

# Multi Robot Object-based SLAM

Siddharth Choudhary<sup>1</sup>, Luca Carlone<sup>2</sup>, Carlos Nieto<sup>1</sup>, John Rogers<sup>3</sup>,  
Zhen Liu<sup>1</sup>, Henrik I. Christensen<sup>1</sup>, and Frank Dellaert<sup>1</sup>

<sup>1</sup> Institute for Robotics and Intelligent Machines,  
Georgia Institute of Technology

<sup>2</sup> Laboratory for Information and Decision Systems,  
Massachusetts Institute of Technology

<sup>3</sup> Army Research Lab

**Abstract.** We propose a multi robot SLAM approach that uses 3D objects as landmarks for localization and mapping. The approach is fully distributed in that the robots only communicate during rendezvous and there is no centralized server gathering the data. Moreover, it leverages local computation at each robot (e.g., object detection and object pose estimation) to reduce the communication burden. We show that object-based representations reduce the memory requirements and information exchange among robots, compared to point-cloud-based representations; this enables operation in severely bandwidth-constrained scenarios. We test the approach in simulations and field tests, demonstrating its advantages over related techniques: our approach is as accurate as a centralized method, scales well to large teams, and is resistant to noise.

## 1 Introduction

The deployment of multiple cooperative robots is an important asset for fast information gathering over large areas. In particular, multi robot SLAM, i.e., the cooperative construction of a model of the environment explored by the robots, is fundamental to geo-tag sensor data (e.g., for pollution monitoring, surveillance and search and rescue), and to gather situational awareness. In this paper we are interested in the case in which the robots operate under severe bandwidth constraints. In this context, the robots have to reconstruct a globally-consistent map by communicating a small amount of information among the teammates.

Dealing with bandwidth constraints is challenging for two reasons. First, most approaches for multi robot SLAM imply a communication burden that grows quadratically in the number of locations co-observed by the robots [1]; these approaches are doomed to quickly hit the bandwidth constraints. In our previous works [2, 3] we alleviated this issue by proposing an approach, based on the distributed Gauss-Seidel method, which requires linear communication. The second issue regards the communication cost of establishing loop closures among robots. When the robots are not able to directly detect each other, loop closures have to be found by comparing raw sensor data; in our setup the robots are equipped with an RGBD camera and exchanging multiple 3D point clouds

quickly becomes impractical in presence of communication bounds. In this paper we address this second issue by using an object-level map representation.

**Related Work.** Traditional approaches for multi robot mapping use low-level primitives like points and lines to model the geometry of the environment [4]; these maps become memory-intensive in long-term operation, contain very redundant information (e.g., use thousands of points to represent a planar surface), and lack semantic understanding, which is a key element in a wide range of tasks (e.g., human robot interaction or manipulation). For these reasons, *semantic mapping* has attracted a conspicuous interest from the community, starting from early papers [5], to more recent works which use object templates [6], door signs [7], or planes [8] for mapping. A recent survey can be found in [9]. *Distributed* estimation in multi robot systems is currently an active field of research, with special attention being paid to communication constraints [10], heterogeneous teams [11] and robust data association [12]. The robotic literature offers distributed implementations of different estimation techniques, including Extended Kalman filters [13], information filters [14], and Gaussian elimination [1].

**Contribution.** In this work we advocate the use of higher-level map representations as a tool to enable operation in bandwidth-constrained multi robot scenarios. Maps augmented with objects provide a number of advantages: objects (including planes and other geometric shapes) can be represented in a compact manner and provide a richer and human-understandable description of the environment. Objects are more discriminative, which helps data association and loop-closure detection. Finally, object representations reduce the computational complexity of SLAM by reducing the number of variables (intuitively, we estimate the pose of few objects rather than the position of several 3D points).

We propose an approach for Multi Robot Object-based SLAM with two distinctive features. The first is the front-end, which performs accurate object detection using *deep learning*. Deep learning provides an effective tool to generalize early work on object-based mapping [6] to a large number of object categories. The second is the back-end, which implements distributed pose graph optimization using the distributed Gauss-Seidel method, described in our previous work [3]. We show that the combination of these two techniques reduces the memory requirement and information exchange among robots, allows accurate and parsimonious large-scale mapping, and scales to large teams.

## 2 Technical Approach

**Problem Statement.** We consider a multi robot system and we denote each robot with a Greek letter, such that the set of robots is  $\boldsymbol{\Omega} = \{\alpha, \beta, \gamma, \dots\}$ . The goal of each robot is to estimate its own trajectory and the pose of a set of objects in the environment, using local measurements, and leveraging occasional communication with other robots. We model each trajectory as a finite set of poses; the pose assumed by robot  $\alpha$  at time  $i$  is denoted with  $\mathbf{x}_{\alpha_i} \in \text{SE}(3)$ ; when convenient, we write  $\mathbf{x}_{\alpha_i} = (\mathbf{R}_{\alpha_i}, \mathbf{t}_{\alpha_i})$ , making explicit that each pose includes a rotation  $\mathbf{R}_{\alpha_i} \in \text{SO}(3)$ , and a position  $\mathbf{t}_{\alpha_i} \in \mathbb{R}^3$ . The trajectory of robot  $\alpha$  is

$\mathbf{x}_\alpha = [\mathbf{x}_{\alpha_1}, \mathbf{x}_{\alpha_2}, \dots]$ . Similarly, we denote with  $\mathbf{o}_{\alpha_k} \in \text{SE}(3)$  the pose of the  $k^{\text{th}}$  object in the coordinate frame of robot  $\alpha$  (Fig. 2).

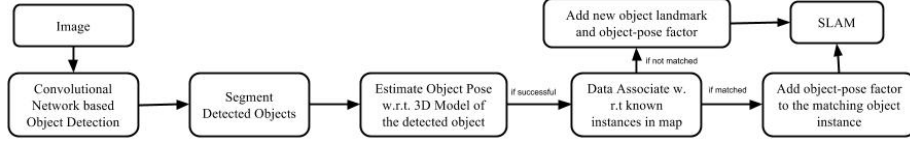


Fig. 1. Flowchart of Object based SLAM

**Object detection and pose estimation.** Each robot collects RGBD data using a depth camera, and measures its ego-motion through wheel odometry. In our approach, each RGB frame (from RGBD) is passed to the YOLO object detector [15], which detects objects at 45 frames per second. Compared to object-proposal-based detectors, YOLO is fast, since it avoids the computation burden of extracting object proposals, and is less likely to produce false positives in the background. We fine-tune the YOLO detector on a subset of objects from the *BigBird* dataset [16]. The training dataset contains the object images in a clean background taken from different viewpoints and labeled images of the same objects taken by a robot in an indoor environment. During testing, we use a probability threshold of 0.3 to avoid false detections.

Each detected object bounding box is segmented using the *organized point cloud segmentation* [17]. The segmented object is matched to the 3D template of the detected object class to estimate its pose. We extract PFHRGB features [18] in the source (object segment) and target (object model) point clouds and register the two point clouds in a Sample Consensus Initial Alignment framework [19]. If we have at least 12 inlier correspondences, GICP (generalized iterative closest point [20]) is performed to further refine the registration and the final transformation is used as the object pose estimate. If less than 12 inlier correspondences are found, the detection is considered to be a false positive. This two-stage process, verifies the detection both semantically and geometrically.

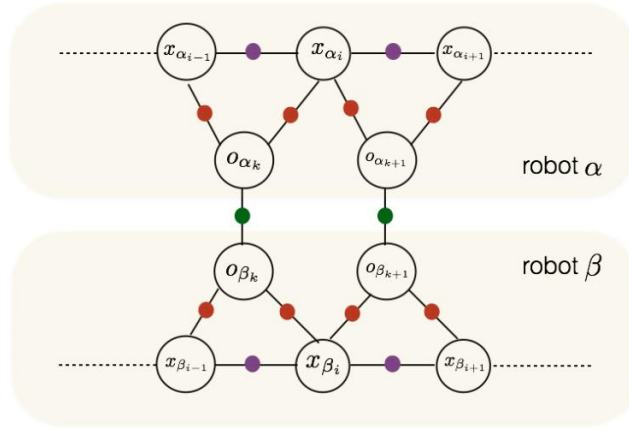
**Object-based SLAM.** If object pose estimation is successful, it is data-associated with other instances already present in the map by finding the object landmark having the same category label within  $2\sigma$  distance of the newly detected object. If there are multiple objects with the same label within that distance, the newly detected object is matched to the nearest object instance. If there exists no object having the same label, a new object landmark is created.

Before the first rendezvous event, each robot performs standard single-robot SLAM using OmniMapper [8]. Both wheel odometry and relative pose measurements to the observed objects are fed to the SLAM back-end, which is based on pose graph optimization [3]. In particular, if object  $k$  at pose  $\mathbf{o}_{\alpha_k}$  is seen by the robot  $\alpha$  from the pose  $\mathbf{x}_{\alpha_i}$ , then an *object-pose* factor is added to the graph:

$$f_{op}(\mathbf{x}_{\alpha_i}, \mathbf{o}_{\alpha_k}, \mathbf{z}_{ik}) \propto \exp\left(-\frac{1}{2}\|\text{Log}(\mathbf{z}_{ik}^{-1}\mathbf{x}_{\alpha_i}^{-1}\mathbf{o}_{\alpha_k})\|_{\Sigma}^2\right)$$

where  $\mathbf{z}_{ik}$  is the relative pose estimate returned by the object-pose estimator described above,  $\Sigma \in \mathbb{R}^{6 \times 6}$  is the corresponding covariance, and  $\text{Log}$  is the logarithm map for  $\text{SE}(3)$ . A flowchart of the SLAM approach is given in Fig. 1.

**Robot Communication.** During a rendezvous between robots  $\alpha$  and  $\beta$ , robot  $\alpha$  communicates the category labels (class) and poses (in robot  $\alpha$ 's frame) of all the detected objects to robot  $\beta$ . We assume that the initial pose of each robot is known to all the robots, hence, given the initial pose of robot  $\alpha$ , robot  $\beta$  is able to transform the communicated object poses from robot  $\alpha$ 's frame to its own frame.<sup>4</sup> For each object in the list communicated by robot  $\alpha$ , robot  $\beta$  finds the nearest object in its map, having the same category label and within  $2\sigma$  distance. If such an object exists, it is added to the list of *shared* objects: this is the set of objects seen by both robots.



**Fig. 2.** Factor graph representation of Multi-Robot Object based SLAM.  $\mathbf{x}_{\alpha_i}$  and  $\mathbf{x}_{\beta_i}$  denote the poses assumed by robot  $\alpha$  and  $\beta$  at time  $i$  respectively. The pose of the  $k^{\text{th}}$  object in the coordinate frame of robot  $\alpha$  and  $\beta$  is denoted with  $\mathbf{o}_{\alpha_k}$  and  $\mathbf{o}_{\beta_k}$  respectively. Green dots shows inter-robot factors whereas orange and purple dots shows intra-robot factors.

The list of shared objects contains pairs  $(\mathbf{o}_{\alpha_k}, \mathbf{o}_{\beta_l})$  and informs the robots that the poses  $\mathbf{o}_{\alpha_k}$  and  $\mathbf{o}_{\beta_l}$  correspond to the same physical object, observed by the two robots. For this reason, in the optimization we enforce that the relative pose between  $\mathbf{o}_{\alpha_k}$  and  $\mathbf{o}_{\beta_l}$  is zero. We do that by adding an *object-object* factor to the pose graph for each pair  $(\mathbf{o}_{\alpha_k}, \mathbf{o}_{\beta_l})$  in the shared object list:

$$f_{oo}(\mathbf{o}_{\beta_l}, \mathbf{o}_{\alpha_k}) \propto \exp \left( -\frac{1}{2} \|\text{Log}(\mathbf{o}_{\beta_l}^{-1} \mathbf{o}_{\alpha_k})\|_{\Lambda}^2 \right)$$

where  $\Lambda \in \mathbb{R}^{6 \times 6}$  specifies the confidence in the data association among the shared set of objects. We remark that, while before the first rendezvous the robots  $\alpha$  and

<sup>4</sup> The knowledge of the initial pose is only used to facilitate data association but it is not actually used during pose graph optimization. We believe that this assumption can be easily relaxed but for space reasons we leave this task to future work.

$\beta$  have different reference frames, the object-object factors enforce both robots to have a single shared reference frame, facilitating future data association.

**Distributed Optimization.** Overall, our approach uses two types of measurements: intra-robot and inter-robot measurements. The *intra-robot measurements* consists of the odometry measurements which constrain consecutive robot poses (e.g.,  $\mathbf{x}_{\alpha_i}$  and  $\mathbf{x}_{\alpha_{i+1}}$ ) and pose-object measurements which constrains robot poses with the corresponding visible object landmarks (e.g.,  $\mathbf{x}_{\alpha_i}$  and  $\mathbf{o}_{\alpha_k}$ ). The *inter-robot measurements* are the ones relating the objects observed by different robots. According to our previous terminology, an inter-robot measurement is associated to each pair in the shared object list. Fig. 2 shows the pose graph containing intra and inter-robot measurements. Given these measurements, we use the distributed Gauss-Seidel (DGS) algorithm [3] to estimate the 3D trajectories of the robots along with the poses of the observed objects.

### 3 Results

We evaluate our approach in large simulations (Section 3.1) and field tests (Section 3.2). The results demonstrate that the proposed approach is accurate, scalable, robust to noise, and requires less memory and communication bandwidth.

We evaluate the accuracy of our approach by comparing it against the standard centralized Gauss-Newton method [21]. In particular, we use three accuracy metrics: (i) the cost at the final estimate, (ii) the average translation error (ATE\*) and (iii) average rotation error (ARE\*) of the robot and landmark poses.

**Average Translation error (ATE\*).** Similar to the formulation by Sturm et al. [22], the average translation error measures the absolute distance between the trajectory and object poses estimated by our approach versus the centralized Gauss-Newton (GN) method. The trajectory ATE\* is defined as follows:

$$ATE^* = \left( \frac{1}{\sum_{\alpha \in \Omega} n_{\alpha}} \sum_{\alpha \in \Omega} \sum_{i=1}^{n_{\alpha}} \|\mathbf{t}_{\alpha_i} - \mathbf{t}_{\alpha_i}^*\|^2 \right)^{\frac{1}{2}} \quad (1)$$

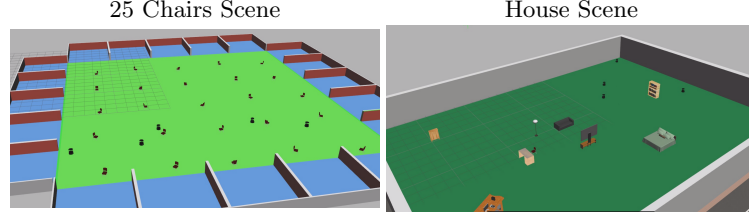
where  $\mathbf{t}_{\alpha_i}$  is the position estimate for robot  $\alpha$  at time  $i$ ,  $\mathbf{t}_{\alpha_i}^*$  is the corresponding estimate from GN, and  $n_{\alpha}$  is the number of poses in the trajectory of  $\alpha$ . A similar definition holds for the object positions.

**Average Rotation error (ARE\*).** The average rotation error is computed by evaluating the angular mismatch between the (trajectory and objects) rotations produced by the proposed approach versus a centralized GN method:

$$ARE^* = \left( \frac{1}{\sum_{\alpha \in \Omega} n_{\alpha}} \sum_{\alpha \in \Omega} \sum_{i=1}^{n_{\alpha}} \|\text{Log}((\mathbf{R}_{\alpha_i}^*)^T \mathbf{R}_{\alpha_i})\|^2 \right)^{\frac{1}{2}} \quad (2)$$

where  $\mathbf{R}_{\alpha_i}$  is the rotation estimate for robot  $\alpha$  at time  $i$ ,  $\mathbf{R}_{\alpha_i}^*$  is the corresponding estimate from GN. A similar definition holds for the object rotations.

Our approach is based on the DGS method and is iterative in nature. Therefore, its accuracy depends on the number of iterations, which in turns depends



**Fig. 3.** Shows the screenshot of 25 Chair and House scenarios simulated in Gazebo.

on the choice of the stopping conditions, see [3] for details. In the following, we present results for two different choices of the stopping condition  $\eta$ .

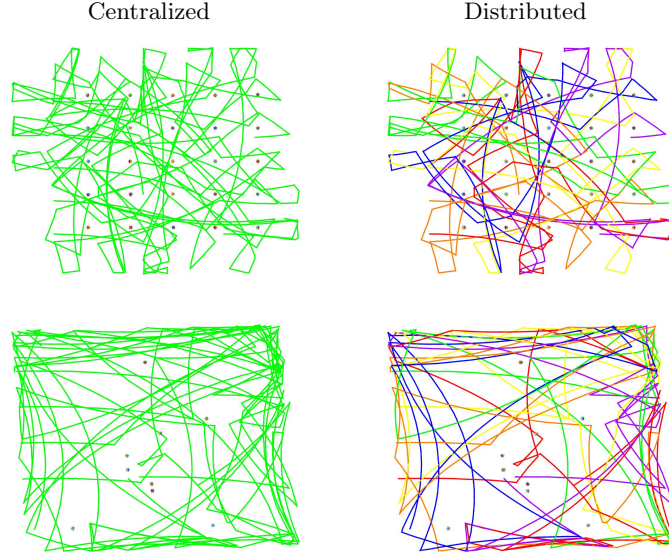
### 3.1 Simulation Experiments

In this section we characterize the performance of the proposed approach in terms of scalability in the number of robots and sensitivity to noise. We test the approach in two scenarios (a) **25 Chairs** and (b) **House**, which we simulated in Gazebo. In the **25 Chairs** scenario, we placed 25 chairs as objects on a grid, with each chair placed at a random angle. In the **House** scenario, we placed furniture as objects in order to simulate an indoor living room environment. Fig. 3 shows the two scenarios. Unless specified otherwise, we generate measurement noise from a zero-mean Gaussian distribution with standard deviation  $\sigma_R = 5^\circ$  for the rotations and  $\sigma_t = 0.2\text{m}$  for the translations. Six robots are used by default. Results are averaged over 10 Monte Carlo runs.

Figs. 4 show the comparison between the object locations and trajectories estimated using multi-robot mapping and centralized mapping for two scenarios. Videos showing the map building for the two scenarios are available at: <https://youtu.be/nXJamypVvY> and <https://youtu.be/nYm2sSHuGjo>.

#Robots	Distributed Gauss-Seidel				Centralized	ATE* (m)		ARE* (deg)	
	$\eta = 10^{-1}$		$\eta = 10^{-2}$		GN Cost	Poses	Lmrks.	Poses	Lmrks.
	#Iter	Cost	#Iter	Cost					
2	5.0	56.1	9.0	56.0	54.7	1.5e-03	8.0e-04	2.1e-01	2.8e-01
4	5.0	118.0	8.0	117.9	113.8	9.7e-04	7.5e-04	2.0e-01	2.8e-01
6	5.0	166.6	7.0	166.5	160.9	3.1e-03	2.1e-03	3.3e-01	4.0e-01

**Table 1.** Number of iterations, cost, ATE\* and ARE\* of our approach as compared to centralized Gauss-Newton method for increasing number of robots. ATE\* and ARE\* are measured using  $\eta = 10^{-1}$  as stopping condition.



**Fig. 4.** Shows the trajectories of the six robots and object locations (shows as dots) estimated using centralized mapping and multi-robot mapping for 25 **Chairs** (top) and **House** scenario (bottom).

**Accuracy in the Number of Robots.** Table 1 reports the number of iterations and our accuracy metrics (cost, ATE\*, ARE\*) for increasing number of robots. The table confirms that the distributed approach is nearly as accurate as the centralized Gauss-Newton method and the number of iterations does not increase with increasing number of robots, making our approach scalable to large teams. Usually, few tens of iterations suffice to reach an accurate estimate.

Measurement  noise  $\sigma_r(^{\circ})$ $\sigma_t(\text{m})$	Distributed Gauss-Seidel				Centralized  GN  Cost	ATE* (m)		ARE* (deg)	
	$\eta=10^{-1}$		$\eta=10^{-2}$			Poses	Lmrks.	Poses	Lmrks.
	#Iter	Cost	#Iter	Cost					
1    0.1	5.0	12.7	6.0	12.7	12.5	1.8e-04	1.3e-04	7.5e-02	9.0e-02
5    0.1	5.0	166.6	7.0	166.5	160.9	3.1e-03	2.1e-03	3.3e-01	4.0e-01
10   0.2	5.0	666.2	8.0	665.9	643.4	1.3e-02	8.8e-03	6.7e-01	8.1e-01
15   0.3	6.0	1498.3	10.0	1497.8	1447.2	3.0e-02	2.1e-02	1.0e+00	1.2e+00

**Table 2.** Number of iterations, cost, ATE\* and ARE\* of our approach as compared to centralized Gauss-Newton approach for increasing measurement noise. ATE\* and ARE\* are measured using  $\eta = 10^{-1}$  as stopping condition.



**Fig. 5.** Objects from BigBird dataset used in Field Experiments



**Fig. 6.** (Left) Clearpath Jackal robot used for the field tests: platform and sensor layout; (right) snapshot of the test facility and the Jackal robots.

**Sensitivity to Measurement Noise.** We further test the accuracy of our approach by evaluating the number of iterations, the cost, the ATE\* and the ARE\* for increasing levels of noise. Table 2 shows that our approach is able to replicate the accuracy of the centralized Gauss-Newton method, regardless of the noise level.

### 3.2 Field Experiments

We tested our approach on field data collected by two Jackal robots (Fig. 6) moving in a MOUT (Military Operations in Urban Terrain) facility. We scattered the environment with a set of objects from the BigBird dataset [16], shown in Fig. 5. Each robot is equipped with an Asus Xtion sensor and uses wheel odometry to measure its ego-motion.

We evaluated our approach in two different scenarios, the **stadium** and the **house**. We did two runs inside the **stadium** (**Stadium-1** & **Stadium-2**) and one run in the **house** with objects randomly spread along the robot trajectories. **Stadium** scenario datasets were collected in an indoor basketball stadium with the robot trajectories bounded in a roughly rectangular area. **House** scenario dataset was collected around the living room and kitchen area of a house.

**Object Detection.** We used 12 objects from the BigBird dataset in all three runs. The two-stage process of object detection (semantic verification) followed by pose estimation (geometric verification) ensured that we do not add false positive detections. Our current distributed optimization technique (DGS) is not robust to outliers. The detection thresholds can be further relaxed when using robust pose graph optimization techniques.





**Fig. 7.** Shows YOLO object detection snapshots in three different scenes, (l to r) stadium, house, UW scene 2.

In the first run (**Stadium-1**), 6 objects were added to the map out of 12 objects kept in the environment. Similarly 5 objects were detected in the other two runs. Fig. 7 shows the bounding box snapshots of the detected object in three different scenes. Videos showing YOLO object detection results on UW Scenes2 dataset [23] is available at <https://youtu.be/urZiIJK2IYk> and <https://youtu.be/-F6JpVm0rc0>.

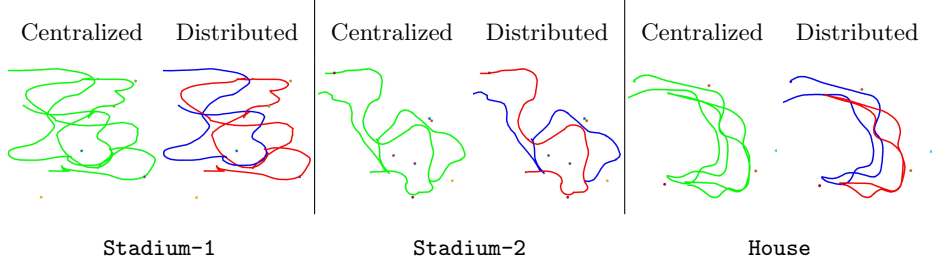
Scenario	Avg. Per-Robot Memory Req. (MB)		Avg. Comm. Bandwidth Req. (MB)	
	PCD	Obj	PCD	Obj
Stadium-1	1.2e+03	1.9e+00	1.9e+01	1.5e-05
Stadium-2	1.4e+03	1.9e+00	1.4e+01	1.1e-05
House	2.1e+03	1.9e+00	1.6e+01	1.3e-05

**Table 3.** Memory and communication requirements for our object based approach (Obj) as compared to Point cloud based approach (PCD) on field data.

**Memory Requirements.** Table 3 compares the average memory requirement per robot to store a dense point cloud map (PCD) with respect to storing a object-based map (Obj). The table also compares the average communication requirements in the case of dense point cloud map and object-based map.

Per-robot memory requirement in the case of dense point cloud is computed as  $n_f K C$  where  $n_f$  is the number of frames,  $K$  is the number of points per frame and  $C$  is the memory required to store each point. In the case of object level map, it is computed as  $n_o P C$  where  $n_o$  is the number of object models and  $P$  is the average number of points in each object model. Table 3 shows that, as expected, the per-robot memory requirement is orders of magnitude smaller with our object-based map as compared to point-cloud-based maps.

When using point clouds, the robots are required sending at least one frame at every rendezvous to estimate their relative pose. So the average communication for dense point cloud map is computed as  $n_c K C$  where  $n_c$  is the number of rendezvous,  $K$  is the number of points per frame and  $C$  is the memory required to send each point. Communication in the case of our object-based map requires sending object category and object pose; a upper bound can be computed as  $n_o L$  where  $n_o$  is the number of objects and  $L$  is the memory required to store



**Fig. 8.** Field tests: estimated trajectories for the our algorithm (distributed Gauss-Seidel) and for the centralized Gauss-Newton method [21]. Trajectories of the two robots are shown in red and blue.

category label and pose of an object. Table 3 confirms that our approach provides a remarkable advantage in terms of communication burden as it requires transmitting 6 orders of magnitude less than a point-cloud-based approach.

Scenario	Initial	Distributed Gauss-Seidel				Centralized	ATE (m)		ARE (deg)	
		$\eta_r = \eta_p = 10^{-1}$		$\eta_r = \eta_p = 10^{-2}$		GN	Poses	Lmrks.	Poses	Lmrks.
	Cost	#Iter	Cost	#Iter	Cost					
Stadium-1	120.73	5.0	1.1e-09	5.0	1.1e-09	1.6e-10	1.9e-10	1.9e-10	1.4e-03	1.2e-04
Stadium-2	310.24	5.0	4.5e-12	8.0	4.4e-12	3.5e-13	2.1e-03	2.2e-03	1.2e-02	1.4e-02
House	43.59	5.0	1.1e-03	6.0	1.0e-03	8.4e-04	4.4e-02	6.2e-02	4.3e-01	4.9e-01

**Table 4.** Number of iterations, cost, ATE\* and ARE\* of our approach as compared to centralized Gauss-Newton method for Field data

**Accuracy.** Fig. 8 shows the trajectories of the two robots in three runs. The figure compares our approach and the corresponding centralized estimate. Quantitative results are given in Table 3 and Table 4, which reports the cost attained by the our approach, the number of iterations, ATE\*, ARE\* as compared to the centralized approach. The table confirms that the distributed approach is nearly as accurate as the centralized Gauss-Newton method and requires very few iterations to compute a good estimate.

## 4 Main Experimental Insights

In our previous work [3], we proposed a distributed Gauss-Seidel method, which reduces the communication burden of distributed SLAM from quadratic to linear in the number of locations co-observed by the robots. However, the work [3], as most related works, requires the exchange of point clouds among the robots, to estimate relative poses during rendezvous. This communication burden is unnecessary, as it leads to exchanging a large amount of uninformative points, and quickly becomes impractical in presence of bandwidth constraints.

In this paper we address this issue by using an object-based representation. Objects provide a suitable abstraction level, and provide a natural tool to

compress large point clouds into a semantically meaningful compact representation. In our system, the robots perform local computation to detect objects and compute their pose. We leverage recent progress in object detection using deep learning: this allows us to reliably detect objects in RGB images at high frame rate. Then, during rendezvous the robots only need to exchange the observed object instances and the measured object poses. This allows the robots to greatly minimize the amount of data exchanged with the teammates.

Experimental evidence shows that our approach leads to a remarkable reduction in the memory footprint (3 orders of magnitude less) and in the communication requirements (6 orders of magnitude less communication), enabling operation in severely bandwidth-constrained scenarios. The experiments also show that our object-based distributed SLAM approach is as accurate as a standard centralized solver and is able to tolerate a large amount of measurement noise.

## 5 Conclusions and Future Work

We proposed a Multi Robot Object-based SLAM approach that uses object landmarks in a multi robot mapping framework. We showed that this approach (i) reduces the memory requirement and information exchange among robots, (ii) is as accurate as the centralized estimate, (iii) scales well to large number of robots and (iv) is resistant to noise.

Our current approach assumes that a model of each observed objects is known in advance. However it can be challenging to store a large number of object models, and to account for intra-class variations. As a future work, we plan to extend our approach to the case where object models are not previously known (at an instance level) and instead object shapes are jointly optimized within our SLAM framework. Another future direction is to improve the robustness of the current pipeline using a distributed algorithm for outlier rejection.

## References

1. Cunningham, A., Indelman, V., Dellaert, F.: DDF-SAM 2.0: Consistent distributed smoothing and mapping. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), Karlsruhe, Germany (May 2013)
2. Choudhary, S., Trevor, A.J.B., Christensen, H.I., Dellaert, F.: SLAM with object discovery, modeling and mapping. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014. (2014) 1018–1025
3. Choudhary, S., Carlone, L., Nieto, C., Rogers, J., Christensen, H., Dellaert, F.: Distributed trajectory estimation with privacy and communication constraints: a two-stage distributed gauss-seidel approach. In: IEEE Intl. Conf. on Robotics and Automation (ICRA). (2016)
4. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6) (2007) 1052–1067
5. Kuipers, B.: The spatial semantic hierarchy. *Artificial Intelligence* **119** (2000) 191 – 233

6. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J.: SLAM++: Simultaneous localisation and mapping at the level of objects. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2013)
7. Rogers, J.G., Trevor, A.J.B., Nieto-Granda, C., Christensen, H.I.: Simultaneous localization and mapping with learned object recognition and semantic data association. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. (Sept 2011) 1264–1270
8. Trevor, A.J.B., Rogers III, J.G., Christensen, H.I.: Planar surface slam with 3d and 2d sensors. In: IEEE Intl. Conf. on Robotics and Automation (ICRA), St. Paul, MN, IEEE (May 2012)
9. Kostavelis, I., Gasteratos, A.: Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems* **66** (2015) 86 – 103
10. Paull, L., Huang, G., Seto, M., Leonard, J.: Communication-constrained multi-AUV cooperative SLAM. In: IEEE Intl. Conf. on Robotics and Automation (ICRA). (2015)
11. Indelman, V., Gurfil, P., Rivlin, E., Rotstein, H.: Graph-based distributed cooperative navigation for a general multi-robot measurement model. *Intl. J. of Robotics Research* **31**(9) (August 2012)
12. Dong, J., Nelson, E., Indelman, V., Michael, N., Dellaert, F.: Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. In: IEEE Intl. Conf. on Robotics and Automation (ICRA). (2015)
13. Roumeliotis, S., Bekey, G.: Distributed multi-robot localization. *IEEE Trans. Robot. Automat.* (August 2002)
14. Thrun, S., Liu, Y.: Multi-robot SLAM with sparse extended information filters. In: Proceedings of the 11th International Symposium of Robotics Research (ISRR'03), Sienna, Italy, Springer (2003)
15. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. *CoRR* **abs/1506.02640** (2015)
16. Singh, A., Sha, J., Narayan, K.S., Achim, T., Abbeel, P.: Bigbird: A large-scale 3d database of object instances. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). (May 2014) 509–516
17. Trevor, A.J.B., Gedikli, S., Rusu, R.B., Christensen, H.I.: Efficient organized point cloud segmentation with connected components. In: Semantic Perception Mapping and Exploration (SPME). (May 2013)
18. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M.E., Beetz, M.: Functional object mapping of kitchen environments. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS). (2008)
19. Rusu, R.B.: Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. PhD thesis, Technische Universität München (2009)
20. Segal, A., Haehnel, D., Thrun, S.: Generalized-icp. In: Proceedings of Robotics: Science and Systems, Seattle, USA (June 2009)
21. Dellaert, F.: Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology (September 2012)
22. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS). (Oct. 2012)
23. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. (May 2011) 1817–1824