DynORecon: Dynamic Object Reconstruction Based on Dynamic SLAM

Yiduo Wang
Australian Centre For Robotics
University of Sydney
2006 Sydney, Australia
Email: yiduo.wang@sydney.edu.au

Jesse Morris
Australian Centre For Robotics
University of Sydney
2006 Sydney, Australia
Email: jesse.morris@sydney.edu.au

Viorela Ila
Australian Centre For Robotics
University of Sydney
2006 Sydney, Australia
Email: viorela.ila@sydney.edu.au

***Abstract*—This paper presents DynORecon, a Dynamic Object Reconstruction system that leverages the information provided by Dynamic SLAM to build a volumetric map of observed moving entities on-the-fly. By exploiting the object motion estimations from the SLAM system, our approach does not only keep updating the representation of dynamic objects in the map to mitigate residual artefacts from previous observations, but also maintains a growing reconstruction for each objects which new observations are constantly integrated into to cover previously unobserved surfaces. Preliminary experiment results of the proposed system using a publicly available dataset are presented in our supplementary video: https://tinyurl.com/dynorecon-demo.**

I. Introduction

In complex real-world scenarios where dynamic objects are abundant, such as urban or natural environments, understanding the motions of these dynamic objects in the scene is essential to any autonomous robotic systems, especially to tasks such as path planning and obstacle avoidance [18, 15, 8]. Therefore, many Simultaneous Localisation And Mapping (SLAM) methods in recent years have been proposed to jointly optimise the motion of dynamic objects together with the egomotion of the robot and the structure of the static background [3, 14, 12]. However, most SLAM systems represent the map in abstract forms such as extracted feature points, lines and planes, which do not assist navigation or exploration tasks as easily as dense volumetric reconstructions [6, 16, 19].

Recent literature has proposed several state-of-the-art reconstruction systems that address the complications of dynamic environments [1, 17, 2]. Dynablox [17] focuses on accurately constructing the free space for navigation and path planning applications using Signed Distance Field (SDF), but does not model the space occupied by dynamic objects apart from the point cloud measurements from LiDAR. The work of Azim *et al*. [1] creates an occupancy map of the whole environment, and detects, segments and tracks moving objects in the scene. However, it does not accumulate the tracked observations of each dynamic object into a complete reconstruction. In contrast, the work of Barad *et al*. [2] focuses on incrementally reconstructing dynamic objects using a 3D Gaussian Splatting (3DGS) framework, which omits the known free space and therefore requires additional heuristics and strategies to be used for applications such as path planning [6, 19]. While

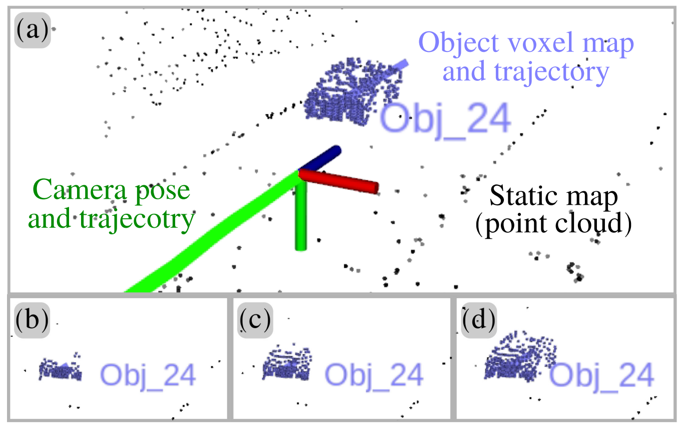

Fig. 1: DynORecon constructing an approaching vehicle incrementally based on a Dynamic SLAM framework. **(a)**: We visualise the voxel reconstruction of a dynamic object (ID: 24) in our experiment sequence in addition to the object trajectory, camera pose and camera trajectory. For visual clarity, we omit the dense reconstruction for the static scene and instead show its sparse SLAM cloud representation. **(b)**: Initial observation of the object. **(c–d)**: Object reconstruction being completed as more observations are integrated into it.

accurately tracking the free space is vital for navigation purposes, modelling obstacles out of the view will further improve the safety and efficiency of planning algorithms. Hence we desire a combination of all these features: mapping each dynamic object, completing their reconstructions using new observations, and explicitly representing known free space.

To this end, we propose DynORecon, a system designed to reconstruct dynamic objects within dynamic environments, and present our preliminary results. Using the dynamic estimates from a state-of-the-art Dynamic SLAM system [14], we create an isolated and small-scale reconstruction for each dynamic object in addition to the static background [20]. The estimated object motions [14] are leveraged to propagate a consistent object reconstruction through both time and space. The proposed system can further integrate new scans of these objects into their corresponding volumetric map while maintaining previous observation, growing the reconstruction incrementally. We also incorporate an occupancy representation that explicitly represents the known free space[10, 19, 6] to better support downstream robotic applications such as path planning and obstacle avoidance in dynamic environments.

Fig. 2: This is an example output on dynamic objects from the SLAM framework [14]. Each object is segmented out in the 2D view and represented by a uniquely coloured dense patch of pixels. Their estimated motions and trajectories are visualised in the 2D view.

The features of this work are as follows:

- Implementation of a dynamic object reconstruction that leverages object motion estimations to completely mitigate residual artefacts from observation history,
- Representation of each dynamic object that constantly grows via the integration of new observations while maintaining previously covered surfaces,
- Explicit representation of known free space that incorporates observed objects and their predicted occupancy.

II. Method

DynORecon expects as inputs the pose of the sensor (e.g. a RGB-D camera), a labelled 3D point cloud which indicates the dynamic instances from the static background, and the estimated motions of observed objects. This information can be provided directly by state-of-the-art Dynamic SLAM systems [3, 12, 14]. An example of detected and segmented dynamic objects and their estimated motions from the SLAM framework [14] is shown in Fig. 2.

A. Notations and Background

Our system uses three main reference frames to reconstruct the dynamic objects. The world frame $\{W\}$ defines the fixed global reference frame. The sensor $\{X_k\}$ and object $\{L_k\}$ frames are associated with the sensor and object poses ${}^W\mathbf{X}_k, {}^W\mathbf{L}_k \in \mathrm{SE}(3)$ in $\{W\}$ at time-step k, respectively.

For each 3D point measurement point in the sensor frame ${}^{X_k}\tilde{\mathbf{m}}_k^i$, ${}^{X_k}\mathbf{m}_k^i = \left[{}^{X_k}\tilde{\mathbf{m}}_k^i, 1\right]^\top$ defines its homogeneous coordinates. For any time-independent variables, i.e. static within the reference frame, we omit the time-step k. For example, a static point in the world frame is ${}^W\mathbf{m}^i = {}^W\mathbf{X}_k\, {}^{X_k}\mathbf{m}_k^i$, and a point on a rigid-body object $\{L_k^j\}$ is ${}^{L^j}\mathbf{m}^i$, where j is a unique object identifier. The same object point can be expressed in the world frame $\{W\}$ as ${}^W\mathbf{m}_k^i = {}^W\mathbf{L}_k^j\, {}^{L^j}\mathbf{m}^i$, where the j becomes implicit.

The motions of an object from time-step $k-1$ to k, ${}^{L_{k-1}}_{k-1}\mathbf{H}_k^j \in \mathrm{SE}(3)$, is defined as:

$$ {}^{L_{k-1}}_{k-1}\mathbf{H}_k^j = {}^W\mathbf{L}_{k-1}^{j\ -1}\, {}^W\mathbf{L}_k^j, \tag{1} $$

DynORecon assumes that moving objects are rigid bodies. For the j-th object with motion ${}^{L_{k-1}}_{k-1}\mathbf{H}_k^j$, there exists a single SE(3) transformation in $\{W\}$ for all points on it [4, 14]:

$$ {}^W\mathbf{m}_k^i = {}^{\ \ W}_{k-1}\mathbf{H}_k^j\, {}^W\mathbf{m}_{k-1}^i, \tag{2} $$

which our SLAM framework estimates directly based on the object points [14]. It can also be computed using common Dynamic SLAM outputs such as object poses and motions [4].

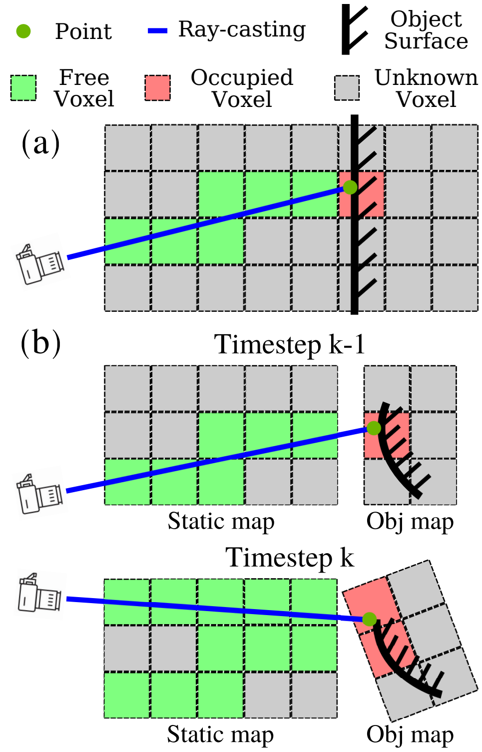

Fig. 3: A simple 2D example of occupancy update for **(a)** static scene and **(b)** dynamic object. The green point represents a point measurement on the surface, and the blue line is the ray cast from the sensor to the observed point. Voxels that the ray intersects with have their occupancy probabilities updated. Other voxels remain unknown.

B. Scan Integration

DynORecon employs OctoMap as the reconstruction core and uses the same logs-odd occupancy fusion strategy proposed by Hornung *et al.* [10] to integrate a new scan. We first cast a ray between the sensor pose ${}^W\mathbf{X}_k$ and each 3D point measurement ${}^W\mathbf{m}_k^i$ and find all voxels that this ray intersects with. The voxel containing the point measurement ${}^W\mathbf{m}_k^i$ has its occupancy probability increased because each point represents a hit on a surface, while all other voxels that the ray intersects with receive a decrease in their occupancy in order to explicitly represent the observed free space. For static scenes, Fig. 3 (a) provides a simple example for the static scene in 2D, and the following sections (II-C and II-D) will explain the reconstruction of dynamic objects in more detail.

C. Map Structure

Instead of creating one monolithic global reconstruction, DynORecon maintains a collection of small-scale maps corresponding to each dynamic object in addition to one global map that covers the static background. This strategy is inspired by previous works such as [20, 19] that use multiple reconstructions to efficiently handle relative movements among maps.

Each object reconstruction is associated with a object pose ${}^W\mathbf{L}_k^j$ that changes over time and moves the reconstruction with it in the world frame. Within each object reconstruction, all points are expressed in the object local frame: ${}^{L^j}\mathbf{m}^i = {}^W\mathbf{L}_k^{j\ -1}\, {}^W\mathbf{m}_k^i$, and due to the rigid-body assumption, they are

TABLE I: Average errors of camera pose, object motion, and object pose reported by the SLAM framework [14] in simulation experiment.

camera pose $E_r(°)$	camera pose E_t(m)	object motion $E_r(°)$	object motion E_t(m)	object pose $E_r(°)$	object pose E_t(m)
2.09	0.29	1.89	0.54	2.91	0.57

static and time-invariant within the object frame $\{L_k^j\}$.

D. Object Local Reconstruction

Different from the conventional scan integration strategy discussed in Section II-B, we integrate the points on dynamic objects into the object local reconstruction so that there is no residual artefacts from movements. DynORecon first conducts the process of ray-casting and occupancy computation as explained in Section II-B using point measurements in the world frame ${}^W\mathbf{m}_k^i$, but we process the occupied and free voxels separately as demonstrated in Fig. 3. Observed free voxels are integrated into the global static map, following the strategy proposed by Dynablox [17]. However, occupied voxels are not updated in the world frame; they are dynamic and unlikely to remain in the same space, which will lead to residual artefacts if integrated into a static world map. Instead, DynORecon switches to the transformed points in the object frame ${}^{L^j}\mathbf{m}^i$ and integrates them into the object local map to incrementally build up the reconstruction of the dynamic object.

The initial reference frame of each object upon its first observation is defined using the centroid of all corresponding 3D point measurements as its position and identity as its orientation. This initial object pose ${}^W\mathbf{L}_{\text{initial}}^j$ is then propagated into following frames using the motion model in (2). While these object poses do not necessarily reflect the true orientation or position of the dynamic object, the rigid-body motion model in (2) ensures that the reference frame $\{L_k\}$ moves consistently with its corresponding object points ${}^{L^j}\mathbf{m}^i$, facilitating a rigid object-centric reconstruction.

E. Occupancy Lookup and Prediction

In addition to incrementally constructing a volumetric model of each dynamic object, DynORecon also provides occupancy information that is useful for robotic tasks such as exploration and navigation. Looking up the occupancy of any 3D point involves searching through all relevant maps [9] because each object is represented by its own reconstruction. In order to improve the computation efficiency, we first check the desired coordinate with the bounding box of each map to skip the occupancy lookup process in any far-away maps. After gathering the occupancy probabilities of the desired coordinate from all close-by maps, we use the log-odds fusion strategy of OctoMap [10], as mentioned in Section II-B, to obtain the final occupancy result.

Given the motion information from the Dynamic SLAM module, we can also predict the future pose of each object by simply propagating the object with its motion, ${}_{k-1}^{W}\mathbf{H}_k^j$, which moves the object local map in the world frame. These predictions can further improve the efficiency of path planning and obstacle avoidance applications [18, 15, 8].

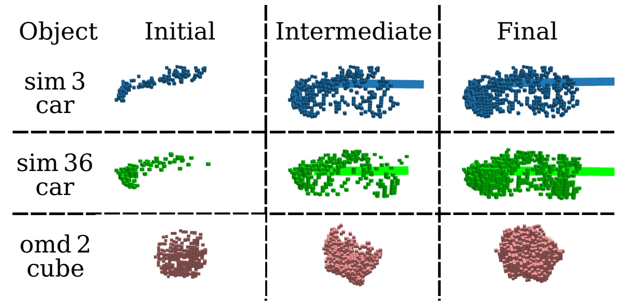

Fig. 4: More examples of dynamic objects (2 cars in simulation [11] and 1 cube in OMD [13]) being incrementally reconstructed by the proposed system on-the-fly.

III. Experiments

We tested our proposed reconstruction system with a simulated outdoor sequence provided by ClusterSLAM [11] using CARLA [5], as well as a real-world sequence from the Oxford Multimotion Dataset (OMD) [13]. The visual inputs were first processed by the Dynamic SLAM system from our previous work [14] to accurately estimate the camera poses and object motions, in addition to providing labelled 3D point clouds. Table I shows the average errors in the SLAM estimations in the simulation experiment. For the comparison of our SLAM framework with other state-of-the-art methods, please refer to our previous work [14]. The voxel resolution DynORecon used in the outdoor experiment was 10 cm due to the density and distance of object points, and 5 cm in the OMD experiment for the small-scale indoor scene.

In the simulation sequence, the camera travelled for roughly 480 m and observed 14 dynamic objects. Fig. 4 presents the close-up reconstruction results of two more vehicles in addition to the one presented in Fig. 1, as well as a cube in OMD [13]. Because the dataset provided by ClusterSLAM [11] does not include mesh models of dynamic vehicles, we are unable to quantitatively evaluate the reconstruction quality. The qualitative demonstration of the proposed system running online is shown in our supplementary video.

We have observed that our SLAM and reconstruction pipeline can sometimes lose track of dynamic objects, for instance due to occlusion. In the current iteration of our pipeline, the SLAM system considers the re-observed dynamic object a new entity, and DynORecon in turn creates a new model for this object instead of continuing the previous map.

The cases of misclassification by instance semantic segmentation are currently addressed by SLAM, as the SLAM component tracks 3D points with consistent object labels and motions based on rigid-body motion assumption between consecutive frames. We plan to further leverage the volumetric occupancy map to filter out 3D point outliers and maintain a consistent reconstruction in future versions of DynORecon.

IV. Conclusion and Future Work

In summary, we present DynORecon, a reconstruction system for dynamic environments with a focus on incrementally mapping dynamic objects. The proposed system further maintains an explicit representation of observed free space for navigation and obstacle avoidance purposes. For dynamic objects, we process observed occupied and free space using different strategies so as to avoid residual artefacts due to their movements. Our preliminary results are presented in our supplementary video.

For further development, we are first interested in handling non-rigid-body objects, which is abundant in natural environments. Our intended strategies include tracking each rigid segments separately, or incorporating a more complicated motion model as inspired by BodySLAM++ [7]. Reconstruction method with a more efficient scan integration process are also desired for large-scale outdoor environments [6].

We are also interested in incorporating the voxel map in the SLAM framework, computing cost functions directly on the dense reconstruction. Additionally, we are interested in exploring the alternative strategy of maintaining one monolithic reconstruction for better memory efficiency and occupancy lookup speed. With object motions accurately estimated, we plan to introduce new temporal variables into conventional occupancy or SDF representations so that dynamic voxels can be easily updated or transformed. We would like to deploy the proposed system with a navigation module in the real world.

References

[1] Asma Azim and Olivier Aycard. Detection, classification and tracking of moving objects in a 3d environment. In *IEEE Intelligence Vehicles Symp. (IV)*, pages 802–807, 2012.

[2] Kuldeep R Barad, Antoine Richard, Jan Dentler, Miguel Olivares-Mendez, and Carol Martinez. Object-centric reconstruction and tracking of dynamic unknown objects using 3d gaussian splatting, 2024.

[3] Berta Bescos, Carlos Campos, Juan D. Tardós, and José Neira. Dynaslam ii: Tightly-coupled multi-object tracking and slam. *IEEE Robotics and Automation Letters*, 6 (3):5191–5198, 2021.

[4] Gregory S. Chirikjian, Robert Mahony, Sipu Ruan, and Jochen Trumpf. Pose changes from a different point of view. In *Proc. of the ASME Intl. Design Engineering Technical Conf. (IDETC)*. ASME, 2017.

[5] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proc. of the Conf. on Robot Learning (CoRL)*, pages 1–16, 2017.

[6] Nils Funk, Juan Tarrio, Sotiris Papatheodorou, Marija Popović, Pablo F. Alcantarilla, and Stefan Leutenegger. Multi-resolution 3d mapping with explicit free space representation for fast and accurate mobile robot motion planning. *IEEE Robotics and Automation Letters*, 6(2): 3553–3560, 2021.

[7] Dorian F. Henning, Christopher Choi, Simon Schaefer, and Stefan Leutenegger. Bodyslam++: Fast and tightly-coupled visual-inertial camera and human motion tracking. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3781–3788, 2023.

[8] Andreas Hermann, Joerg Bauer, Sebastian Klemm, and Ruediger Dillmann. Mobile manipulation planning optimized for gpgpu voxel-collision detection in high resolution live 3d-maps. In *Intl. Symp. on Robotics/Robotik*, pages 1–8, 2014.

[9] Bing-jui Ho, Paloma Sodhi, Pedro Teixeira, Ming Hsiao, Tushar Kusnur, and Michael Kaess. Virtual Occupancy Grid Map for Submap-based Pose Graph SLAM and Planning in 3D Environments. *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 1 (2):2175–2182, 2018.

[10] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

[11] J. Huang, S. Yang, Z. Zhao, Y. Lai, and S. Hu. Clusterslam: A slam backend for simultaneous rigid body clustering and motion estimation. In *Proc. of the Intl. Conf. on Computer Vision (ICCV)*, 2019.

[12] Kevin M Judd and Jonathan D Gammell. Multimotion Visual Odometry (MVO). *Intl. J. of Robotics Research*, 2024.

[13] Kevin Michael Judd and Jonathan D Gammell. The Oxford Multimotion Dataset: Multiple SE(3) Motions with Ground Truth. *IEEE Robotics and Automation Letters*, 4(2):800–807, 2019.

[14] Jesse Morris, Yiduo Wang, and Viorela Ila. The importance of coordinate frames in dynamic slam. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2024.

[15] Mike Phillips and Maxim Likhachev. Sipp: Safe interval path planning for dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5628–5635, 2011.

[16] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto. Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps. *IEEE Robotics and Automation Letters*, 5(1): 227–234, 2020.

[17] Lukas Schmid, Olov Andersson, Aurelio Sulser, Patrick Pfreundschuh, and Roland Siegwart. Dynablox: Real-time detection of diverse dynamic objects in complex environments. 8(10):6259 – 6266, 2023.

[18] J. van den Berg, D. Ferguson, and J. Kuffner. Anytime path planning and replanning in dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2366–2371, 2006.

[19] Yiduo Wang, Milad Ramezani, Matias Mattamala, Sundara Tejaswi Digumarti, and Maurice Fallon. Strategies for large scale elastic and semantic lidar reconstruction. *J. of Robotics and Autonomous Systems*, 155, 2022.

[20] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 5231–5237. IEEE, 2019.