Available online at www.sciencedirect.com

ScienceDirect

CAAI Transactions on Intelligence Technology 1 (2016) 90–103
http://www.journals.elsevier.com/caai-transactions-on-intelligence-technology/

Original article

A framework for multi-session RGBD SLAM in low dynamic workspace environment

Yue Wang [a], Shoudong Huang [b], Rong Xiong [a,*], Jun Wu [a]

[a] State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, PR China
[b] Centre for Autonomous Systems, Faculty of Engineering and IT, University of Technology, Sydney, Australia

Available online 4 June 2016

Abstract

Mapping in the dynamic environment is an important task for autonomous mobile robots due to the unavoidable changes in the workspace. In this paper, we propose a framework for RGBD SLAM in low dynamic environment, which can maintain a map keeping track of the latest environment. The main model describing the environment is a multi-session pose graph, which evolves over the multiple visits of the robot. The poses in the graph will be pruned when the 3D point scans corresponding to those poses are out of date. When the robot explores the new areas, its poses will be added to the graph. Thus the scans kept in the current graph will always give a map of the latest environment. The changes of the environment are detected by out-of-dated scans identification module through analyzing scans collected at different sessions. Besides, a redundant scans identification module is employed to further reduce the poses with redundant scans in order to keep the total number of poses in the graph with respect to the size of environment. In the experiments, the framework is first tuned and tested on data acquired by a Kinect from laboratory environment. Then the framework is applied to external dataset acquired by a Kinect II from a workspace of an industrial robot in another country, which is blind to the development phase, for further validation of the performance. After this two-step evaluation, the proposed framework is considered to be able to manage the map in date in dynamic or static environment with a noncumulative complexity and acceptable error level.

Keywords: Multi-session SLAM; RGBD sensor; Low dynamic mapping

1. Introduction

Simultaneous localization and mapping (SLAM) has been a core technique enabling the autonomy of robots, such as robot car [1] and autonomous underwater vehicle [2]. Compared to these high cost and large devices, the small to medium sized devices, such as mobile manipulator, flying robot and hand-hold devices began to raise the attention in recent years due to the high flexibility, low cost and thus the highly promising application. These devices also call for SLAM to achieve the capacity of long-term operation. The main challenge for such a solution includes three aspects: (1) the flying robot and hand-hold devices have a motion pattern with more frequent changes of orientation due to the free environment (no ground plane); (2) these devices aim on low cost, light-weighted and small scale, hence expensive or heavy sensors cannot be equipped; (3) these devices usually work periodically in a pre-defined human sharable workspace with low dynamics, which means objects in the workspace may be moved, added or removed across multiple sessions.

Consumer-level RGBD sensor has made it very convenient to collect both intensity and depth information at a low cost. For the first two challenges, we apply the RGBD sensor for perception, enabling the 3D pose estimation but also a dense environment map for subsequent navigation. The third challenge is to deal with the change of objects (move, add, remove) across multiple sessions. A quick solution is to build

[*] Corresponding author. State Key Laboratory of Industrial Control and Technology, Zhejiang University, 38 Zheda Road, Xihu District, Hangzhou, 310027, PR China.

E-mail address: rxiong@iipc.zju.edu.cn (R. Xiong).

Peer review under responsibility of Chongqing University of Technology

http://dx.doi.org/10.1016/j.trit.2016.03.009

the map at each session, but this method discards all history experience. Our solution is to manage the dynamics in a map. Specifically, the multi-session SLAM component was utilized to accumulate the map building. On the top of that, a map management component was proposed to keep the map compact and in track of the environment changing. With this framework, we are able to address all three challenges.

In the previous studies, various SLAM methods have been presented for mapping the environment with this kind of sensors. Existing RGBD mapping methods were mainly on single session and for relatively static environment or with high dynamics [3–5]. However, the low dynamics emerging in multi-session scenario did not draw much attention. Some methods [6–8] were proposed to deal with the challenge. They used vision or planar laser sensor, which captured limited dynamics and cannot be simply extended to that using RGBD sensor. The methods using vision sensor can tell whether a frame has a significant change in appearance as it is feature based. Since the RGBD sensor also provides depth information, we can capture the geometric change and know exactly what is changed in a frame. The methods using laser sensor usually took a 2D grid occupancy map as its map representation, which is not available in RGBD SLAM system due to the high complexity of 3D grid. Besides, the dynamics captured in 2D is only a slice of the 3D dynamics, which can be semantically insufficient.

To the best of our knowledge, our system may be the first one that build the map over the low dynamic environment using only a RGBD sensor in a scenario of 6 DoF multi-session SLAM. We proposed a framework that can build the map keeping track of the current environment, preventing the change of environment in previous sessions incorporated. Fig. 1 gives the comparison between the final map generated by multi-session SLAM system with and without considering low dynamics in a workspace in office environment. The objects (books, cans, boxes and so on) are added, removed and moved across the sessions. After 10 sessions of SLAM, the SLAM without considering the low dynamics mixed the current and out-of-dated information together, leading to a useless map with incorrect duplicated objects, while the proposed system considering the low dynamics, demonstrated the current environment in the map.

The main contributions of this paper include:

- A framework is proposed for multi-session RGBD SLAM in low dynamic environment consists of two components: multi-session SLAM and graph management. The multi-session SLAM component has a graph model with each node being a pose and each edge being a constraint, thus fusing the information from previous sessions and current session to keep the map in one global coordinates. The graph management component can keep the graph model in date and with non-accumulative complexity using the out-of-dated scan identification module and redundant scan identification module.
- An out-of-dated scan identification module is proposed to find the previous pose with RGBD scan on the environment which is changed in current session. The goal of this module can be explained by setting an example, a cup was on the desk in previous sessions, but is removed in the current session. Then, the poses observing that cup on the desk should be found and pruned to keep the map in track of the environment changes. Because the unavailability of grid occupancy model, our idea is to adopt camera projection model and connected component detection to find the difference between the maps generated by the scans in the previous sessions and that in the current session. With this method, the poses reserved are always with in-dated scans and robust to noise and holes occurred in RGBD sensor.
- A redundant scan identification module is proposed to find the pose with RGBD scan having large overlapping part with others. This module is to reduce the number of poses if the number of in-dated scan is higher than a pre-defined threshold, which enables the computational time of the SLAM relevant to the size of the map and one session SLAM, instead of the all sessions SLAM. The idea of our method is to find a subset of poses that can generate a map similar to the original one using all poses, in the measure of Kullback–Leibler divergence. By applying this method, when a robot executes multi-session SLAM in a fixed sized static region in low dynamic environment, the computation complexity will keep constant since poses with redundant scans have been pruned despite that they are in date.

To show the performance of the framework, we tune and test the algorithm in a 2-session and 10-session dataset on a workspace in office environment with multiple objects moved, added and removed across the sessions, which is collected by a hand hold Kinect sensor. The result in Fig. 1 validates the effectiveness of the proposed method. After that, the framework was applied on a 5-session external dataset captured in workspace of an industrial robot with various sized boxes manipulated across sessions, which is blind to the development phase, for evaluation of real performance. The remainder of the paper is organized as follows: In Section 2, related works on mapping dynamic environments and pose pruning will be discussed. In Section 3, the proposed framework for multi-session RGBD SLAM in low dynamic environment is introduced. In Sections 4 and 5, the proposed out-of-dated scans identification and redundant scans identification will be presented in detail. In Section 6, we will demonstrate the experimental results using the real world datasets. The conclusion and future work will be discussed in Section 7.

2. Related works

The multi-session SLAM in static environment is first presented in vision based methods [1,6,9]. These works formulate the basic concept that the robot cannot simply start a new mapping session without using the information in previous sessions, since the constraints in past sessions provide

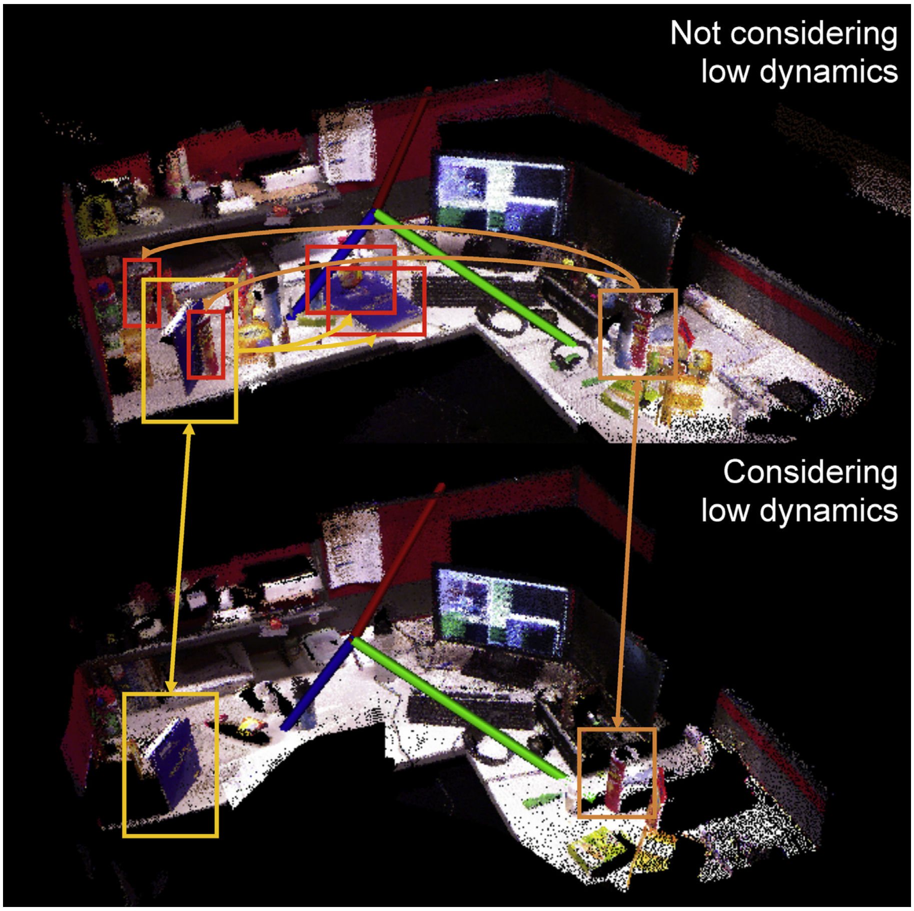


Fig. 1. A comparison of the reconstructed low dynamic environment in point cloud with 10 sessions mapping using multi-session SLAM without considering the low dynamics (top) and proposed framework considering the low dynamics (bottom). One can see that the book, box, bottles and plastic bags are repeated, making the scene with incorrect duplicated information. The book and the chip can are highlighted using light and dark orange rectangles. Their out-of-dated positions are highlighted using red rectangles. The arrows demonstrate the correspondence.

information for better poses configuration estimation. So in these works, anchor nodes [9], weak links [1,6] are introduced to solve the problem. The vision based SLAM in low dynamic environment has also been studied. In Ref. [10], multiple poses formed a view cluster, in which the images with the similar view would be updated over time. This method can tell whether a frame is out-of-dated but cannot show which part has been changed as it was a sparse visual feature based method.

Most existing methods dealing with SLAM in dynamic environment is based on 2D laser SLAM. In Refs. [11,12], the set of scans in global coordinates was updated by sampling after each new session to build an in-dated map. In their work, poses were estimated by SLAM at the first session. For the later sessions, the poses were estimated by localization, not included in the SLAM framework. In Refs. [8,13], both works described the dynamic with each cell in grid occupancy map having an independent Markov Model. In Ref. [7], a dynamic environment map was modeled as a pose graph. After each session, the out-of-dated poses are identified and removed based on 2D occupancy grid map built from the laser data. In Ref. [14], the poses related to the low dynamics were removed to enhance the robust of the optimizer.

In the context of RGBD SLAM, most works apply the graph model, followed by a global optimization backend. In Ref. [15], both visual features and depth information are employed to form an edge in the pose graph. Besides the formulation, an environment measurement model was proposed for pose graph edge selection in Ref. [3]. In Ref. [4], a dense visual odometry is used as frontend to formulate the pose graph, which is more accurate than sparse feature based visual odometry. In Ref. [5], non-rigid deformation is combined with the pose graph optimization for globally consistent dense map, which takes the map mesh into consideration. Extension of these RGBD SLAM systems to multi-session can be achieved by applying the methods developed in Refs. [1,6,9]. But the detection of dynamics by simply using the laser based method is difficult, as the methods employed an occupancy grid map for information fusion and de-noise. When it comes to the case of RGBD sensor, the 3D occupancy grid map is intractable due to the high complexity. So methods should be developed on the raw sensor data, making the problem more challenging.

Besides the mechanism for dealing with dynamic environment, a framework for RGBD SLAM also needs node pruning to keep the computational complexity noncumulative.

The objective of graph pruning is to relate the number of nodes to the size of mapping area instead of the trajectory. In Ref. [16], a reduced pose graph was proposed for mapping a large scale multi-session dataset by merging the edges when a loop closure occurred. But this method cannot control the graph size to a pre-defined number. In both Ref. [17] on 2D laser mapping and our recent work Refs. [18,19] on feature mapping, the methods are derived from the information gain of sensor readings. Thus the graph size can be controlled as users want. From the perspective of a framework, a controllable node pruning method compatible to other modules should be designed.

One of the most similar research to the presented framework is [7], where multi-session 2D laser SLAM in low dynamic environments is studied. Their method was different from our work in several aspects: First, we use RGBD sensor which makes out-of-dated scan identification more difficult. As a result, the completed dynamic objects can be captured while in 2D map it is almost impossible. Second, we apply redundant scans identification to keep the size of the map related to the mapping area instead of the size of the mapping session, which will lead to the complexity noncumulative. Third, the initial pose at each session need not to be known in our framework. Another similar work is [20], their work can describe the evolution of the dynamics in a room, but the localization of their system depended on a 2D laser, while ours fully depended on a RGBD sensor. Therefore, their method was not developed in the context of 3D SLAM, thus cannot be applied in a hand-hold or flying scenario.

3. Framework

The system consists of a multi-session SLAM component and a graph management component. The former includes a SLAM frontend and backend, which will be presented in this section, while the latter, out-of-dated and redundant scans identification, as well as marginalization, will be introduced in later sections. Its schematic is shown in Fig. 2. The process in each timestep is: (1) The multi-session SLAM component yields a map using the RGBD sensor data; (2) The out-of-dated scans identification module will identify whether the scans corresponding to past poses are out of date, if so, the nodes will be pruned since they are no longer useful for map building and loop closure; (3) The redundant scans identification module will continue pruning poses if the number of in-dated nodes is higher than a threshold, which is related to the size of the mapping area; (4) The graph is marginalized to reserve the information after the node is pruned, forming an integrated constraint for the next session, which follows the method in Ref. [21]. An illustrative example of the whole process at session t is shown in Fig. 3 when the threshold is set 3.

The map is generated by registering the scans at the corresponding poses capturing them. Therefore, the goal of multi-session SLAM component is to estimate the poses in a global consistent metric using multi-session of RGBD sensor data. The pose graph was employed to represent the map with each node being the pose and edge being the constraint, which is a pose transform between two poses assigned by the sensor data alignment. The conventional SLAM system optimizes the graph to get the configuration of nodes that best fit all the constraints, which are the estimated poses. The multi-session SLAM component will further investigate the sensor data alignment across the sessions, so that the new session can be added into the graph built by the previous sessions, thus leveraging the isolated information into a universal representation for mapping building.

Specifically, the frontend in the multi-session SLAM component is to estimate intra and inter session loop constraints based on the RGBD sensor data. The backend is to perform pose graph optimization. A pose graph is defined as a state vector $\widehat{x}$ and its corresponding information matrix $\widehat{\Omega}$. Each state in the state vector is a pose. We have following notations:

- Denote final pose graph at session $t-1$ as $\widehat{x}^{t-1,p}$ and $\widehat{\Omega}^{t-1,p}$ with K poses. These poses are in previous sessions. When in the first session, this graph is null with $K=0$.
- Denote initial graph at session t as $\widehat{x}^{t,c}$ and $\widehat{\Omega}^{t,c}$ with N poses, where $N > K$. The first K states $\widehat{x}^{t,c}_{1:K}$ are corresponding to the same poses with the states in $\widehat{x}^{t-1,p}$, while last $N{-}K$ poses are added the session t.
- Denote constraints connecting pose i and pose j at session t as $\widehat{x}^{t}_{ij}$ and $\widehat{\Omega}^{t}_{ij}$.

When a new session begins, the new poses will be added to a new pose graph before an inter-session loop closure is found, hence there are two isolated sub-graphs in the pose graph. The loop closure detection follows the method in Ref. [3], but is conducted among poses from previous sessions. If the detected loop closure constraint connecting to the poses in previous session, an inter-session loop closure is found. Then the two isolated sub-graphs are transformed into universal coordinates, the state vector and information matrix are concatenated. Generally, the number of isolated sub-graphs in the pose graph indicates the number of coordinates of the map. Optimization will be applied to each sub-graph. In most cases, there will be only one sub-graph after a session unless the new session is conduct at a new place, leading to no inter-session loop closure found.

To estimate the pose transform for both inter and intra-session constraints, we apply a feature-based alignment followed by a dense ICP alignment. SURF [22] features are extracted and matched for RANSAC based 3D-2D pose alignment [23]. It was used as the initial value for ICP, which in the system is a point to plane EM derived version [24]. In the backend at session t the optimization problem is formulated as

$$\widehat{x}^{t,c} = \operatorname{argmin} \sum_{i,j} \left\| \widehat{x}^{t}_{ij} - f\left(x_i, x_j\right) \right\|^2_{\widehat{\Omega}^{t}_{ij}} + \left\| \widehat{x}^{t-1,p} - x_{1:K} \right\|^2_{\widehat{\Omega}^{t-1,p}}$$

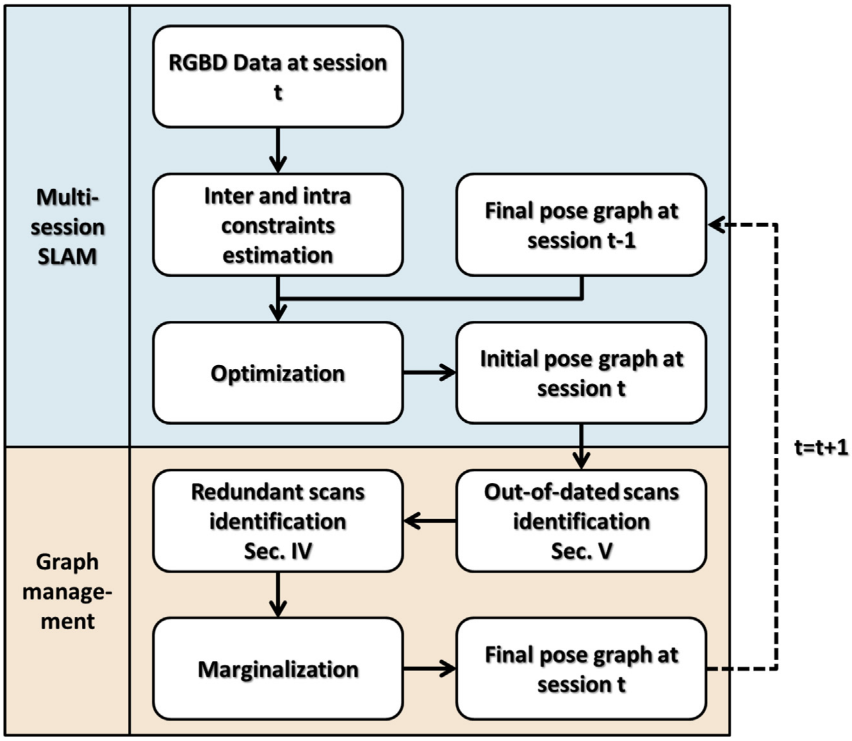

Multi-session SLAM
RGBD Data at session t
Inter and intra constraints estimation
Final pose graph at session t-1
Optimization
Initial pose graph at session t
t=t+1
Graph manage-ment
Redundant scans identification Sec. IV
Out-of-dated scans identification Sec. V
Marginalization
Final pose graph at session t

Fig. 2. The framework of our multi-session RGBD SLAM in dynamic environment system. At each time a new frame comes, the multi-session SLAM component yields a map using the RGBD sensor data, in which the out-of-dated scans are identified and pruned since they are no longer useful for map building and loop closure. Then the redundant scans identification module will continue pruning poses if the number of in-dated nodes is higher than a threshold. Finally, the graph is marginalized to reserve the information after the node is pruned, forming the integrated constraint for the next session.

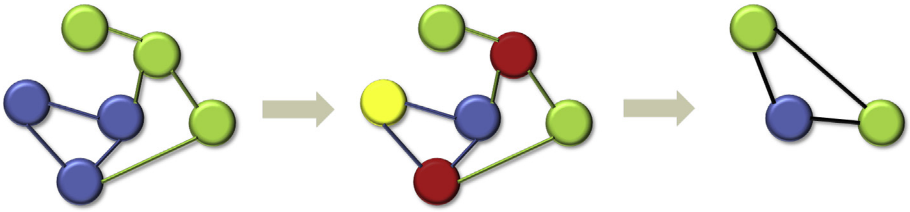


Fig. 3. An example of the procedures in the proposed algorithm. In the left graph, the blue nodes and edges indicate the final pose graph at session $t-1$ with a size of 3, the green nodes and edges are poses and constraints obtained at the session t, the whole graph is the initial pose graph at session t. In the middle graph, the red nodes are identified as redundant nodes, which can be either poses at current session or at previous sessions, the yellow node is identified as out-of-dated node, which can only be the pose at previous sessions. In the right graph, the redundant and out-of-dated nodes are marginalized, forming the final pose graph at session t, also the integrated constraint for the next session, which has the same size to the final pose graph at session $t-1$, the black edges are generated through marginalization.

where x_i is the ith pose, f is a function mapping two poses to their relative pose transform. In the first session, since the second term is null, the equation becomes a standard pose graph SLAM optimization problem. In the later sessions, the constraint in the second term is formed by the final pose graph at last session.

The number of poses in $\hat{x}^{t,c}$ is N, which should be reduced to K in the next session. It is achieved as shown in Fig. 2 through identifying the out-of-dated and redundant scans from $\hat{x}^{t,c}$. By connecting the graph management component to the multi-session SLAM component, the system is able to track the map in time with controlled complexity.

4. Out-of-dated scans identification

In this section, we propose a method for out-of-dated scans identification which can achieve similar results in 3D mapping as using 2D occupancy grid in 2D pruning but computationally more efficient than 3D occupancy grid. Before introducing our out-of-dated scans identification module, we first review the occupancy grid map based method. The occupancy grid map is built by fusing the multiple measurements of grids' status, which are determined by ray casting of each pixel in the scans. There are three status in each grid: occupied, free and unknown. By comparing the status of the grids in occupancy grid map generated by scans of first K poses (from previous sessions) in $\hat{x}^{t,c}$ with the corresponding ones in the map generated by scans of other $N-K$ poses (from current session), the dynamic part can be detected. The rule is very simple that if a pair of grids with one being free (occupied) and the other, occupied (free), then this pair is labeled as change in environment.

Now we return to RGBD scans. First, the point cloud generated by scans of first K poses in $\hat{x}^{t,c}$ is put into a volume, called previous volume (PV). So does one that by scans of other N−K poses, called current volume (CV). The two volumes should be in the same size. Then by comparing PV and CV, we have classifications for grids below:

- Case 1: the grid in CV contains points while the corresponding one in PV, does not.
- Case 2: the grid in CV does not contain points while the corresponding one in PV, does.
- Case 3: the grid in CV and the corresponding one in PV both have point.
- Case 4: neither grid in CV nor the corresponding one in PV has point.

A grid here is a voxel in the volume, containing a cubic space. Note that it is not the same as that in occupancy grid mapping. It is only a container that saves a series of end points lying in its cubic region. Hence no ray casting is conduct. One can see that the change must be contained in the grids belonging to case 1 and case 2. A naive method is to detect the change by simply using the similar rule in grid occupancy map based method, that is to find the grid belonging to case 1 (case 2), we have the result as shown in the upper left graph in Fig. 4.

The poor result is due to the lack of the grid occupancy model, which solves the two problems below implicitly:

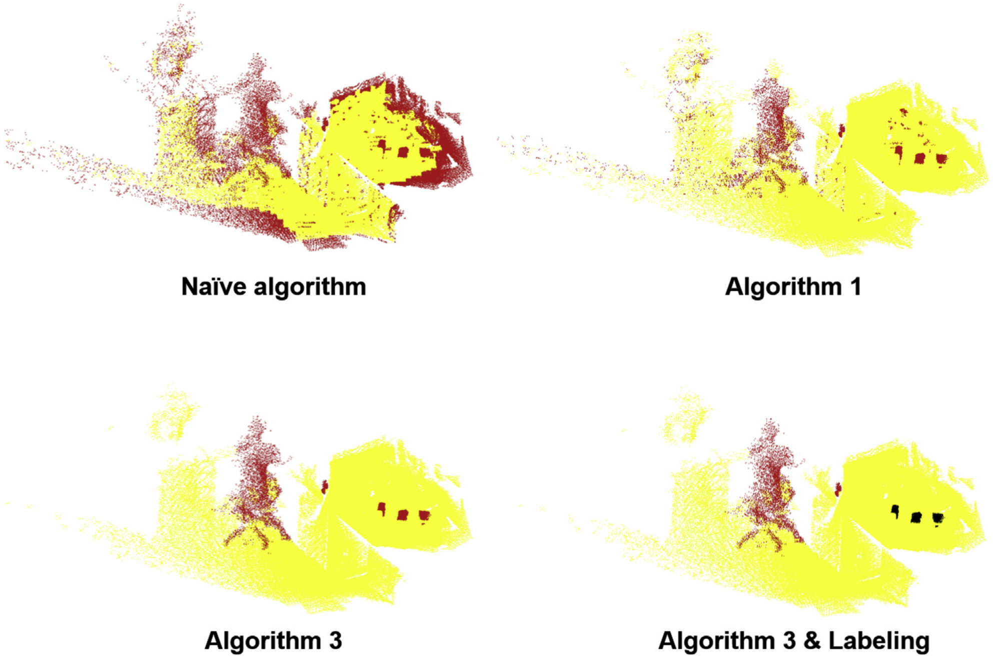
- There is no unknown status in our point cloud volume, so that the part that is not observed during the current session (previous sessions) is regarded equivalent to free status in occupancy grid map. Actually, such part cannot be regarded as dynamic since no information is acquired in the current session (previous sessions).
- The point cloud acquired is of bad quality and no fusion mechanism as grid occupancy map can be applied.

So in our method, we explicitly employ a measurement model to identify which part is out-of-dated or not sensed, whose input is clusters of potential point cloud that is in case 1 and case 2, hence the number of measurement model applied can be reduced, and more robust to noise. In sequel, the method will be introduced step by step to show it clearer.

In grid occupancy map, if a grid has a status of unknown, it means there is no beam traversing that grid. If a grid is free, it means there is a beam traversing and passing through that grid. This indicates that a sensor measurement model is applied during the map building. In grid occupancy map, this model is applied implicitly using the ray casting when a new scan is registered into the map. Inspired by this insight, a camera projection model is applied explicitly to those points in grid belonging to case 1 and case 2. The measurement model is as follows,

$$u = PR^T(p - t)$$

where P is camera intrinsic matrix, R and t is the pose, p is the point. The third entry of u, $u(3)$, is the depth of p from this pose. At the same time, we have the real measurement d in the pixel ($u(1)/u(3)$, $u(2)/u(3)$) of the depth image. If d is smaller than $u(3)$, then this point is occluded from this pose, thus no information is acquired. If d is larger than $u(3)$, then it means from this pose the point should have been observed but actually is not observed. The only reason is that this point is removed when measurement is taken in this pose, which gives the cues to the change in environment. The algorithm is shown


Naïve algorithm

Algorithm 1

Algorithm 3

Algorithm 3 & Labeling

Fig. 4. Detection result using naive method (top left), Algorithm 1 (top right) and Algorithm 3 (bottom). The point cloud corresponding to dynamic part in red and static part in yellow. In the lower right one, the added part is in red and the removed part is in black.

in Algorithm 1 where ε is a parameter for tolerance of depth noise.

Algorithm 1. RGBD sensor measurement model.

```
Data: Point $p$, Pose $(R,t)$, Depth image $D$
Result: $Status$
If $(u(1)/u(3), u(2)/u(3))$ is valid in $D$ then
    $d = D(u(1)/u(3), u(2)/u(3))$
    If $d \geq u(3) + \epsilon$ then
        //Current depth is significantly larger than the predicted one
        $Status = dynamic$
    end
end
```

An illustration of the model is shown in Fig. 5. Given Pose *i* in previous sessions and Pose *j* in the current session, the points with black boundary are seen by Pose *j*. Note that Point A cannot be seen by Pose *j* because its projection is out of the field of view (FoV) of Pose *j*. Point B and Point C are seen by both poses. For Point D, it is projected into the FoV of Pose *j*, but it is occluded by Point B, so the projected depth will be evidently larger than the real depth value. Hence it is correct for Point D that cannot be seen by Pose *j*. When it comes to Point E, the ray from Pose *j* in this direction pierces it, which gives that the projected depth is obviously smaller than the real depth value. This situation only occurs if Point E is absent when the scan is taken at Pose *j*. As a result we can know Point E is a point on the dynamic object.

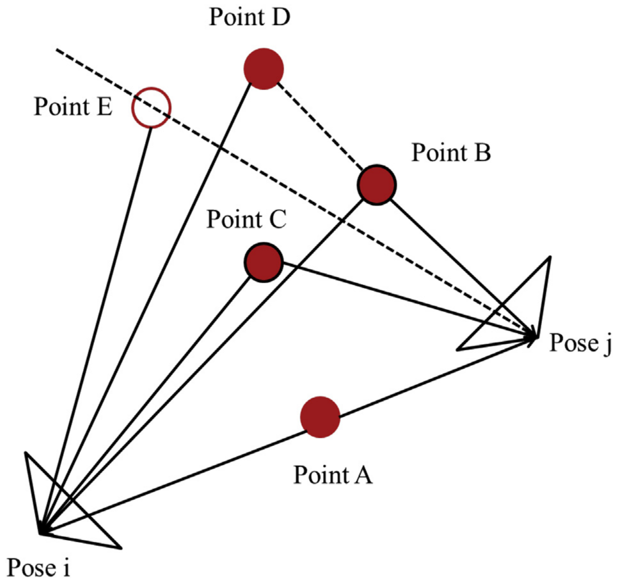


Fig. 5. An example of the scan analysis to decide whether a point can or cannot be seen by a pose. Point A cannot be seen by Pose *j* because its projection is out of the FoV of Pose *j*. Point B and Point C are seen by both poses. Point D cannot be seen by Pose *j* because it is occluded by Point B. Point E can be seen by Pose *j* but is actually not seen because Point E is absent when Pose *j* acquires its observation.

By applying this model to the points in case 1 and case 2, we have result shown in upper right graph in Fig. 4. The result is now much better, but some noise like points is also detected dynamic, which is due to the second reason summarized. In grid occupancy map, the fusion mechanism can reduce the noise effectively. But in our case, there is no fusion mechanism. In addition, the quality of Kinect raw data is worse than that acquired by laser, especially it has holes. So we cannot assume that the grid is independent as [7,8,13]. Since the change in environment is usually in the level of object, the connected component is applied to cluster the grids in case 1 (case 2), resulting in clusters with each one having a series of neighboring grids in the same case, which is now much more object like and more robust to noise. The algorithm to identify a dynamic connected component is shown in Algorithm 2, where t_0 is the threshold to eliminate connected component with few points, t_1 is the threshold to eliminate connected component that has little evidence support it is dynamic.

Algorithm 2. Connected component identification.

```
Data: Component $C$, Poses $\Theta$
Result: $Label$
If $|C| < t_0$ then
    // Small-size component is regarded as noise
    Return
end
// Label all points in the component
For each point $p$ in $C$ do
    For each pose $(R,t)$ in $\Theta$ do
        $Status = \mathbf{RGBDSensorMeasurementModel}(p,(R,t))$
        If $Status == dynamic$ then
            $Add(p) = 1$
            break
        end
    end
end
// A component is dynamic if some of its points are dynamic
If $Mean(Add) > t_1$ then
    $Label = dynamic$
end
```

Put all things together, the proposed out-of-dated scans identification method is shown in Algorithm 3. The result is shown in lower row in Fig. 4, in which one can see the detected change is clear and correctly codes the change part. The main steps of our method is about a traversal of all grids in the volume, two connected components and detection using measurement model in the level of connected component. The points fed to the measurement model step are only a small part of all points. If a 3D occupancy model is applied, the computational burden is about formation of the occupancy volume and a traversal of all grids in the occupancy volume. The formation takes time for ray casting on all pixels (has equal number as points). Besides, ray casting is more time consuming than the simple matrix multiplication. These two factors enables our method more efficient.

Algorithm 3. : Out-of-dated scans identification.

```
Data: Point Cloud $Pt$, Poses $\hat{x}^{t,c}$
Result: $LabelList$
$PPose = \hat{x}^{t,c}_{1:K}$
$CPose = \hat{x}^{t,c}_{K+1:N}$
$CV, PV = PutPointsinVolume(Pt, CPose, PPose)$
$CG = FindFridsinCase1(CV)$
$PG = FindFridsinCase2(PV)$
//Find components in current and previous map as dynamic candidates
$CC = ConnectedComponent(CG)$
$PC = ConnectedComponent(PG)$
//If the components in the current session cannot be seen in the last session, it //is a
dynamic object (added)
For each component $C$ in $CC$ do
    $Label = ConnectedComponentIdentification(C, PPose)$
    If $Label == dynamic$ then
        $LabelList \leftarrow added$
    end
end
//If the components in the last session cannot be seen in the current session, it //is a
dynamic object (removed)
For each component $C$ in $PC$ do
    $Label = ConnectedComponentIdentification(C, CPose)$
    If $Label == dynamic$ then
        $LabelList \leftarrow removed$
    end
end
```

5. Redundant scans identification

The input to this module is the set of in-dated scans. If the number of such scans is still higher than a threshold, the redundant scans identification module will select the poses to be pruned furthermore as shown in Fig. 2. So this module guarantees the size of the final graph at this session is bounded, thus the key factor enabling the noncumulative complexity. The method is to find a subset of poses generating a map close to the one generated by the full pose set. As this is an NP-hard problem, we instead use a greedy strategy to select one pose at a time. In this subsection we introduce a pose pruning algorithm that will generate a map close to the original one in KL divergence. The problem is stated as follows

$$\hat{z}_j = \text{argmin} \sum_i D_i(z_j)$$

$$D_i(z_j) = \sum_{m_i=0,1} \ln \frac{p(m_i|Z)}{p(m_i|Z-z_j)} p(m_i|Z)$$

where z_j is a scan transformed into the global coordinates using the optimized pose $\hat{x}^{t,c}_j$. Z is the set of all such global coordinated scans. Z—z_j is a subset set of all scans except z_j. The ith grid in the volume obtained from the out-of-dated scans identification module has m_i valued 1 or 0 to indicate the grid is occupied or not. Each point in the grid is regarded as a positive observation meaning that this grid is occupied. Thus the idea behind is to find a subset of scans that generate a volume that has similar occupancy to the original one.

Different from the grid occupancy mapping based method [17], our method uses the model only considering the end point of a beam, so that the volume obtained during out-of-dated scans identification can be employed directly in this step. The expensive computation of ray casting to build 3D occupancy grid map is also avoided. In this model, there will be no negative observations, which gives information that a grid is unoccupied. For the ith grid, the pth positive observations in the scan $\hat{z}_j$ is denoted as $\{o_{ijp}\}$. Denote $a = p(o_{ijp}|m_i = 1)$ and $b = p(o_{ijp}|m_i = 0)$. By setting a uniform prior to m_i, we have

$$p(m_i = 1|Z) = \frac{p(Z|m_i = 1)}{p(Z|m_i = 1) + p(Z|m_i = 0)} = \frac{a^{\sum_j |o_{ijp}|}}{a^{\sum_j |o_{ijp}|} + b^{\sum_j |o_{ijp}|}}$$

$$p(m_i = 0|Z) = \frac{p(Z|m_i = 0)}{p(Z|m_i = 1) + p(Z|m_i = 0)} = \frac{b^{\sum_j |o_{ijp}|}}{a^{\sum_j |o_{ijp}|} + b^{\sum_j |o_{ijp}|}}$$

Denote $n = \sum_j |o_{ijp}|$, leading to

$$D_i(z_j) = |o_{ijp}| \frac{a^N \ln a + b^N \ln b}{a^N + b^N} + \ln \frac{a^{N-|o_{ijp}|} + b^{N-|o_{ijp}|}}{a^N + b^N}$$

which measures the information contribution of a scan. This measure can be used to find a subset of pose generating the map with minimal information loss. Through repeating this procedure, the number of reserved poses can be equal to the threshold. This is the crucial part of the framework to achieve a noncumulative complexity.

The low dynamic environment includes the static environment as a special case. When a robot executes a multi-session SLAM in a fixed sized static environment, the robot has loop closures all the time. If no extra pruning technique is employed, the size of the graph will keep growing as all scans are in-dated in such environment without change. Besides the environmental dynamics, this example also shows that in long term there exists redundancy due to continuous re-visiting of a mapped static area even in a low dynamic environment.

6. Experimental results

In this section, we demonstrate the performance of the proposed algorithm using dataset collected from the real world. There are three steps to evaluate the performance of the framework. We firstly show the effectiveness of redundant scans and out-of-dated scans identification by comparing them with other algorithms. The framework on the top of the two algorithms is evaluated on a 2-session dataset to illustrate the process of the proposed framework. The parameters are also tuned on this dataset. Secondly, with the best parameters, the framework is validated on a 10-session dataset to show the performance. Both the 2-session and 10-session datasets are collected using a hand hold Kinect in our laboratory, so that this step is a split dataset testing. Thirdly, to further test the performance, we collect another 5-session dataset from workspace of an industrial robot using Kinect II in another country, which is totally blind to our development of algorithm. This external dataset is expected to show the real performance of the proposed framework. The selected parameters are demonstrated in Table 1.

The laboratory, which generates the 2-session and 10-session datasets, is a typical workspace environment shared by the human and robots. The workspace in the experiment is a test bench for service robot, in which the objects are added, removed and moved by both service robot and human frequently. The workspace of the industrial robot is arranged like a factory environment. There are boxes with various sizes manipulated by the robot and human across the time. The map cannot tell the current status if it is not updated, thus confusing the robot to do the task. Besides, the target and self-localization of the robot can both be affected if the out-of-dated images or point cloud provide out-of-dated clues. These problems can be solved if the proposed mapping system can identify the dynamics and keep the map in track of the environment.

6.1. Redundant scans identification result

The objective of the redundant pose identification module is to cover the volume as much as possible using a fixed number of poses. Treat the original volume before pruning as a binary labeled volume, classified by whether a grid in the volume has point. Then the volume built by the poses has three cases:

- Case 1: the grid in the original volume has point, while the corresponding one, not.
- Case 2: the grid in the original volume and pruned volume both have point.
- Case 3: neither grid in the original volume nor the pruned volume has point.

Now we can define the measure of coverage as #*case* 2/ #*case* 1 + #*case* 2. We select the pose set in the 2 sessions from our 10 sessions to show the ratio of the method. For comparison, a random pruning method is used. The result is shown in Table 2, where the column in the left means the size of the final pose graph and RSI indicates the proposed

Table 1
The parameters used in the experiments.

Parameter	Value	Meaning
ε	0.05	Tolerance of depth difference
K	30	Number of poses in the final graph
Grid size	0.02	Size of the grid in volume
t_0	25	Min. number of points in a component
t_1	0.3	Min. percentage of points in a dynamic object

Table 2
Comparison on coverage measure. Bold indicated that, best performance in the corresponding configuration.

Subset/total set	RSI	Random mean	Random std.
10/68	**0.8246**	0.5064	0.0652
20/68	**0.9311**	0.7109	0.0489
30/68	**0.9669**	0.7956	0.0685
40/68	**0.9835**	0.8895	0.0388
50/68	**0.9926**	0.9294	0.0253
10/56	**0.7931**	0.5028	0.0608
20/56	**0.9141**	0.7266	0.0443
30/56	**0.9759**	0.8168	0.0548
40/56	**0.9937**	0.9159	0.0396
50/56	**0.9991**	0.9649	0.0241

redundant scans identification. One can see that for the number of poses more than 30 after pruning, the performance of coverage is more than 95 percent, which means covering almost the whole map using half of the poses when proposed method is applied. So the size of final pose graph at each session is set 30.

6.2. 2-Session result

In this experiment, the dynamics between two sessions including: a bottle and a mug are removed, a box is moved and a sitting person appears in the second session. In Fig. 6, a glimpse of the scene in each session is shown. One can see the difference mentioned above. The result of out-of-dated scans identification is shown in the lower right figure in Fig. 4, in which the bottle and mug are removed, the moved boxes, and the newly appeared sitting person are updated.

The dense map generated by multi-session pose SLAM using all information without pruning is shown in left image in Fig. 7. One can see that all objects appearing on the desk (bottle, mug and two duplicated boxes indicated by yellow circle). The map in the right image in Fig. 7 using the proposed method keeps track of the current environment as the figures in Fig. 6.

6.3. Out-of-dated scans identification result

To evaluate the performance of the out-of-dated scans identification, we compare the proposed algorithm with 3D occupancy grid map based algorithm, which is a direct extension of 2D occupancy grid map in Ref. [7], on the 10-session dataset. Between two consecutive sessions, an event is defined as adding or removing an object in the scene. Moving an object from one place to another in the scene consists of two events. There are in total 42 dynamic events in the dataset. An identification is defined as a component was labeled as dynamic. If the component is corresponding to the real dynamic event, the identification is defined as true positive. The precision and recall are defined as the ratio of the number of true positive over the number of identification and the number of events respectively. The computational time is included as an indicator of efficiency. The results were calculated based on the dynamic events across all sessions.

In Table 3, one can see that the proposed method outperforms the 3D occupancy grid based method in both precision and recall. The main reason is that the grids in occupancy grid map are regarded equally. When two occupancy grid

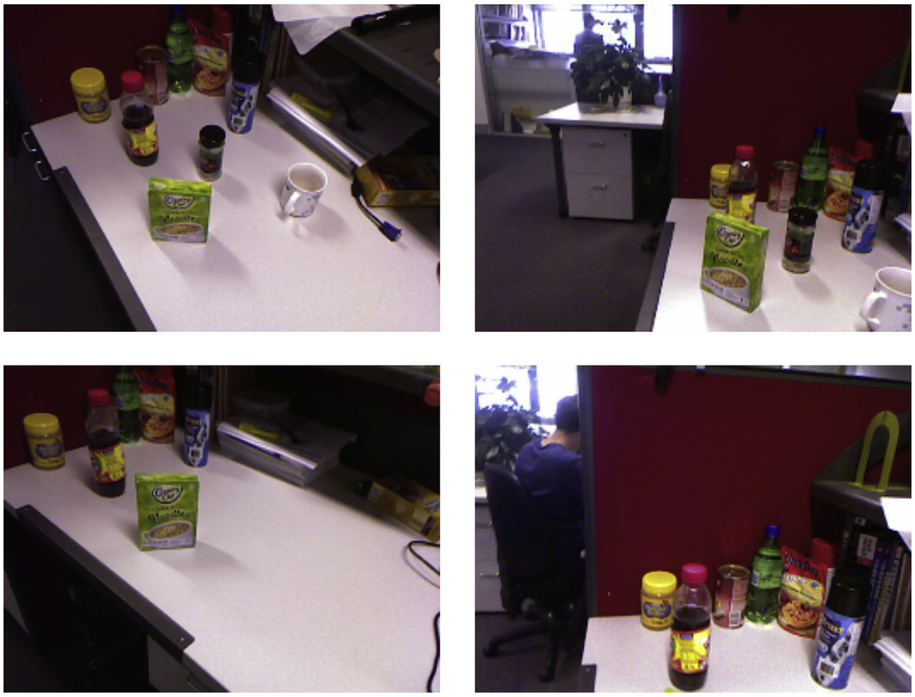


Fig. 6. A glimpse of the scene. The upper row shows the scene in the first session while the lower row, the second session. One can see the bottle in the middle and the mug are removed in the second session. The yellow box at right is moved in the second session. Besides, a people sitting at the next desk appears in the second session.

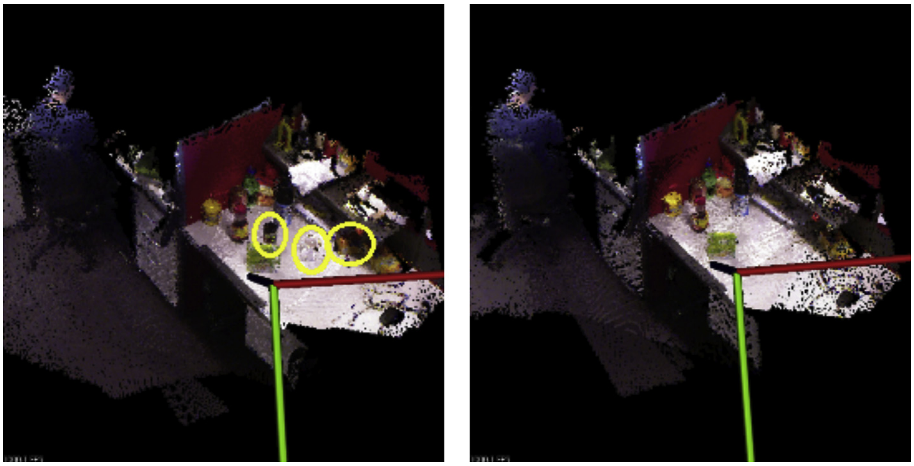


Fig. 7. The left one is the map reconstructed by multi-session SLAM using all frames without considering the low dynamics. The right one is the map reconstructed by proposed method.

maps are compared to derive the dynamics, the tolerance of noise for each grid is the size of grid, which can be very sensitive to the noisy RGBD sensor measurement, especially when the depth is large due to the increasing uncertainty with respect to growing depth. If the grid size is enlarged to increase the tolerance of noise, the resolution degenerates. However, in our model, the points in grids are projected back to the depth image plane, the tolerance of noise then can be modeled in the image plane, which is decoupled from the size of grid, hence more appropriate for the RGBD sensor. Besides, the 85.4 times faster computation, which may be argued by implementation details, can at least show that our method was much more efficient by replacing the expensive ray casting with matrix multiplication, validating our hypothesis in Section 4.

6.4. 10-Session result

For quantitative comparison, three frameworks are tested in this experiment including:

- **No Pruning** multi-session pose graph SLAM using all information without any pruning (The best pose estimation one can achieve. So it is used as benchmark).
- **Framework I** out-of-dated pose pruning + marginalization.
- **Framework II** out-of-dated pose pruning + redundant pose pruning + marginalization.

Table 3
Comparison on identification of dynamic events.

Indicators	Proposed	Occupancy grid [7]
#Identifications	34	31
#True positives	**29**	24
Precision	**0.853**	0.774
Recall	**0.690**	0.571
F-measure	**0.763**	0.632
Relative time	1×	85.4×

To evaluate the performance, the last two schemes are compared with the first scheme in a 10-session SLAM in a low dynamic environment. The objects on the desk are added, removed or moved during sessions. Between some consecutive sessions, the map is set static, to show the difference between Framework I and Framework II. The map reconstructed after 10 sessions by SLAM using all information is shown in Fig. 1, in which one can see that some objects appear more than one times. Actually, the number of each object is exactly one. So the map is hard to provide the accurate information of the environment, not appropriate for some grid or octree based localization and navigation techniques. The result after 10 sessions using proposed framework is shown in Fig. 1, where each object appears in its final position.

The size of the pose graph after each session kept in the system is shown in Fig. 8. There is no dynamics between session 8 and 9 as well as session 1 and 2. Then in Fig. 8, the size of the graph for Framework I during these sessions has the same increasing trend as that of No Pruning, which indicates that the Framework I will degenerate to No Pruning if the environment is static. However, Framework II works in both low dynamic and static environment, since its size keeps bounded during the 10 sessions. For image based localization techniques, all these schemes work since they save the images in the current session, but the Framework II has the smallest search space because of the smallest graph size.

The time applied for graph optimization is shown in Fig. 8. Note that the size of the graph for optimization is different from the size shown in Fig. 8, which is the size after pruning. But one can see that the size is still bounded for Framework II both in environment with and without changes.

To evaluate the accuracy, the relative translational and rotational difference (RTD and RRD) similar to [25] is employed compared to the No Pruning framework. The result is shown in Fig. 9. In this dataset, one can see that the difference is in level of millimeter. Here we will not state which one is more accurate. The main thing we want to show is that the error level after 10 sessions is still acceptable.

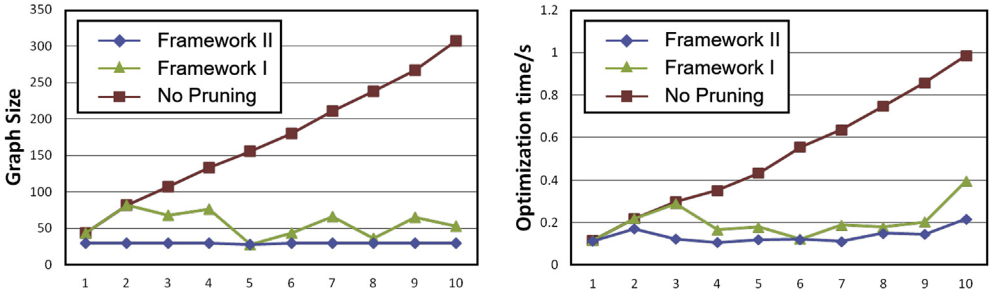


Fig. 8. Evolution of the graph size (left) and computational time for optimization (right) during sessions.

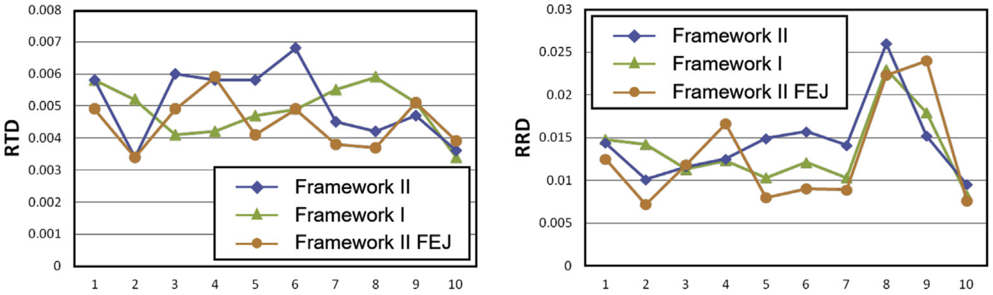


Fig. 9. Evolution of relative pose accuracy measures on translation (left) and rotation (right) during sessions.

6.5. External 5-session dataset result

The No Pruning and Framework II are applied on the external 5-session dataset for further performance evaluation. As in the 10-session result, the graph size, computational time, RRD and RTD are the indicators of the performance. There are environmental changes after each session. Between first and second sessions as well as third and fourth sessions, the identified dynamics are zoomed up and demonstrated in Fig. 10. One can see there is no false alarms. However, between third and fourth sessions, the removal of the control panel of the industrial robot is not identified. The evolution of graph size and the computational time for optimization are shown in Fig. 11. The graph size of the No Pruning keeps monotonically increasing. The gap between No Pruning and Framework II are also growing, indicating that the trend of No Pruning cannot be controlled while the maximum size of Framework II is below 30, as desired by the pre-defined parameters. For the computational time, both frameworks keep increasing, while No Pruning grows faster. In theory, the computational time complexity for Framework II is with respect to the sum of the graph size in previous sessions and current session, hence constant. The ascent of the computational time in the last 1 session is due to the large number of poses in the last 1 session. In long term, it is guaranteed to keep stable since the size of graph has been stable after session 3, and the number of poses in a session is bounded. In Fig. 12, the difference in both translation and rotation are noncumulative as tested in 10-session dataset. Note that the error level in translation is in millimeter, in accordance with that in 10-session dataset too. Therefore, the external dataset further validates the proposed framework with its similar performance to that on split dataset.

6.6. Discussion

In both laboratory and workspace of industrial robot, there are three kinds of typical failures for out-of-dated scans identification: (1) The small objects are captured by the RGBD sensor with low quality, which is regarded as noise by the algorithm. This failure may be solved by adding the resolution of grid. (2) The two objects are regarded as one dynamic object by the algorithm. This problem cannot be solved if no high level object segmentation or detection of the object is considered. (3) The object is removed, but the new object is added in the same position. Such change cannot be reflected by the geometric shape. This problem calls for more information from appearance clues, such as vision and semantics. However, the last two potential improvements are out of the scope in this paper.

The error in the marginalization comes from the error in value where Jacobian is estimated [26] and potential loop closures a pruned pose will have in the later sessions. To decrease the first kind of error, we include Framework II FEJ (first estimate Jacobian) into the comparison, which is similar to Framework II except that it estimates the Jacobian using the values when they are marginalized. Theoretically speaking, Framework II FEJ will be the best. However, the experimental results shown in Fig. 9 do not demonstrate such a trend. The main reason we think is that: firstly the pose is usually mature when it is marginalized since it is estimated using an optimization of one or more sessions. Secondly a pose does not exist in the graph for a long time since the environment is dynamic, so the error may not be accumulated. For the second kind of error, it focuses on whether a pruned pose has loop closures in the future actually. It is possible that the pruned poses due to observing the dynamic do not have future loop

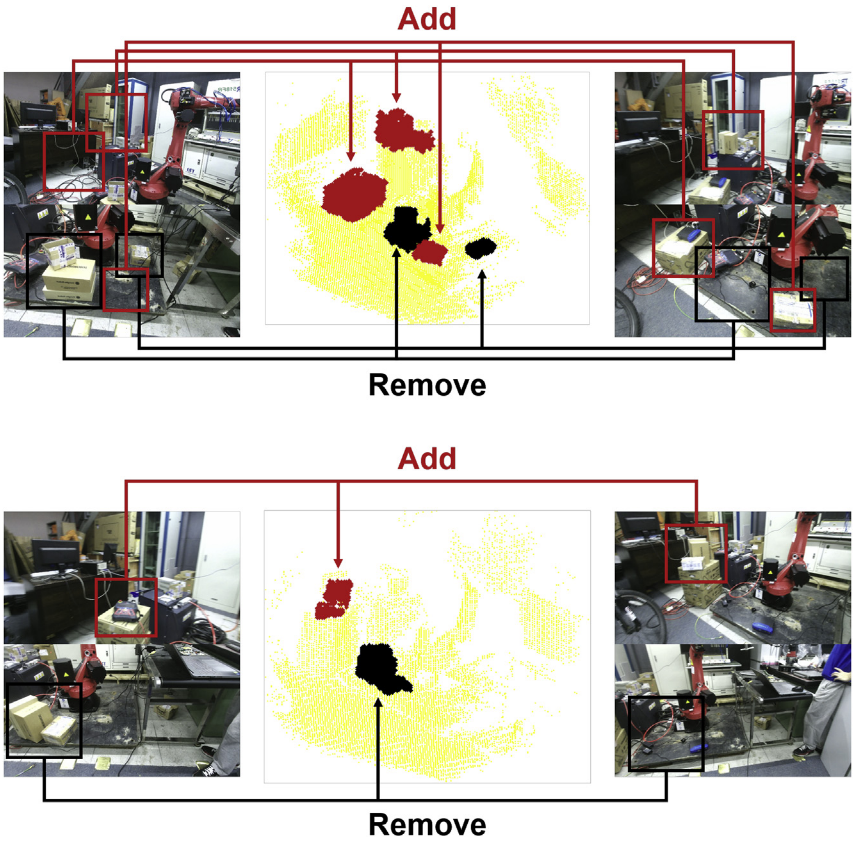


Fig. 10. The identified dynamics, adding in red and removing in black between first and second session (top) as well as third and fourth session (bottom). The correspondence are illustrated by lines and arrows. The yellow points indicated for static part.

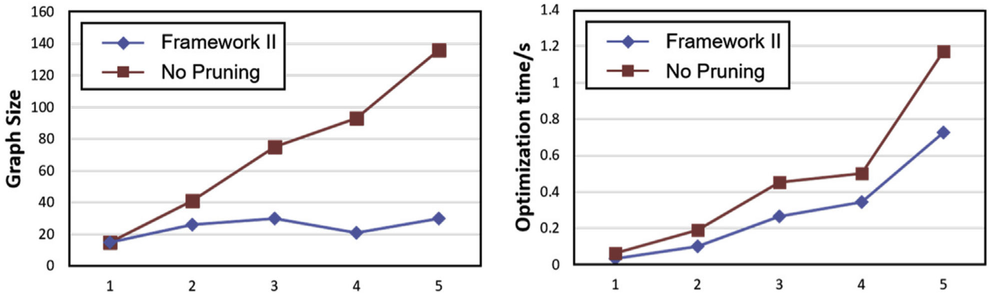


Fig. 11. Evolution of the graph size (left) and computational time for optimization (right) during sessions in external dataset.

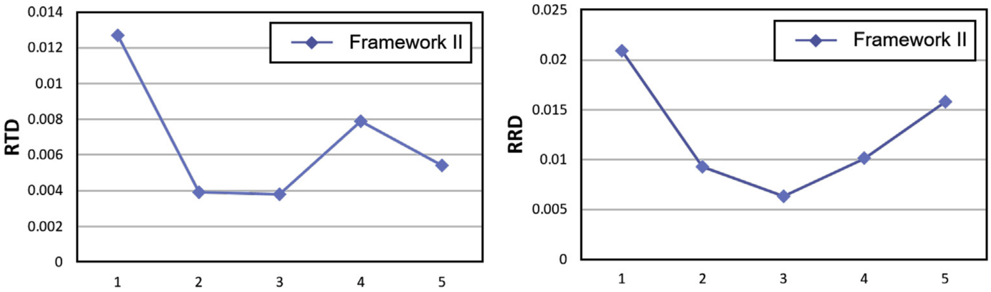


Fig. 12. Evolution of relative pose accuracy measures on translation (left) and rotation (right) during sessions in external dataset.

closures. For the poses marginalized due to redundancy, there are other poses at the similar place still existing in the graph. Thus missing potential loop closures does not have obvious effects on the result.

7. Conclusion and future works

In this paper, the problem of multi-session RGBD SLAM in low dynamic environment has been studied in the context of pose graph SLAM. We propose a framework by equipping the multi-session SLAM system with an out-of-dated scans identification module and a redundant scans identification module to achieve noncumulative complexity both in dynamic or static environment. Finally, the experiments on the split 10-session dataset and external 5-session dataset collected from real world demonstrate and validate the correctness and effectiveness of our method.

To increase the efficiency of optimization, the method in Ref. [27] can be applied to sparsify the information matrix. For large scale mapping in dynamic environment, we want to achieve it by a submap joining technique. For each submap, the computation would be efficient. Besides, a comparison will be better if the ground truth is available as benchmark. Since we have not seen a dataset on multi-session low dynamic environment, we are planning to calibrate a Kinect with motion capture to collect and publish such a dataset.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. NSFC: 61473258, U1509210), and the Joint Centre for Robotics Research (JCRR) between Zhejiang University and the University of Technology, Sydney.

References

[1] Paul Newman, Gabe Sibley, Mike Smith, Mark Cummins, Alastair Harrison, Chris Mei, Ingmar Posner, et al., Int. J. Robot. Res. 28 (11—12) (2009) 1406—1433.
[2] Ayoung Kim, Ryan M. Eustice, IEEE Trans. Robot. 29 (3) (2013) 719—733.
[3] Felix Endres, Jurgen Hess, Jurgen Sturm, Daniel Cremers, Wolfram Burgard, IEEE Trans. Robot. 30 (1) (2014) 177—187.
[4] Christian Kerl, Jurgen Sturm, Daniel Cremers, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2013, pp. 2100—2106.
[5] Thomas Whelan, Michael Kaess, John J. Leonard, John McDonald, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),, IEEE, 2013, pp. 548—555.
[6] Kurt Konolige, James Bowman, J.D. Chen, Patrick Mihelich, Michael Calonder, Vincent Lepetit, Pascal Fua, Int. J. Robot. Res. (2010).
[7] Aisha Walcott-Bryant, Michael Kaess, Hordur Johannsson, John J. Leonard, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),, IEEE, 2012, pp. 1871—1878.
[8] Gian Diego Tipaldi, Daniel Meyer-Delius, Wolfram Burgard, Int. J. Robot. Res. 32 (14) (2013) 1662—1678.
[9] John McDonald, Michael Kaess, C. Cadena, José Neira, John J. Leonard, Robot. Auton. Syst. 61 (10) (2013) 1144—1158.
[10] Kurt Konolige, James Bowman, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, IROS 2009, IEEE, 2009, pp. 1156—1163.
[11] Peter Biber, Tom Duckett, in: Robotics: Science and Systems, 2005, pp. 17—24.
[12] Peter Biber, Tom Duckett, Int. J. Robot. Res. 28 (1) (2009) 20—33.
[13] Jari Saarinen, Henrik Andreasson, Achim J. Lilienthal, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2012, pp. 3489—3495.
[14] Donghwa Lee, Hyun Myung, Sensors 14 (7) (2014) 12467—12496.
[15] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, Dieter Fox, Int. J. Robot. Res. 31 (5) (2012) 647—663.
[16] Hordur Johannsson, Michael Kaess, Maurice Fallon, John J. Leonard, in: 2013 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2013, pp. 54—61.
[17] Henrik Kretzschmar, Cyrill Stachniss, Int. J. Robot. Res. 31 (11) (2012) 1219—1230.
[18] Yue Wang, Rong Xiong, Qianshan Li, Shoudong Huang, in: 2013 European Conference on Mobile Robots (ECMR), IEEE, 2013, pp. 32—37.
[19] Yue Wang, Rong Xiong, Shoudong Huang, Adv. Robot. 29 (10) (2015) 683—698.
[20] Rares Ambrus, Nils Bore, John Folkesson, Patric Jensfelt, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), IEEE, 2014, pp. 1854—1861.
[21] Nicholas Carlevaris-Bianco, Michael Kaess, Ryan M. Eustice, IEEE Trans. Robot. 30 (6) (2014) 1371—1385.
[22] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, Comput. Vis. Image Underst. 110 (3) (2008) 346—359.
[23] Vincent Lepetit, Francesc Moreno-Noguer, Pascal Fua, Int. J. Comput. Vis. 81 (2) (2009) 155—166.
[24] Yue Wang, Rong Xiong, Qianshan Li, Int. J. Robot. Automat. 28 (3) (2013).
[25] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, Daniel Cremers, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2012, pp. 573—580.
[26] Guoquan Huang, Anastasios I. Mourikis, Stergios I. Roumeliotis, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2011, pp. 65—72.
[27] Guoquan Huang, Michael Kaess, John J. Leonard, in: 2013 European Conference on Mobile Robots (ECMR), IEEE, 2013, pp. 150—157.