Robust Dense Mapping for Large-Scale Dynamic Environments

Ioan Andrei Bârsan[1], Peidong Liu[1], Marc Pollefeys[1,2] and Andreas Geiger[1,3]

***Abstract*— We present a stereo-based dense mapping algorithm for large scale dynamic environments. In contrast to other existing algorithms, we simultaneously reconstruct the static background, the moving objects and the potentially moving but currently stationary objects separately, which is desirable for high-level mobile robotics tasks such as path planning in crowded environments. We use both instance-aware semantic segmentation and sparse scene flow to classify objects as either background, moving or potentially moving, thereby ensuring that the system is able to model objects transitioning from static to dynamic, such as parked cars. Given camera poses estimated using visual odometry, both the background and the (potentially) moving objects are reconstructed separately by fusing the depth maps computed from the stereo input. In addition to visual odometry, sparse scene flow is also used to estimate the 3D motions of the detected moving objects, in order to reconstruct them accurately. A map pruning technique is further developed to improve reconstruction accuracy and reduce memory consumption, leading to increased scalability. We evaluate our system thoroughly on the well-known KITTI Vision Benchmark Suite. Our system is capable of near real-time operation on a PC at approximately 2.5Hz, with the primary bottleneck being the instance-aware semantic segmentation, which is a limitation we hope to address in future work.**

I. INTRODUCTION

Over the course of the past two decades, the field of 3D visual perception has undergone a dramatic evolution, enabling many new applications such as self-driving cars, automatic environment mapping and high-quality object scanning using consumer-grade sensors such as the Microsoft Kinect. Nevertheless, despite the field's rapid evolution, numerous open problems still remain. Among them is the task of real-time large-scale mapping for dynamic outdoor environments.

Map representations range from the conventional point cloud [2], [18], [22] and mesh-based [30] approaches to more sophisticated ones based on, e.g., volumetric methods [21], [24], [31]. While point-based approaches can provide accurate reconstructions, the level of detail depends on the number of points used, which can become prohibitive for real-time applications. Furthermore, without additional meshing steps, sparse representations do not easily accommodate tasks like path planning, which require line-of-sight computations. In contrast, volumetric representations can be used directly in path planning. Such methods represent 3D geometry as a regularly sampled 3D grid by means of an implicit function which either models occupancy or encodes the distance to the closest surface. Although memory usage is one of the main limitations of these methods, multiple techniques have been proposed for dealing with this problem using either hashing [20], octrees [26] or data-adaptive discretization [12].

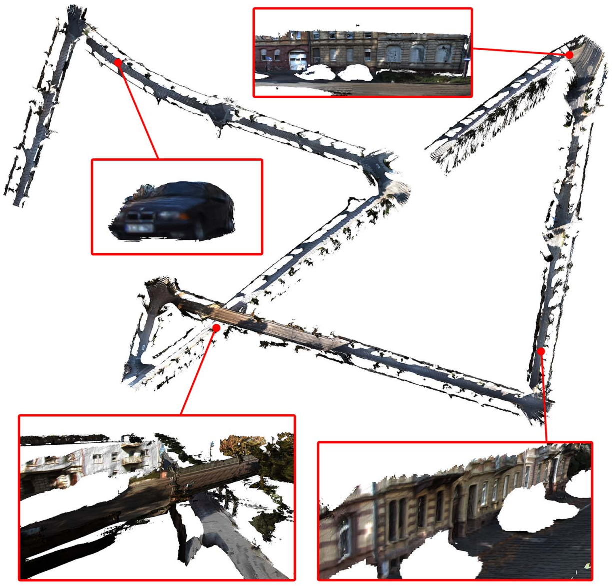


Fig. 1: A static map with all moving and potentially moving objects removed, as well as the reconstruction of one independently moving vehicle encountered in the environment.

The sensors typically used in 3D mapping include depth [20], monocular [31], and stereo [23] cameras. While depth cameras can readily provide high-quality depth measurements, their usage is limited to indoor environments. On the other hand, conventional monochrome/RGB cameras can operate successfully in more environments, i.e., both indoors and outdoors. As a prerequisite for 3D mapping, the camera poses are usually estimated by visual (inertial) odometry and are used to fuse 2.5D depth maps into a consistent 3D model. While both monocular and stereo cameras can be used for visual odometry, monocular cameras suffer from the so-called unobservable scale problem which results in the inferred 3D model lacking a metric scale. Therefore, additional sensors such as inertial measurement units (IMU) are typically used together with a monocular camera to unambiguously recover its poses [31]. However, IMU measurements are usually noisy, especially on a mobile robot with vibrating hardware components, such as an engine or actuators, which, in turn, affects the accuracy of camera pose estimations. Stereo cameras do not suffer from this

[1] Computer Vision and Geometry Group, ETH Zürich, Switzerland andrei.ioan.barsan@gmail.com

[2] Microsoft, Redmond, USA

[3] Autonomous Vision Group, Max Planck Institute for Intelligent Systems, Tübingen, Germany

problem and can observe the metric scale directly for use in 3D mapping [14].

For real-world applications, mobile robots usually need to operate in large-scale, complex environments, e.g., autonomous driving in a crowed city, which is full of pedestrians and moving vehicles. These scenarios pose a challenge for most existing 3D mapping algorithms, which usually assume that the surrounding environments are static, meaning that dynamic objects would corrupt the reconstructed map if not explicitly considered. While numerous existing mapping systems are capable of operating within dynamic environments, they typically achieve this robustness by treating sensory input associated with dynamic objects as noise, and ignoring it in order to preserve the consistency of the static map [23], [27], [29]. However, this process discards valuable information about the dynamic objects in the environment, which can potentially be used in applications such as path planning and traffic understanding.

In this paper, we propose a robust dense mapping algorithm for large scale dynamic environments. In particular, we use a volumetric representation based on voxel block hashing [11], [20] for large scale environments. A pair of stereo cameras is used to infer the egomotion and reconstruct the surrounding world. Semantic cues and the sparse scene flow are used to classify objects as either static background, moving, or potentially moving but currently stationary. The objects are then treated separately such that a high-quality corruption-free dense map of the static background can be inferred. Instead of discarding dynamic objects, we reconstruct them separately, which has potential for future applications such as high-level robotic decision making. Each object is represented by a separate volumetric map and its motion is tracked. To reduce the overall memory consumption and increase the accuracy of the resulting reconstructions, a map pruning technique is proposed. Fig. 1 shows a reconstructed static map with (potentially) moving objects removed, and the reconstruction of an independently moving vehicle encountered in the environment.

The main contributions of our algorithm are summarized as follows. We develop an efficient stereo-based dense mapping algorithm robust to dynamic environments. The system builds a high-quality static map and separate 3D reconstructions of moving and potentially moving objects in an online manner. A map pruning technique is proposed to further improve the mapping accuracy and reduce memory consumption, thereby increasing the system's scalability.

II. RELATED WORK

There is a large body of recent and ongoing research in fields related to autonomous robotics, such as dense mapping, object tracking, and semantic segmentation. In this section, we focus on the parts of this research most relevant to our work.

Reddy et al. [23] describe a stereo-based algorithm for robust SLAM that works in both static and dynamic environments. They separate the static background from the moving objects by minimizing a joint CRF-based semantic motion segmentation energy function. Both the ego-motion and the geometric structure are optimized using bundle adjustment by incorporating additional geometric and semantic priors. The experimental results are good but not without limitations. The reconstructed moving objects are not consistent with the real objects due to noisy depth estimation. The system is not applicable to large scale environments and is not able to run in (near) real-time due to the expensive optimization of the aforementioned energy function.

Vineet et al. propose a large scale semantic dense reconstruction algorithm in [27]. The system runs in near real time, but moving objects are not considered explicitly, being reconstructed together with the static background. To avoid map corruption, they propose an object class-dependent weighting scheme for the depth map fusion. In particular, areas of the map corresponding to moving objects are updated using larger weights, i.e., more aggressively, in order to ensure that they remain consistent.

KinfuSeg [32] extends the RGB-D camera-based KinectFusion [19] pipeline for dynamic indoor environments. They reconstruct both the static and dynamic parts of an environment separately. However, by using the same map representation as KinectFusion, this system inherits its limitations, being unable to operate in environments larger than room-sized ones. Moreover, the system does not support more than one dynamic object at any given time.

Kochanov et al. [10] propose a large scale dense mapping system for dynamic environments. They use stereo cameras to build a dense semantic map of an environment which also incorporates information on the dynamics of the scene. Their method is centered around augmenting a sparse voxel grid to also include a running estimate of each cell's scene flow, in addition to the semantic and occupancy information. However, the described pipeline does not run in real time. Among its components, it leverages the dense scene flow method of Vogel et al. [28], which can take up to 300 seconds for a single frame. The system is also limited to detecting and tracking dynamic objects, without attempting to reconstruct them.

Jiang et al. [8] present a SLAM system for dynamic environments which uses LIDAR and cameras to reconstruct both the static map, as well as the dynamic objects within, using 3D sparse subspace clustering of key-point trajectories to identify and segment dynamic objects. Nevertheless, the presented system does not run in real time, requiring roughly 12 minutes to process a 70-frame (i.e., seven-second) sequence. A follow-up work from the same authors [9] improves the system's robustness to challenges such as partial occlusions by using 3D flow field analysis.

Co-Fusion [15] is another real-time dense mapping system capable of reconstructing both a static background map, as well as dynamic objects from a scene. The paper presents a real-time pipeline which can segment and track dynamic objects based on either motion or semantic cues, reconstructing them separately using a surfel-based representation. While most of the experiments are performed on indoor RGB-D sequences, both real-world and synthetic, some limited results on the Virtual KITTI dataset [3] are also presented. However, both the quality and the scale of the reconstructions performed on the Virtual KITTI data are limited, in comparison to the indoor sequences. Moreover, the authors do not address the problem of noise reduction for 3D reconstruction, which is an important consideration for stereo-based large-scale dense mapping of outdoor environments.

III. DYNAMIC RECONSTRUCTION

Fig. 2 shows an overview of our dense mapping system. At every frame, the system first computes the dense depth map, the instance-level semantic segmentation, the sparse scene flow, and the camera egomotion from the input stereo images. The semantic segmentation results are then used to identify the dynamic objects (e.g., vehicles and pedestrians), separating them from the static background. The motion states of the recognized objects are inferred from the sparse scene flow, i.e., either moving or potentially moving but currently stationary. To obtain consistent 3D models of the moving objects, their 3D position is tracked and their inferred motion is used in the volumetric fusion. Dense depth maps corresponding to different categories (i.e., static background, moving objects and stationary dynamic objects) are then fused separately to obtain consistent high-quality 3D models. Compared to those produced by RGB-D cameras, the depth maps computed from a stereo camera tend to be significantly noisier. To remove artifacts associated with this noise, a fixed-lag map regularizer is used to prune noisy voxels. This process also reduces the memory consumption of the system by a large margin. We will now describe the details of each component of the pipeline.

A. Dense depth map computation

For calculating depth maps we leverage state-of-the-art real-time stereo estimation techniques. In particular, we consider Efficient Large-scale Stereo Matching (ELAS) [5] and DispNet [16]. The former method solves the ambiguities associated with stereo matching by first computing a sparse "support" disparity map by robustly matching keypoints between the two images. It then uses the Delaunay triangulation of the support keypoints to guide the computation of the final dense disparity map. DispNet is an end-to-end neural network architecture which directly estimates full-resolution disparity maps from rectified stereo input pairs, casting the problem as a pixel-wise regression task. We have chosen to evaluate our system using disparity maps produced by both methods given their different approaches, which exhibiting different strengths and weaknesses: ELAS depth maps tend to be sharper but incomplete, with many gaps present in reflective and transparent areas, while the results of DispNet are denser and robust to non-lambertian surfaces, but less sharp, especially around object boundaries.

B. Object segmentation and 2D tracking

An instance-aware semantic segmentation algorithm is used to recognize dynamic and potentially dynamic objects from a single image (e.g., both actively moving and parked cars). We use Multi-task Network Cascades [1] (MNC), a state-of-the-art deep neural network architecture for this task. This component detects and classifies object instances in an input image using the 20 labels from the Pascal VOC2012 dataset.

The identities of recognized objects are assigned per image. Therefore, an additional association (tracking) step is performed in order to obtain a consistent identity across consecutive frames for each recognized object. We achieve this by associating new detections with existing tracks by ranking them based on the Intersection-over-Union (IoU) score between a new detection and the most recent frame in a track.

C. Sparse scene flow and visual odometry

Both the sparse scene flow and the vehicle egomotion are computed using `libviso2` [6]. Following the method described in [6], the computation of the sparse scene flow is based on two-view and temporal stereo. More specifically, simple blob and corner features are matched between the current left and right frames, and the previous ones, resulting in pairs of 3D points from consecutive time steps, i.e., the scene flow. These values are then used in the computation of the visual odometry, as well as in estimating the 3D motion of the tracked objects. A RANSAC-based approach is used to compute visual odometry from the sparse scene flow vectors. At every frame, the six-degree-of-freedom pose of the stereo camera relative to the previous frame is estimated by minimizing the reprojection error of the 3D points triangulated from the feature positions.

D. 3D object tracking

3D object tracking occurs on a frame-to-frame basis and uses the sparse scene flow computation described in the previous pipeline stage as input. The masked scene flow associated with a specific object instance is used as input to estimate the motion of a virtual camera with respect to the object instance, which is assumed to be static. If the estimation is successful, then the 3D motion of the object is equal to the inverse of the virtual camera's motion. For static objects, this obviously means that the resulting 3D object motion will be nearly identical to the camera's egomotion. This can be used to classify objects with known motion as either static or dynamic. The motion of the virtual camera is computed the same way as the egomotion, using the approach from `libviso2` described in the previous section.

E. Static map and individual object reconstructions

Our system uses InfiniTAM [11] for volumetric fusion. InfiniTAM is an efficient framework for real-time, large scale depth fusion and tracking. Instead of fusing depth maps into a single volumetric model, we separate the static background from the dynamic objects. The vehicle egomotion computed by the visual odometry is used to fuse the static parts of the input color and depth maps, which are identified based on the output of the instance-aware semantic segmentation component. Both moving and potentially moving objects

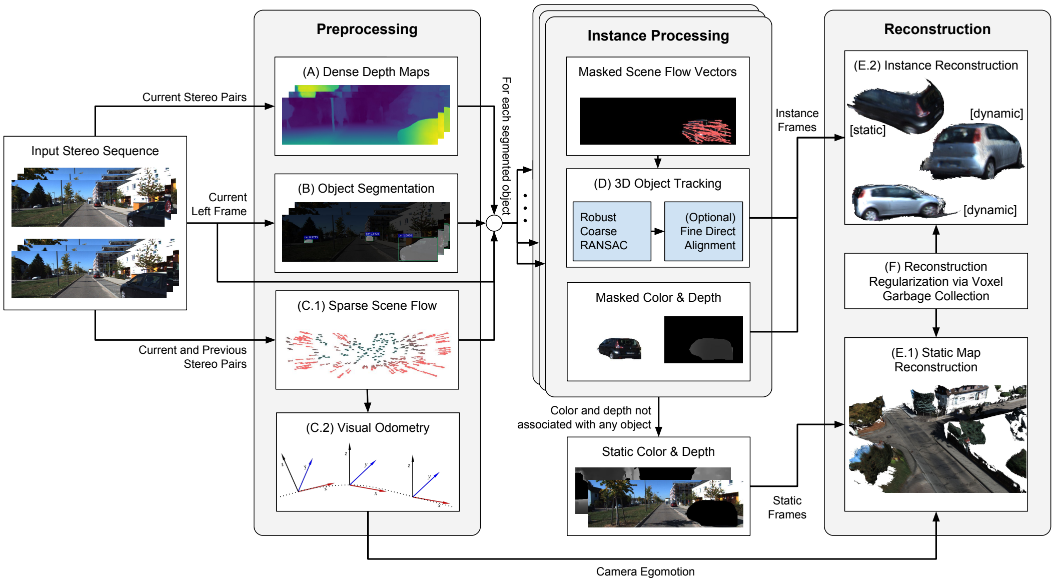


Fig. 2: An overview of the system pipeline.

are reconstructed individually. Each object is represented by a single volumetric model, whose coordinate system is centered around the pose of the camera in the frame where the object was first observed. The estimated 3D motion of the individual objects are used for the object volumetric fusion. The corresponding depth map of each object from each frame is also obtained by the instance-level semantic segmentation algorithm.

F. Fixed-lag map regularization

When performing volumetric reconstruction at a small scale, such as when scanning room-sized environments with an RGBD camera, the magnitude of the noise associated with the depth estimation is relatively small, compared to the error associated with estimating depth from stereo [7], [13]. Therefore, the development of methods for actively tackling noise-induced artifacts is not a primary consideration when dealing with small-scale reconstructions, but it becomes a necessity when switching to outdoor stereo reconstruction. Fig. 3 shows an example of the artifacts produced when rendering a stereo depth map in 3D. The outlines of most objects have a comet-like trail facing away from the direction from which they were perceived. The streaks not only reduce the map quality, but also lead to increased memory usage, since they occupy large volumes, negatively impacting the system's scalability.

In order to reduce the impact of the noise, and improve the reconstruction quality while reducing the memory footprint of our system, we turn to a technique first proposed by Nießner et al. [20]. Their work describes a garbage collection method for removing voxel blocks allocated due to noise and moving objects. For every $8 \times 8 \times 8$ voxel block, the minimum absolute value of the truncated signed distance function (TSDF) is computed together with the maximum measurement weight. Blocks with a minimum absolute TSDF above a certain threshold, which correspond to areas far enough from the represented surface so as not to contribute meaningfully to it, and blocks with a maximum weight of zero, which are blocks not containing any measurement information, are removed from the reconstruction, reducing artifacts and freeing up memory.

We extend this method in two ways: First, we increase its granularity, making it capable of operating on a per-voxel basis as opposed to a per-voxel block basis, meaning that noisy voxels can be deleted even when the block to which they belong cannot be. Note that memory is still only freed when entire voxel blocks are deleted. This is due to the voxel block hashing-based map representation used in InfiniTAM, which does not allow individual voxels to be deallocated.

Second, we improve the method's scalability by allowing it to take advantage of InfiniTAM's visible block lists, removing the need to run the method on the entire map at every frame, which would have prohibitively high computational costs. This improvement is based on two insights: (1) The movement of the vehicle-mounted camera is not "loopy," as is common in indoor sequences, with loop closures being rare. (2) Cleaned-up areas do not need to be re-processed until they are revisited by the camera.

InfiniTAM keeps track of a list of currently visible blocks, for efficiency reasons. By storing a snapshot of this list at every time step, the history of which blocks were visible over time can be recorded. At every time step t, we process the voxel blocks visible at time $t - k_{\text{minAge}}$, where k_{minAge} is the minimum age of a voxel block in terms of time steps, which is set empirically such that the majority of the voxel blocks processed by the collection are no longer visible[1]. In other words, the garbage collection is performed in lockstep with

[1] In most of our experiments it is set to 300, corresponding to 30 seconds of driving time.

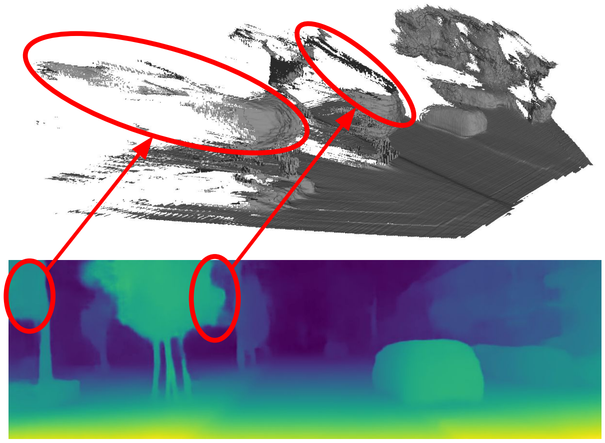


Fig. 3: Examples of the streak-like artifacts produced in 3D reconstructions by noisy depth maps.

the fusion: after processing the nth input frame the system performs voxel garbage collection on the blocks visible at the time of input frame $n - k_{\text{minAge}}$.

In scenes where the static map can easily occupy hundreds of thousands of voxel blocks, the visible list at every given frame typically holds around 8,000–12,000 blocks, resulting in a 10-fold reduction of computational costs, as compared to running the process over the entire volume at every step. On an nVIDIA GeForce GTX 1080 GPU, this results in an overhead of less than 1ms at every frame. Fig. 4 shows the impact of the voxel garbage collection on a scene reconstruction. Note that we only apply this technique when performing voxel garbage collection on the static map. Given their limited dimensions, car reconstructions are always pruned in their entirety at every frame.

IV. EXPERIMENTS

We use the well-known KITTI Vision Benchmark Suite [4] to evaluate our system. Both qualitative and quantitative evaluations of the accuracy of the obtained 3D reconstructions are performed. The effectiveness of the fixed-lag map regularizer is also evaluated.

A. *Qualitative accuracy evaluation*

Fig. 5 shows two example reconstructions of the same view by dynamic-aware fusion and standard fusion on a KITTI sequence. It shows that individually moving vehicles corrupt the environment map when ignored, while the dynamic-aware fusion provides a corruption-free environment map.

B. *Quantitative accuracy evaluation*

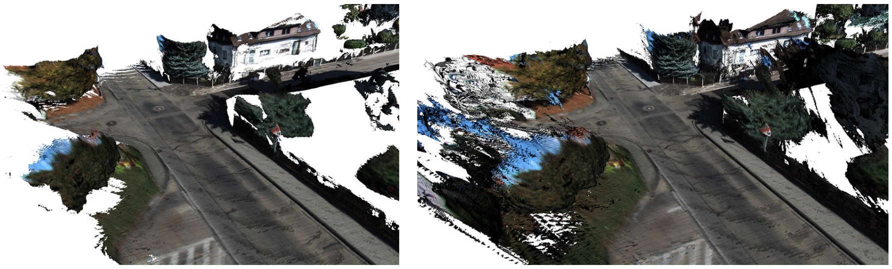
We base our experiments on the video sequences from the KITTI odometry and tracking benchmarks, using the LIDAR as a ground truth for evaluating the quality of both the reconstructed static maps, as well as of the dynamic object instances. Similar to the works of Sengupta et al. [25] and Vineet et al. [27], we compare the LIDAR ground truth to the current reconstruction and to the input depth map at every frame by projecting everything to the left camera's frame. This follows the same principle as the 2015 KITTI stereo benchmark with the main difference being that we average the accuracy scores across all the frames in a sequence instead of a limited number of selected frames. The accuracy at every frame is computed using the methodology from the 2015 KITTI stereo benchmark: we compare disparities (input disparities and disparity maps computed from the active reconstruction against the ground truth disparities derived from LIDAR), and consider points whose delta disparity is greater than 3px *and* 5% of the ground truth disparity as erroneous. The completeness of the input and fused disparity maps is also computed as the percentage of ground truth points in the left camera's field of view which have a corresponding value in either the input or the fused disparity maps.

In order to separately evaluate the reconstruction accuracy of the background and the dynamic objects in a manner which compares the fused map and the input disparity maps in a fair way, we perform *semantic-aware evaluation*. To this end, we use the semantic segmentation results produced by the Multi-task Network Cascades. The input and fused depth maps are evaluated as follows: (1) Ground truth points associated with no potentially dynamic objects are counted towards the *static map* statistics. (2) Ground truth points associated with potentially dynamic objects which are being reconstructed are counted towards the *dynamic object* statistics. (3) The remaining ground truth points which correspond to dynamic objects not undergoing reconstruction (e.g., bikes, pedestrians, or distant cars whose 3D motion cannot be computed reliably) are ignored. This method is, obviously, imperfect, as it relies on the computed semantic segmentation, which is not always fully reliable. Nevertheless, we have found it to work well in practice, allowing us to draw numerous insights about our system's performance under various conditions, as will be described in detail below.

Fig. 6 presents the accuracy and completeness of the system's background and object reconstructions, aggregated over all frames in the 11 KITTI odometry training sequences. Several conclusions can be drawn from the results. First, using ELAS depth maps leads to more accurate static maps than DispNet, but less accurate object reconstructions. This is explained by DispNet's superior robustness to challenges such as reflective and transparent surfaces, which are much more widespread when reconstructing cars.

Second, the fusion process improves the reconstruction accuracy when using ELAS depth maps, but not when using DispNet. This is due to the more conservative nature of the results produced by ELAS, which tend to have gaps but to be sharper overall. The fusion process benefits from the sharpness while compensating for the occasional gaps.

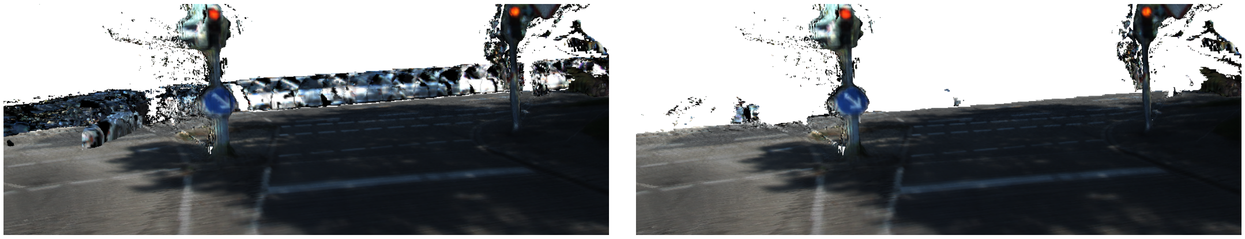
Third, the reconstruction accuracy of the dynamic objects is usually not improved by fusion when compared to the input maps, and the overall variance across frames is much higher. This follows from the fact that vehicles are considerably more challenging to reconstruct than, e.g., road surfaces, fences, and buildings, due to their non-lambertian properties,



(a) Voxel GC with $k_{\text{minAge}} = 6$.

(b) No voxel GC.

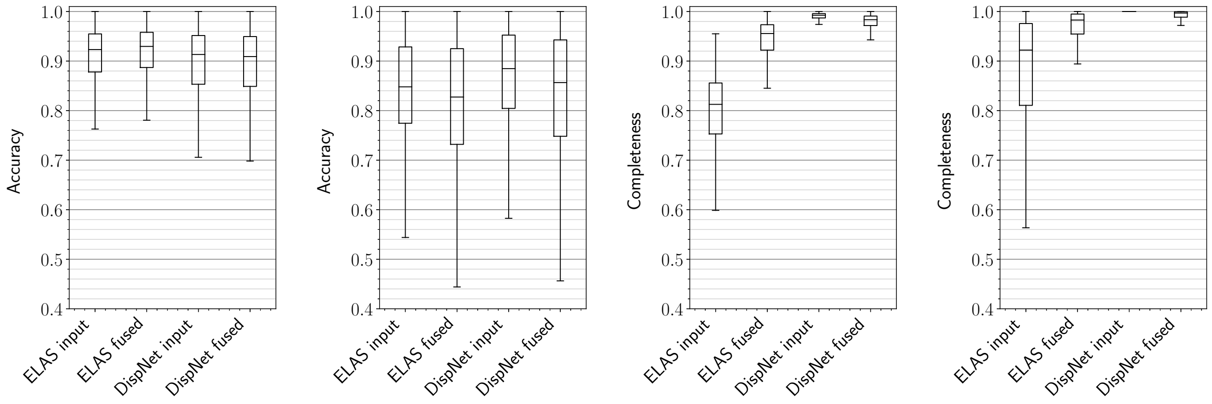
Fig. 4: Comparison of the same scene reconstructed with voxel garbage collection and without it.



(a) Static fusion is prone to corrupt the environment map with streaks and other artifacts produced by independently moving objects.

(b) Dynamic fusion, the primary operating mode of our system, prevents vehicle trails and leftover halos from being integrated into the map.

Fig. 5: Reconstructions produced by Dynamic and Standard fusion on KITTI tracking sequence 01.



(a) Reconstruction accuracy for the static background.

(b) Reconstruction accuracy for the dynamic objects.

(c) Reconstruction completeness for the static background.

(d) Reconstruction completeness for the dynamic objects.

Fig. 6: Aggregate reconstruction accuracy and completeness on the KITTI odometry benchmark training sequences.

as well as their independent motion.

Fourth, DispNet leads to much denser, i.e, more complete, reconstructions of both the static map and of the objects than ELAS, but this gap is reduced by volumetric fusion over multiple frames. This follows from the fact that ELAS depth maps are sparser, but more accurate, meaning that they benefit more from the fusion than DispNet.

Finally, the variance of the ELAS depth map completeness is high because it often produces very sparse results in challenging lighting conditions such as after exiting a tunnel, before the automatic exposure of the camera manages to adapt. Nevertheless, the magnitude of this effect is reduced significantly by the fusion process.

Additionally, we compare the reconstruction accuracy of our system to two baselines on the first ten sequences from the KITTI tracking benchmark. The first baseline is InfiniTAM itself, using its built-in ICP-based tracker. Naturally, this baseline does not perform well, as InfiniTAM by itself is not designed for outdoor environments and the fast camera motion encountered in the KITTI dataset. The next baseline is similar to the first, except that it uses `libviso2` for robust visual odometry. However, just like the first, it does not attempt to detect, remove, or reconstruct dynamic objects. The results of this comparison are presented in Table I.

As in the previous series of experiments, the ELAS-based results are more accurate than their DispNet-based counter-

TABLE I: Aggregate accuracy and completeness scores on the first 10 sequences from the KITTI tracking benchmark.

Method	Accuracy	Completeness
InfiniTAM + ELAS	tracking failure	tracking failure
InfiniTAM + DispNet	tracking failure	tracking failure
InfiniTAM + Robust VO + ELAS	0.921	0.974
InfiniTAM + Robust VO + DispNet	0.875	0.984
Our system + ELAS	0.923	0.961
Our system + DispNet	0.879	0.973

parts, but less complete. Moreover, our system outperforms the robust VO InfiniTAM baseline in terms of accuracy. Despite its lower accuracy, the robust VO baseline does lead to slightly higher completeness scores than dynamic fusion (our system). This is explained by the fact that object instance removal is not perfect, and tends to sometimes also lead to small areas of the background being removed along with the objects. While this does not affect the quality of the instance reconstructions, with the additional background fragments being prime candidates for voxel garbage collection, it does explain the slightly lower scores of the dynamic fusion in terms of completeness. These results showcase our method's ability to improve the quality of the static maps by actively preventing dynamic objects from corrupting them.

C. Effectiveness of fixed-lag map regularizer

We evaluate the impact of the map regularization on the system's memory consumption and reconstruction accuracy. The methodology for evaluating the accuracy is the same as in the previous section. In order to also capture the effect of the voxel garbage collection, which is performed in lockstep with the reconstruction, but with a fixed delay of k_{minAge} frames, we also add a delay to the evaluation. That is, at time t, the voxel garbage collection is processing the blocks visible at time $t - k_{\text{minAge}}$, and the evaluation is performed using the depth map and camera pose from $t - k_{\text{minAge}} - \tau$. The additional offset τ ensures that the map viewed by the camera at that time has been processed by the regularization. In our experiments, we set $\tau = k_{\text{minAge}}$.

Given the limitations of the ground truth, which is provided in the form of per-frame LIDAR readings, we only evaluate the accuracy of the static reconstructions under the effect of voxel garbage collection. The delayed evaluation scheme described above prevents us from also evaluating dynamic object reconstructions, as their positions are only known to the system while they are being observed, and not $k_{\text{minAge}} + \tau$ frames in the past.

To this end, we use the same semantic-aware evaluation scheme defined previously with the only difference being that we only evaluate the static map. We perform our experiments on the first 1000 frames of KITTI odometry sequence number 9, as it contains a small number of dynamic objects, while at the same time being diverse in terms of encountered buildings and vegetation. Even through we also use the regularization for the vehicle reconstructions, they only represent a small fraction of the system's overall memory usage. We therefore focus on evaluating the memory usage of the static map.

Fig. 7 illustrates the accuracy and completeness of the map reconstructions as a function of k_{weight}, the noise threshold, as well as the memory usage and an F1 score combining accuracy (A) and completeness (C) as

$$\mathrm{F}_1(\text{frame}) = 2 \cdot \frac{\mathrm{A}(\text{frame}) \cdot \mathrm{C}(\text{frame})}{\mathrm{A}(\text{frame}) + \mathrm{C}(\text{frame})}. \tag{1}$$

The maps produced using DispNet are denser than those based on ELAS, which is also reflected by the completeness of the reconstruction. At the same time, as in the previous experiments, despite being less accurate than DispNet according to the KITTI Stereo Benchmark [17][2], the depth maps produced by ELAS lead to more accurate, albeit less complete, reconstructions. This trend is maintained even with increasing k_{weight}. Based on the memory usage plot from Fig. 7, it becomes clear that the impact of the noise on the memory consumption of the reconstructions is strong. Even light regularization using $k_{\text{weight}} = 1$ or 2 can already reduce the memory footprint of a reconstruction by more than 30%, with only small costs in terms of discarded (useful) information.

For additional qualitative and quantitative results, we would like to direct the reader to the video and the supplementary results[3] accompanying this publication, which showcase the system's real-time operation, resulting map and object reconstructions, as well as the impact of the regularization strength on the resulting maps.

Our system is able to run in near-real time at around 2.5Hz on a PC equipped with an nVIDIA GeForce GTX 1080 GPU.

V. CONCLUSIONS

We presented a robust dense mapping algorithm for large-scale dynamic environments. Our algorithm employs a state-of-the-art deep convolution neural network for semantic segmentation, which separates the static background from the dynamic and potentially dynamic objects. The sparse scene flow is used to estimate both the camera egomotion, as well as the individual motions of dynamic objects encountered in the environment. The static background and dynamic objects are then reconstructed separately. Real-world experiments show that our algorithm can output high-quality dense maps for both static background and dynamic objects, which are important for high-level robotic decision making tasks such as path planning. To further benefit resource-constrained mobile robots, a map regularizer by means of voxel garbage collection is also proposed to reduce the overall memory consumption.

REFERENCES

[1] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[2] http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo

[3] Available on the project's web page at https://andreibarsan.github.io/dynslam.

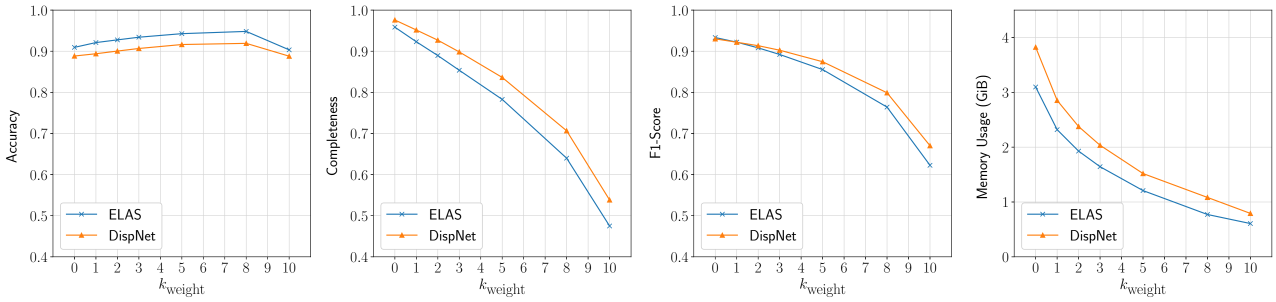


Fig. 7: Reconstruction accuracy, completeness, F1-score, and memory usage (in GiB) as functions of the regularization strength, comparing its effect on reconstructions using ELAS and DispNet depth maps. Larger values of k_{weight} correspond to more aggressive voxel garbage collection.

[2] Y. Furukawa and J. Ponce. Accurate, dense and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[3] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[5] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large scale stereo matching. In *Asian Conference on Computer Vision ACCV*, 2010.

[6] Andreas Geiger, Julius Ziegler, and Christoph Stiller. StereoScan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968. IEEE, jun 2011.

[7] Bernardes Vitor Giovani, Alessandro Correa Victorino, and Janito Vaqueiro Ferreira. Stereo Vision for Dynamic Urban Environment Perception Using Semantic Context in Evidential Grid. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015-Octob:2471–2476, 2015.

[8] Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Static-Map and Dynamic Object Reconstruction in Outdoor Scenes Using 3-D Motion Segmentation. *IEEE Robotics and Automation Letters*, 1(1):324–331, 2016.

[9] Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cedric Demonceaux. Static and dynamic objects analysis as a 3d vector field. In *HAL archive*, 2017.

[10] Deyvid Kochanov, Aljosa Osep, Jorg Stuckler, and Bastian Leibe. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (3):1785–1792, 2016.

[11] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1241–1250, 2015.

[12] P. Labatut, J. Pons, and R. Keriven. Efficient multi-view reconstruction of large scale scenes using interest points. In *International Conference on Computer Vision (ICCV)*, 2007.

[13] Philip Lenz, Julius Ziegler, Andreas Geiger, and Martin Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, volume 44, pages 926–932. IEEE, jun 2011.

[14] Peidong Liu, Lionel Heng, Torsten Sattler, Andreas Geiger, and Marc Pollefeys. Direct visual odometry for a fisheye-stereo camera. In *IROS*, 2017.

[15] Martin Rünz and Lourdes Agapito. Co-Fusion: Real-time Segmentation, Tracking and Fusion of Multiple Objects. *International Conference on Robotics and Automation*, pages 4471–4478, 2017.

[16] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, June 2016.

[17] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. pages 3061–3070, 2015.

[18] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[19] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011*, pages 127–136, 2011.

[20] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3D Reconstruction at Scale Using Voxel Hashing. *ACM Trans. Graph.*, 32(6), 2013.

[21] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[22] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[23] N. Dinesh Reddy, Prateek Singhal, Visesh Chari, and K. Madhava Krishna. Dynamic Body VSLAM with Semantic Constraints. *MSc Thesis*, pages 1897–1904, 2015.

[24] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys. 3D modeling on the go: Interactive 3D reconstruction of large-scale scenes on mobile devices. In *International Conference on 3DV*, 2015.

[25] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip HS Torr. Urban 3d semantic modelling using stereo vision. pages 580–585, 2013.

[26] F. Steinbruecker, J. Sturm, and D. Cremers. Volumetric 3D mapping in real-time on a CPU. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[27] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 75–82, May 2015.

[28] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1384, 2013.

[29] Trung-Dung Vu, Olivier Aycard, and Nils Appenrodt. Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments. *Intelligent Vehicles Symposium, 2007 IEEE*, pages 190–195, 2007.

[30] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, Thomas Whelan, John Mcdonald, Michael Kaess, Maurice Fallon, Hordur Johannsson, and John J Leonard. Kintinuous: Spatially Extended KinectFusion. 2012.

[31] Zhenfei Yang, Fei Gao, and Shaojie Shen. Real-time monocular dense mapping on aerial robots using visual inertial fusion. In *International Conference on Robotics and Automation (ICRA)*, 2017.

[32] Lv Zhaoyang. Kinfuseg: a dynamic slam approach based on kinect fusion. *Department of Computing, Imperial College London, London, England, Msc Thesis*, 2013.