta)-\Psi(\alpha)} - (\frac{\Phi(\alpha)-\Phi(\beta)}{\Psi(\beta)-\Psi(\alpha)})^2] \tag{14}$$

Here we set $a = 0$, as distance is always no less than 0, and $b = d$, where d is the maximum search distance of OICP registration algorithm. On the interval of $[0,d]$, μ_t describes the average misalignment distance, and σ_t^2 describes the degree of misalignment divergence.

D. Inconsistency Testing

In this part, a test is performed to detect the three types of inconsistency definted in Sec.III.

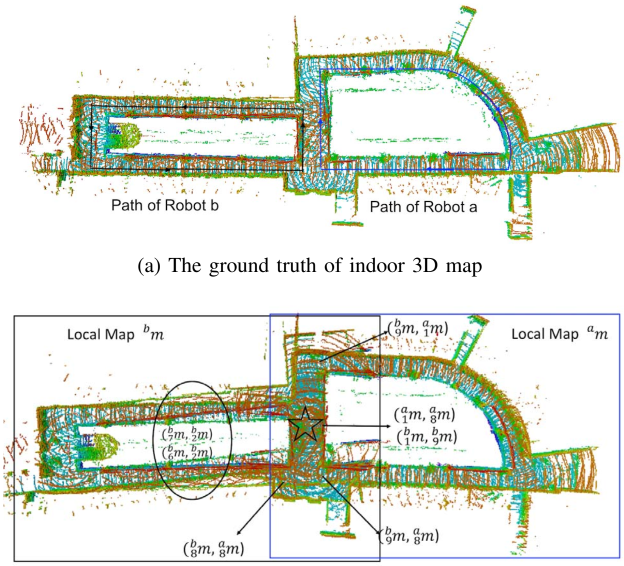
Given all the submaps $\{{}^{x}_{i}m\}_{i\in n_x, x\in\mathcal{R}}$, the overlapping between submaps is firstly tested. Then the histogram of misalignment distances is modeled into truncated Gaussian distribution. An $N \times N$ square matrix Γ is generated that contains Γ_{ij} value defined in Eq.(15), where $N = \sum_{x\in\mathcal{R}} n_x$ is the total number of submaps. Here, the sparsity of matrix Γ depends on the pairs of overlapping submaps.

$$\Gamma_{i,j} = \frac{1}{\mu_t \cdot \sigma_t^2} \tag{15}$$

Inconsistency testing is performed for each $\Gamma_{i,j}$, a large value of $\Gamma_{i,j}$ indicates consistency, while a low value implies that large misalignment compared with the perfect map alignment. A global map is consistent if all tests are successful. For inconsistent submaps, they share a common area, however, are not aligned properly. Hence, we apply a novel map registration algorithm [4] to align the inconsistent submaps and generate a $N \times N$ matrix Θ. By comparing matrix Γ with Θ, the large difference verifies the detected inconsistency . An overall algorithm is shown below.

Algorithm 1 Submap-Based Inconsistency Detection

```
Require: The generated submaps $\{{}^{x}_{i}m\}_{i\in n_x, x\in\mathcal{R}}$
Ensure: Detecting sequential, local and global inconsistency
  for $\{{}^{x}_{i}m\}_{i\in n_x, x\in\mathcal{R}}$ with overlapping pairs do
    for single robot level do
      Sequential inconsistency: ${}^{i}_{i-1}\triangle = p({}^{a}_{i}m|{}^{a}_{i-1}m)_{\{i=2:n_a\}}$
      Local inconsistency: ${}^{i}_{j}\triangle = p({}^{a}_{i}m|{}^{a}_{j}m)_{\{i=k+1:n_a, j=1:i-k\}}$
    end for
    for multi-robot level do
      Global inconsistency: ${}^{i}_{j}\Lambda = p({}^{b}_{j}m|{}^{a}_{i}m)_{\{i=1:n_a, j=1:n_b\}}$
    end for
  end for
  Generating inconsistency matrix $\Gamma_{N\times N}$
  for $\Gamma_{(i,j)} > 0$ do
    Generating matrix $\Theta$ after performing map registration
  end for
  Verified inconsistency matrix $\Gamma^v = \Theta - \Gamma$
```


Path of Robot b
Path of Robot a
Local Map ${}^{b}m$
Local Map ${}^{a}m$
$({}^{b}_{9}m, {}^{a}_{1}m)$
$({}^{a}_{1}m, {}^{a}_{8}m)$
$({}^{b}_{1}m, {}^{b}_{9}m)$
$({}^{b}_{7}m, {}^{b}_{2}m)$
$({}^{b}_{6}m, {}^{b}_{2}m)$
$({}^{b}_{8}m, {}^{a}_{8}m)$
$({}^{b}_{9}m, {}^{a}_{8}m)$

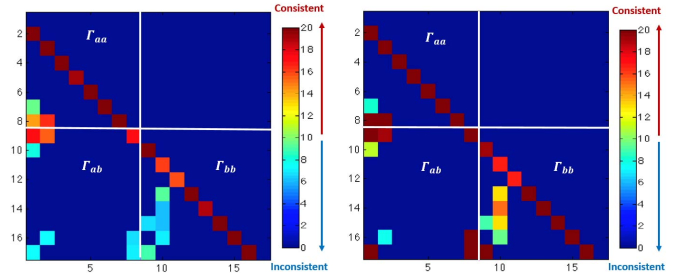
(a) The ground truth of indoor 3D map

(b) The inconsistent indoor 3D map

Fig. 5. The ground truth and inconsistent map generated in indoor environment. For local map ${}^{a}m$ is consistent while the local map ${}^{b}m$ can't close the loop when it comes back to the starting point.

V. Experimental Results

Experiments performed using two robots in indoor and outdoor environments are presented in this section. Two robots were teleoperated in both indoor and outdoor environment at Nanyang Technological University. The robot was equipped with a Hokuyo Laser range finder for pose estimation using Gmapping [18] and a Velodyne VLP-16 for 3D perceptions, as shown in Fig. 4. The individually generated 3D grid maps are post-processed using C++ and Matlab to analyze their inconsistency. Submaps are generated at $10m$ intervals. The resolution of the 3D occupancy grid map in both experiments is set to be $0.1m$.



(a) Inconsistency matrix Γ

(b) Inconsistency matrix Θ after map registration

Fig. 6. The deeper the color, the submap pair is more consistent. The large difference of the color in the same pixel verifies the inconsistency.

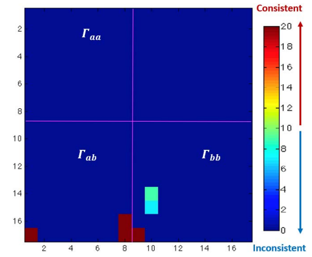


Fig. 8. The matrix Γ^{v} shows the verified inconsistent pairs by identifying the difference between Fig .6a and Fig. 6b in indoor scenario

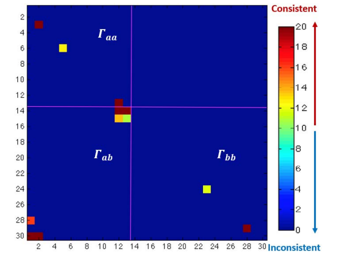


Fig. 9. Verified inconsistency matrix in outdoor environment. The deeper color represents a higher probability of inconsistency

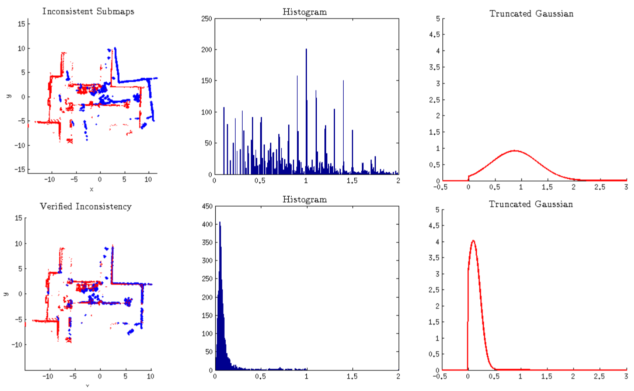


Fig. 7. The top row shows the detected global inconsistency between submap ${}^{b}_{9}m$(blue color) and ${}^{a}_{8}m$(red color), while the bottom row verifies the inconsistency after performing map registration

A. Indoor Environment

The ground truth map and the inconsistent map are shown in Fig. 5. The ground truth of the map was generated with a very high number of particles and by moving the robot extremely slowly and smoothly. For experiments, the number of particles was decreased to the default value of 30. The two separated local maps are fused to generate the global map and the result is shown in Fig. 5b. For local map ${}^{a}m$, 8 submaps were generated. And 9 submaps were generated for the local map ${}^{b}m$.

Square matrix $\Gamma_{17\times17}$ is generated to describe the inconsistency as shown in Fig. 6a. As for single robot level inconsistency detection, $\Gamma_{[1:8,1:8]}(\Gamma_{aa})$ shows the inconsistency between submaps ${}^{a}_{1:8}m$, and $\Gamma_{[9:17,9:17]}(\Gamma_{bb})$ shows the inconsistency between submaps ${}^{b}_{1:9}m$. For multi-robot level inconsistency, $\Gamma_{[9:17,1:8]}(\Gamma_{ab})$ represents the global consistency between submaps ${}^{b}_{1:9}m$ and submaps ${}^{a}_{1:8}m$. The two map sessions started and ended at the same position and is marked with a star in Fig. 5b. Note that only sequential inconsistency detection is performed, so the matrix is not symmetric.

1) Single robot level: For local map ${}^{a}m$ and ${}^{b}m$, the sequential consistency ${}^{i}_{i-1}\triangle$ are accepted. As shown in matrix Γ, all the pixels representing sequential inconsistency is with high values. Local inconsistency for map ${}^{a}m$ is also accepted, which means the local loop is closed. For example, the high value of $\Gamma_{(8,1)}$ and $\Gamma_{(7,1)}$ indicates last two submaps ${}^{a}_{7}m$, ${}^{a}_{8}m$ aligned properly with first submap ${}^{a}_{1}m$. However, the low value for $\Gamma_{(13:16,10)}$, $\Gamma_{(15,9)}$, and $\Gamma_{(17,9)}$ indicate the local inconsistency detected in ${}^{b}m$. For example, $\Gamma_{(14,10)}$ shows the inconsistency between ${}^{b}_{6}m$ and ${}^{b}_{2}m$. And $\Gamma_{(17,9)}$ shows the misalignment between first submap ${}^{b}_{1}m$ and last submap ${}^{b}_{9}m$.

2) Multi-robot level: The measured global inconsistency is presented in lower left part in Fig. 6a, and is based on the overlapping area between ${}^{a}m$ and ${}^{b}m$ in Fig. 5b. Pixel $\Gamma_{(9,1)}$ is with a high value, because robot a and b started at the same position and aligned properly. Then $\Gamma_{(17,1)}$ and $\Gamma_{(16,2)}$ are with a low value, due to the misalignment between last two submaps ${}^{b}_{9}m$, ${}^{b}_{8}m$ and start position of ${}^{a}_{1}m$ and ${}^{a}_{2}m$. The low values of $\Gamma_{(16,8)}$ and $\Gamma_{(17,8)}$ also detects the inconsistency between submaps (${}^{b}_{8}m$,${}^{a}_{8}m$) and (${}^{b}_{9}m$,${}^{a}_{8}m$).

3) Map Registration: As mentioned before, inconsistency is caused by misalignment between submaps. Here, a map registration algorithm is applied to verify the potential inconsistency detected. Square matrix Θ is generated and is shown in Fig. 6b, which shows that the previously inconsistent submaps have been aligned after performing submap registration.

At single robot level, the sequential inconsistency remains unchanged, which validated our measurements. For ${}^{a}m$, the local inconsistency is also validated. For ${}^{b}m$, the local inconsistency value of $\Theta_{(14,10)}$, $\Theta_{(15,10)}$, $\Theta_{(17,9)}$ have increased sharply, which indicates these pairs have been aligned after map registration. At multi-robot level, the global inconsistency value $\Theta_{(16,8)}$, $\Theta_{(17,1)}$ and $\Theta_{(17,8)}$ have all been detected with a big change. An example of $\Theta_{(17,8)}$ is shown in Fig. 7, which is a detailed explanation of the inconsistency between submap ${}^{b}_{9}m$ and ${}^{a}_{8}m$.

4) Verified Inconsistency Matrix: The verified inconsistency matrix Γ_{v} is generated by identifying the large differences before and after map registration in corresponding consistency matrix Γ and Θ. As illustrated in Fig. 8, local inconsistency is verified in local map ${}^{b}m$ for $\Gamma^{v}_{(14,10)}$, $\Gamma^{v}_{(15,10)}$ and $\Gamma^{v}_{(17,9)}$. Global inconsistency between ${}^{a}m$ and ${}^{b}m$ is verified for $\Gamma^{v}_{(17,1)}$, $\Gamma^{v}_{(16,8)}$ and $\Gamma^{v}_{(17,8)}$. The shade of color indicates the confidence level of the inconsistency detected. Inconsistency detected in $\Gamma^{v}_{(13,10)}$, $\Gamma^{v}_{(16,10)}$ and $\Gamma^{v}_{(15,9)}$ have been rejected due to the slight change after performing map

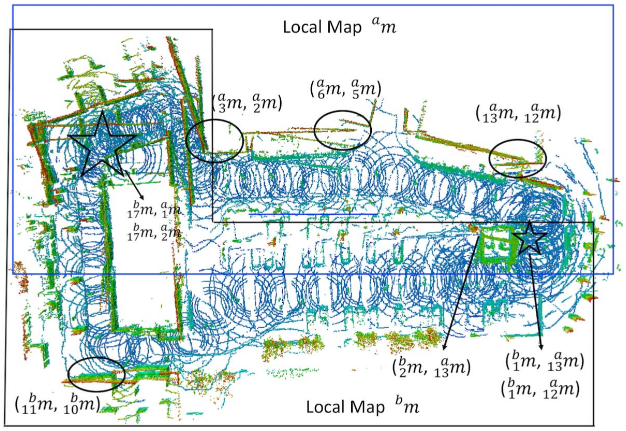


Fig. 10. The inconsistent map generated in a car park. Due to the unstructured environments and moving objects, the map is quite inconsistent. Our method is able to detect the inconsistency accurately.

registration. This is usually caused by low overlapping areas.

B. Outdoor Environment

The outdoor experiment was performed in an open car park in NTU. As can be seen from Fig. 4b, the environment was quite challenging with moving cars and people. Ground truth map was hard to generate in the environment, hence we only show the inconsistent map and inconsistency detected in Fig. 10. The start and end position are marked with stars. For local map $^a m$, 13 submaps were generated. And 17 submaps were generated for the local map $^b m$.

Since there is no local loop for the local maps $^a m$ and $^b m$, only sequential inconsistency and global inconsistency is detected. The verified inconsistency matrix Γ^v is shown directly to describe the inconsistency in Fig. 9.

a) single robot Level: The inconsistency of robot a is shown in $\Gamma^v_{[1:13,1:13]}$. Pairwise inconsistency is detected in $\Gamma^v_{(3,2)}$, $\Gamma^v_{(6,5)}$ and $\Gamma^v_{(13,12)}$. For local map $^b m$, the inconsistency is shown in $\Gamma^v_{[14:30,14:30]}$ for submaps $^b m_{1:17}$. Sequential inconsistency is detected in $\Gamma^v_{(24,23)}$ and $\Gamma^v_{(29,28)}$. Three pairs of inconsistent submaps are marked with ellipses in Fig. 10.

b) Multi-Robot Level: $\Gamma^v_{[14:30,1:13]}$ represents the global inconsistency between submaps $^b_{1:17}$ and $^a_{1:13}$. Global inconsistency is detected in $\Gamma^v_{(30,1)}$, $\Gamma^v_{(30,2)}$ and $\Gamma^v_{(28,1)}$. In addition, $\Gamma^v_{(14,12)}$, $\Gamma^v_{(14,13)}$, $\Gamma^v_{(15,12)}$ and $\Gamma^v_{(15,13)}$ represents the inconsistency detected between $(^b_1m, ^a_{12}m)$, $(^b_1m, ^a_{13}m)$ $(^b_2m, ^a_{12}m)$ and $(^b_2m, ^a_{13}m)$. The global inconsistency detected is shown in the overlapping area between $^a m$ and $^b m$ in Fig. 10.

VI. Conclusion

In this paper, the problem of measuring the inconsistency of maps generated by multi-robot mapping missions is addressed. We proposed a multi-robot map inconsistency detection strategy that evaluates the inconsistency on local and global levels. The inconsistency is measured by modeling the submap-wise misalignment error metric distribution into a truncated Gaussian distribution. In addition, a map registration algorithm is applied to verify the detected inconsistency. Our method successfully detects the inconsistency in the challenging indoor and outdoor environment. More importantly, as shown in Fig.(8) and Fig.(9), the inconsistency of different types(ie. sequential, local and global) are detected with a probability distribution, which can be the uncertainty of post-processed optimization or the signal of resetting mapping mission. The proposed submap-based framework is demonstrated to reduce the problem complexity and to accurately detect inconsistency in the global maps.

In future work, it can be integrated with a multi-robot exploration mission where it provides the ability to recover and continue the mission when the global map is inconsistent.

References

[1] Z. Yan, N. Jouandeau, and A. A. Cherif, “A survey and analysis of multi-robot coordination,” *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[2] D. Scaramuzza, M. C. Achtelik, L. Doitsidis *et al.*, “Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments,” *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 26–40, Sept 2014.

[3] S. Saeedi, M. Trentini, M. Seto, and H. Li, “Multiple-robot simultaneous localization and mapping: A review,” *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[4] Y. Yue, D. Wang, P. Senarathne, and D. Moratuwage, “A hybrid probabilistic and point set registration approach for fusion of 3D occupancy grid maps,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2016, pp. 1975–1980.

[5] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, 2013.

[6] C. Fu, A. Carrio, and P. Campoy, “Efficient visual odometry and mapping for unmanned aerial vehicle using arm-based stereo vision preprocessing system,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2015, pp. 957–962.

[7] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based slam,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[8] P. Agarwal, G. D. Tipaldi, and L. Spinello, “Robust map optimization using dynamic covariance scaling,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 62–69.

[9] N. Sünderhauf and P. Protzel, “Switchable constraints for robust pose graph slam,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1879–1884.

[10] Y. Latif, C. Cadena, and J. Neira, “Robust loop closing over time for pose graph slam,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1611–1626, 2013.

[11] D. Hähnel, S. Thrun, B. Wegbreit, and W. Burgard, “Towards lazy data association in slam,” in *Robotics Research. The Eleventh International Symposium*. Springer, 2005, pp. 421–431.

[12] M. Mazuran, G. D. Tipaldi, and L. Spinello, “A statistical measure for map consistency in slam,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3650–3655.

[13] I. Bogoslavskyi, M. Mazuran, and C. Stachniss, “Robust homing for autonomous robots,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2550–2556.

[14] J. L. Blanco, J. A. Ferndez-Madrigal, and J. Gonzez, “Toward a unified bayesian approach to hybrid metric–topological slam,” *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 259–270, April 2008.

[15] K. Ni and F. Dellaert, “Multi-level submap based slam using nested dissection,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2558–2565.

[16] M. J. Schuster, C. Brand, and b.-p. y. o. Hirschmüller, Heiko, “Multi-robot 6d graph slam connecting decoupled local reference filters.”

[17] E. Rehder and A. Albrecht, “Submap-based slam for road markings,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1393–1398.

[18] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb 2007.