SLAMBench 3.0: Systematic Automated Reproducible Evaluation of SLAM Systems for Robot Vision Challenges and Scene Understanding

Mihai Bujanca†, Paul Gafton†, Sajad Saeedi*, Andy Nisbet†, Bruno Bodin‡, Michael F.P. O'Boyle‡,
Andrew J. Davison*, Paul H.J. Kelly*, Graham Riley†, Barry Lennox†, Mikel Luján†, Steve Furber†
University of Manchester†, University of Edinburgh‡, Imperial College London*, UK
The first two authors have equal contribution, the order just reflects alphabetical order.

***Abstract*—As the SLAM research area matures and the number of SLAM systems available increases, the need for frameworks that can objectively evaluate them against prior work grows. This new version of SLAMBench moves beyond traditional visual SLAM, and provides new support for scene understanding and non-rigid environments (dynamic SLAM). More concretely for dynamic SLAM, SLAMBench 3.0 includes the first publicly available implementation of DynamicFusion, along with an evaluation infrastructure. In addition, we include two SLAM systems (one dense, one sparse) augmented with convolutional neural networks for scene understanding, together with datasets and appropriate metrics. Through a series of use-cases, we demonstrate the newly incorporated algorithms, visualisation aids, and evaluation metrics (6 new metrics, 4 new datasets and 5 new algorithms).**

1. Introduction

SLAM is becoming a key component of robotics and augmented reality. While a large number of SLAM algorithms have been published, there has been little effort to develop frameworks to enable researchers to perform complete comparisons of their capabilities. SLAMBench 2.0 [4] introduced a dataset-agnostic and sensor-agnostic framework for qualitative, quantitative and easily reproducible evaluation of SLAM systems with plug-and-play algorithm support. This was a significant step forward towards benchmarking SLAM systems, considering the number and variety of systems included.

The SLAMBench 2.0 framework provided a solid performance baseline as well as a good number of examples for including new algorithms. From Tables 1 and 2, we can see that SLAMBench covered key algorithms published up until 2016. However, since then new SLAM algorithms have appeared. At one end of the spectrum, we find BundleFusion [10] focusing on 3D reconstruction with RGB-D sensors, although requiring two powerful GPUs to run in real-time. At the other end, we encounter FLaME [23] focusing on MAVs and thus addressing low-power devices using monocular sensors, while still trying to offer a dense SLAM. One possible criticism of previous SLAMBench versions was the lack of coverage of SLAM systems for non-rigid environments (e.g. moving humans). Another important trend previously not supported was the growing importance of scene understanding within SLAM by taking advantage of Machine Learning techniques; chiefly Convolutional Neural Networks (CNNs).

When AlexNet [34] won the ImageNet [52] challenge in 2012, it paved the way for the rise of CNNs to become the standard tool for image processing for object labelling. The additional information held by multiple scene perspectives of an object [65] enables SemanticFusion [40] to label 3D scenes.

The insight behind *DynamicFusion* [45] is that SLAM systems operating under a static model of reality can suffer from localisation failures, and ultimately build corrupted or incomplete reconstructions. Since 2015, a number of systems have addressed real-time non-rigid 3D reconstruction in SLAM algorithms by modeling reality as a world where topological/deformable changes are allowed.

Given the increasingly wider range of datasets and algorithms, it is often complex and time-consuming to compare new algorithms with existing ones. As SLAM systems become increasingly diverse and new challenges emerge, better tools are needed for easy and efficient benchmarking. This paper presents the following contributions of SLAMBench3.0, which encompasses a wider set of SLAM algorithms, datasets, and metrics:

- BundleFusion [10] integration, which provides state-of-the-art 3D reconstruction, along with new metrics and visualizations to measure the accuracy of the algorithm qualitatively and quantitatively.
- Metrics to assess depth prediction quality are incorporated and demonstrated on FLaME [23] (see section 3.2), which optimises its depth prediction algorithm for computationally constrained platforms.
- Enhancements that allow joint evaluation of quality for both reconstruction and semantic segmentation, using the NYU RGB-Dv2 [44] and ScanNet [8] datasets. This is demonstrated on two systems: SemanticFusion [40] and ORB-SLAM2-CNN [53], which construct labelled dense and sparse scene maps, respectively.
- An implementation, along with evaluation infrastructure for DynamicFusion [45], the first real-time non-rigid 3D reconstruction system.

2. Background and Related Work

SLAM is the problem of enabling a mobile robot to simultaneously build a consistent map and determine its location within that map, when placed at an unknown position in an unknown environment. We briefly review the variety of current SLAM algorithms together with benchmarking tools and datasets.

Simultaneous Localisation And Mapping — SLAM has benefited from increased attention after the release of MonoSLAM [11], one of the first accurate real-time monocular SLAM systems. PTAM [32] presented a system capable of mapping small spaces using FAST corners [50], and demonstrated one of the first markerless augmented reality applications using their system. Recent relevant works include LSD-SLAM [17] which performs semi-dense mapping, ORB-SLAM [42] which performs sparse mapping using ORB [51] features, and OKVIS [36], based on BRISK [35] features. New developments in the area of monocular SLAM address dense depth estimation: CNN-SLAM [61] and CodeSLAM [3] use learning-based approaches with good results, but require relatively powerful hardware. Meanwhile, FLaME [23] uses a graph-based optimisation process, achieving remarkable low-latency on computationally-constrained devices, but trading accuracy for speed.

The introduction of Kinect and other consumer depth cameras enabled researchers to work on producing dense, accurate 3D maps. KinectFusion [46] introduced an algorithm that estimates the pose of a moving sensor and uses the Truncated Signed Distance Function (TSDF) [7] to store the reconstruction. An improvement to this technique is VoxelHashing [47], which proposes a hierarchical hashing approach to store and access voxels. Recent developments include BundleFusion [10], which performs on-the-fly surface reintegration in real-time. This algorithm, described in Section 3.1, represents the state-of-the-art in terms of reconstruction quality.

Semantic SLAM — Semantic labelling may be solved more accurately by considering additional information from multiple views of an individual object within a scene. Furthermore, being able to correctly recognise objects within a scene may provide a vital context for a robot that needs to navigate a 3D scene. Performing online semantic segmentation along with 3D reconstruction has been actively studied in the past few years: [18], [27], [28], [39], [62] [9]. SemanticPaint [22], [64] employs InfiniTAMv2 [31] to perform live reconstruction and DenseCRF [33] for segmentation. Vineet *et al*. [65] presents a system for large-scale semantic reconstruction, with impressive results on the KITTI dataset [21]. Nguyen *et al*. [63] proposes an annotation tool that integrates both 2D and 3D segmentation, while providing a means to correct any inaccuracies the automatic segmentation system might have produced. SemanticFusion [40] is built upon ElasticFusion [67] to perform SLAM and extends the image semantic segmentation CNN proposed by [48].

Datasets for training and evaluating semantic segmentation in 2D and 3D scenarios are increasingly common. Sun3D [68] contains 8 annotated sequences, in different spaces. NYU RGB-Dv2 [44], and SceneNet RGB-D [41] provide pixel-level annotations. ScanNet [8] holds 2.5 million RGB-D views annotated with 3D camera poses, surface reconstructions, and semantic labels. On the other hand, Armeni *et al*. [2] provides a dataset of RGB-D with instance-level semantic labels.

Dynamic SLAM — DynamicFusion [45] proposes the first system able to capture non-rigidly deforming scenes in real-time. VolumeDeform [30] improves on this technique by computing SIFT [38] features to improve frame alignment. Guo *et al*. [25] introduces a pipeline that uses shading information of dynamic scenes to improve the non-rigid registration and temporal correspondences to estimate surface appearance. KillingFusion [57] and SobolevFusion [58] use displacement vectors directly on the TSDF volume, rather than explicit correspondences. BodyFusion [69] fits a skeleton template for tracking, while HybridFusion [70] uses eight inertial measurement units attached to the reconstructed subject. SurfelWarp [20] employs surfels rather than a TSDF volume and a deformation graph similar to DynamicFusion for computing correspondences. Fusion4D [14] and [13] achieve impressive results wielding complex setups that involve four stereo-camera sensors positioned around a moving subject.

Finally, Wasenmüller *et al*. [66] notes the lack of benchmarking tools and datasets for non-rigid reconstruction makes quantitative evaluations and comparisons with other algorithms difficult.

Benchmarks — An analysis of the literature shows that SLAM algorithms and datasets are increasingly diverse and complex, and often address a wide spectrum of related problems. Unfortunately, this variety comes at the cost of having to deal with different, sometimes not directly compatible interfaces for datasets or for comparing against other algorithms.

Previously, benchmarking tools such as the KITTI Benchmark Suite [21] and TUM RGB-D benchmark [59] have been used to evaluate performance of SLAM systems. More recently, open-source approaches to benchmarking have appeared. EVO [24] is a framework for evaluating visual odometry algorithms integrating a few relevant datasets. In [1] four SLAM systems were compared, and recently a visual-inertial comparative study of several algorithms was preformed in [12]. Similarly, tools exist for deep learning and AI tasks such as DAWNBench [6].

SLAMBench2 [4] was the first framework to integrate a variety of algorithms, datasets, and metrics, providing users with the necessary tools to effortlessly compare traditional SLAM systems. Given the diversity of techniques (geometric, semantic, and dynamic SLAM), a benchmarking framework that incorporates these concepts is needed. To this end, SLAMBench 3.0 addresses this issue by integrating semantic and dynamic SLAM, as well as depth prediction, including appropriate metrics, datasets, and example algorithms.

Algorithm	Type	Sensors	Implementations	Year
ORB-SLAM [42]	Sparse	RGB-D, Stereo, Monocular	C++	2016
OKVIS [36]	Sparse	Stereo, IMU	C++	2015
SVO [19]	Sparse	Monocular	C++	2014
MonoSLAM [11]	Sparse	Monocular	C++, OpenCL	2007
PTAM [32]	Sparse	Monocular	C++	2007
BundleFusion [10]*	Dense	RGB-D	CUDA	2016
ElasticFusion [67]	Dense	RGB-D	CUDA	2015
InfiniTAM [31]	Dense	RGB-D	C++, OpenMP, CUDA	2015
KinectFusion [46]	Dense	RGB-D	C++, OpenMP, OpenCL, CUDA	2011
LSD-SLAM [17]	Semi-Dense	Monocular	C++, PThread	2014
SemanticFusion [40]*	Dense, semantic	RGB-D	CUDA	2016
ORB-SLAM2-CNN [53]*	Sparse, semantic	Monocular	C++	2018
DynamicFusion [45]*	Dense, non-rigid	RGB-D	CUDA	2015
FLaME [23]*	Depth estimation	Monocular	C++	2017

TABLE 1: SLAM algorithms included. * denotes algorithms introduced in SLAMBench 3.0

Name	Sensors	Trajectory	3D Point Cloud	2D semantic labels	Non-rigid	Synthetic
ICL-NUIM [26]	RGB-D	Yes	Yes	No	No	Yes
TUM RGB-D [59]	RGB-D, IMU	Yes	No	No	No	No
InteriorNet [37]	RGB-D, IMU	Yes	Yes	Yes	No	Yes
ICL [54]	RGB-D, IMU	Yes	Yes	No	No	Yes
EuRoC MAV [5]	Stereo, IMU	Yes	Yes	No	No	No
NYU RGB-Dv2* [44]	RGB-D	No	Yes	Yes	No	No
ScanNet* [8]	RGB-D	Yes	Yes, semantic	Yes, partial	No	No
VolumeDeform* [30]	RGB-D	No	Yes	No	Yes	No
Elanttil *et al** [16]	RGB-D	No	Yes	No	Yes	Yes

TABLE 2: Datasets provided by our benchmark suite. * denotes datasets introduced in SLAMBench 3.0.

3. SLAMBench 3.0

SLAMBench 2.0 is a dataset-agnostic and sensor-agnostic framework for qualitative, quantitative and easily reproducible evaluation of SLAM systems with plug-and-play algorithm support [4]. Tables 1 and 2 summarise the algorithms and datasets included.

The SLAMBench framework is structured into four core components. The *I/O component* defines a straightforward unified format that supports a variety of sensors and *ground-truth* formats. The *API component* provides a generic interface for SLAM algorithms integration, with functions for configuration, processing and output extraction. The *Metrics component* provides a robust infrastructure for comparing the output of the algorithms with the ground-truth and extracting relevant quantitative metrics. The *UI component* allows loading the inputs, outputs and ground-truth of a running SLAM towards a visualisation pipeline for qualitative evaluation. The first version of SLAMBench [43] only included one algorithm, KinectFusion [46] with the above components directly integrated with the SLAM system, rather than provided as external modules. On the other hand, it provided and benchmarked multiple implementations of KinectFusion (C++, OpenMP, OpenCL and CUDA).

The framework has now matured significantly: SLAMBench 3.0 introduces new algorithms (Bundle Fusion, SemanticFusion, ORB-SLAM2-CNN, DynamicFusion, and FLaME), along with new datasets and metrics.

3.1. Bundle Fusion

BundleFusion [10] is a recent SLAM algorithm that obtains accurate dense scene reconstructions from RGB-D input. At its core, there is a coarse-to-dense hierarchical tracking algorithm, which continuously optimizes the global trajectory. The accuracy of the tracking system, along with a novel technique for reintegrating frames into the reconstruction when a better estimate of their position is available, leads to high quality dense reconstructions.

The SLAMBench API has been updated to be able to benchmark BundleFusion and its contributions. Extensions have been added for evaluating not only the initial pose estimate, but the full optimized trajectory at every frame. We have added a point cloud metric for online evaluation of the reconstruction, as opposed to the offline reconstruction error accuracy introduced by SLAMBench 2.0. Based on it, we generate an evolving heat-map of the reconstruction, as shown in Figure 4.

3.2. FLaME (Fast Lightweight Mesh Estimation)

Low-power devices that lack depth cameras such as drones and mobile phones can benefit from monocular depth estimation when performing a variety of tasks such as path planning and augmented reality. We integrate the public implementation of FLaME [23], a system for dense monocular depth estimation built upon a graph-based variational optimization framework, tailored for computationally-constrained platforms. Unlike previous approaches, FLaME provides surface prediction at every frame, rather than using

keyframes, which is ideal for low-latency applications. The algorithm builds a Delaunay graph over a set of features tracked across multiple frames, continuously optimizing the depth estimation as data from different perspectives is revealed. The system facilitates control over the trade-off between accuracy and speed via a hyperparameter.

Given the uniqueness of how depth is calculated and the identified trend to use CNNs for this purpose, SLAMBench 3.0 introduces the following metrics proposed by Eigen *et al*. [15] and used widely in the literature to evaluate depth prediction:

absolute relative difference – normalised sum of differences between ground-truth and estimated distances)

$$\frac{1}{|T|}\sum_{y\in T}|y-y^*|/y^*;$$

accurate depth percentage – percentage of accurate pixels within threshold, per frame.)

$$max\left(\frac{y_i}{y_i^*},\frac{y_i^*}{y_i}\right)=\delta<thr, thr\in\{1.25,1.25^2,1.25^3\}.$$

Table 6 shows an accuracy analysis using these metrics.

3.3. Semantic SLAM

Semantic SLAM algorithms assist with scene understanding by producing a labelled map of the environment. Labels identify the classes of objects present in a scene. We describe two semantic algorithms and evaluation datasets, followed by the proposed metrics for benchmarking.

Algorithms — SemanticFusion [40] is an algorithm that produces labelled dense 3D reconstructions. It contains a CNN, based on the work of Noh *et al*. [49], which performs frame-by-frame segmentation and runs in parallel with ElasticFusion [67]. The CNN was trained on the NYU RGB-Dv2 [44] dataset, using 13 semantic classes. The 2D predictions of this neural network are projected onto the map of ElasticFusion and fused with the existing geometry.

Similar to SemanticFusion, ORB-SLAM2-CNN [53] is based on ORB-SLAM2 [42], projecting the segmentation of a modified version of MobileNet [29] to label the keypoints of ORB-SLAM2-generated map. Thus, the key difference from SemanticFusion is that this algorithm produces a labelled sparse map, rather than a dense one. In Section 4, these two algorithms are compared using the following datasets and metrics.

Datasets — In order to facilitate the evaluation of semantic SLAM algorithms, new datasets with semantic ground-truth labels have been added. *NYU RGB-Dv2* [44] contains 464 indoor scenes from 26 different environments. All the scenes of this dataset contain a few densely labelled frames which can serve as ground-truth in evaluating the accuracy of a segmentation algorithm. The dataset features very fine semantic labelling, with over 894 different semantic labels in 1449 labelled frames.

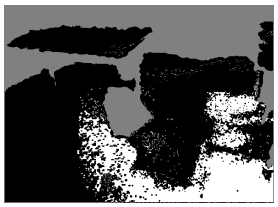
While *NYU RGB-Dv2* has densely labelled 2D frames, it lacks three dimensional ground-truth. To complement it, SLAMBench 3.0 integrates *ScanNet* [8] a RGB-D dataset containing 1513 partially annotated indoor scenes with 3D camera poses. ScanNet features surface reconstructions generated by BundleFusion [10], hand-labelled semantic segmentations, and partially labelled frames with 1163 semantic classes.

NYU RGB-Dv2 and ScanNet have been used by SemanticFusion and ORB-SLAM2-CNN respectively, to train the CNNs that perform the labelling, and test their algorithms. Thus, we can reproduce the experiments, by testing each algorithm on the same dataset that has been used for training the neural networks, and we can show how well the CNN generalizes when faced with a different dataset with the same type of scenes.

Metrics — Additional metrics have been added to the SLAMBench framework for measuring the segmentation accuracy. Since a common form of semantic ground-truth are labelled frames, we have introduced infrastructure for evaluating the accuracy of the segmentation by comparing the reprojected segmentation at a given camera pose against the ground-truth labelled frame. Similar techniques are widely used in the literature, including in SemanticFusion and ORB-SLAM2-CNN.

By comparing the ground-truth with a reprojection of the 3D segmentation, we can evaluate the segmentation both qualitatively and quantitatively. The *Pixel accuracy* metric represents the proportion of correctly labelled pixels out of the total number of labelled pixels. The matching of the predicted labels to the ground-truth can be visualized within our framework, as shown in Figure 1. This visualization, along with the *Confusion matrix* of the segmented projection provides valuable insight into understanding the performance of the algorithm.

(a) Frame 301 of the NYU RGB-Dv2 bathroom_0003 sequence



(b) Frame 535 of the ScanNet sequence 187

Figure 1: Projected segmentation of SemanticFusion onto the ground-truth labelling. The white pixels are correctly labelled, the black pixels are mislabeled, and the grey pixels are not labeled in the ground-truth or by the algorithm.

3.4. Dynamic SLAM

SLAMBench 3.0 provides new infrastructure for evaluating SLAM systems capable of reconstructing objects that demonstrate non-rigid movement, such as humans. In this context, we consider non-rigid movement to be any change in topology over time, as opposed to only rotation and translation.

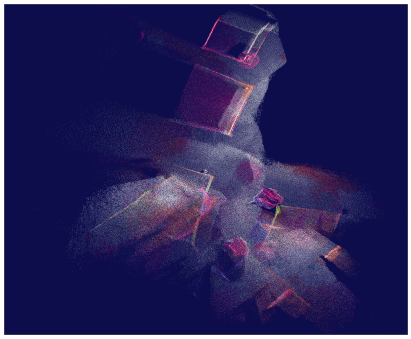
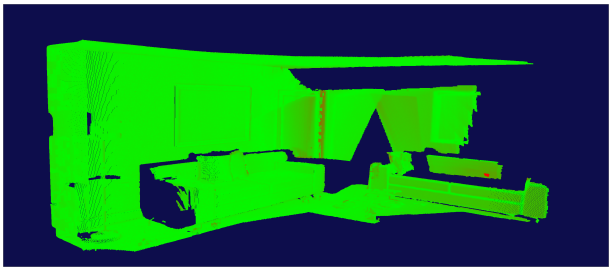
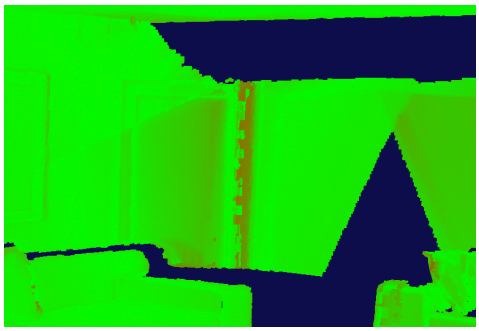


Figure 2: The semantic point cloud generated from the mesh of SemanticFusion, visualized within SLAMBench 3.0. The color scheme is the original color scheme of the algorithm.

Figure 3: Example reconstructions sequence from the upperbody sequence after frames 0, 100 and 200.



Heat-map of frame 1066 of the living room 2 sequence



Zoomed in heat-map of frame 1066

Figure 4: A heat-map showing the errors in the reconstruction of BundleFusion. Green indicates small errors and red indicates large errors.

To the best of our knowledge, no implementation of any real-time non-rigid reconstruction system is publicly available. We contribute an open-source implementation of DynamicFusion [45], the first real-time non-rigid reconstruction algorithm, to serve as a baseline for future evaluation and benchmarking.

DynamicFusion generalises the KinectFusion [46] pipeline to work in dynamic settings. The reconstruction is represented by a canonical, rigid model and a coarse warp field controlled by a sparse deformation graph [60], which is used for transforming the canonical model into the live frame.

3.4.1. Datasets. The lack of publicly-available code and datasets makes the evaluation of non-rigid reconstruction algorithms challenging. Creating a reliable ground-truth reconstruction is inherently difficult as it requires specialised, costly devices such as motion capture systems. In contrast, synthetic datasets contain reliable ground-truth, but often have unrealistic inputs that do not simulate real-world conditions accurately.

To address this issue we introduce two datasets for evaluating non-rigid reconstruction into SLAMBench. VolumeDeform [30] contributes a dataset containing eight RGB-D sequences with non-rigidly moving objects, captured using a PrimeSense sensor. For each sequence, the dataset offers several texture-mapped meshes extracted every 100 frames using their non-rigid reconstruction algorithm, which extends the DynamicFusion pipeline by using SIFT features to improve speed and accuracy. Secondly, Elanttil *et al*. [16] provide a synthetic dataset with two scenes containing RGB-D inputs and ground-truth reconstructions at every frame. For each scene, there are four sequences with 253 frames each, using different trajectories of a camera moving around a non-rigidly moving subject. One of the significant advantages of the synthetic dataset is that it provides ground-truth pose, allowing separate evaluation of non-rigid registration and camera pose estimation.

4. Experiments

The experiments with SLAMBench 2.0 [4] showed that ORB-SLAM2 was a good overall SLAM system when considering execution time, memory and accuracy. In these experiments, we focus mainly on the new metrics introduced and not on execution time.

BundleFusion — We compare BundleFusion against previous SLAM algorithms on multiple trajectories of the living room sequence in the ICL-NUIM [26] dataset, a synthetic dataset with accurate ground-truth for the 3D geometry of the scene and trajectory. The results of the experiments listed in Table 4 show that the quality of the BundleFusion reconstruction either matches or outweighs both the dense and the sparse systems we have been compared it against.

Semantic SLAM — The labelling accuracy of SemanticFusion and ORB-SLAM2-CNN has been measured by comparing a projection of the segmented geometry with a labelled frame on both ScanNet and NYU RGB-Dv2 datasets. While for the former we report an average accuracy over multiple frames in the sequence, for the latter we only report the accuracy of specific frames where

Sequence	Frame number	SemanticFusion Pixel accuracy	ORB-SLAM2-CNN Pixel accuracy
NYU RGB-Dv2 office_0003	271	74.34%	N/A
NYU RGB-Dv2 office_0005	31	75.54%	0%
NYU RGB-Dv2 bathroom_0003	301	63.66%	5.96%
ScanNet scene_0187	100-1600	26.57% (max 54.79%)	77.09% (max 100%)
ScanNet scene_0423	100-400	25.16% (max 44.9%)	22.28% (max 68.15%)

TABLE 3: Comparison between SemanticFusion and ORB-SLAM2-CNN.

Sequence	Frames	Algorithm	Reconstruction error
living_room_traj_1	1-950	ORB-SLAM2	0.112
		ElasticFusion	0.2
		BundleFusion	0.0172
living_room_traj_2	1-825	ORB-SLAM2	0.0392
		ElasticFusion	0.135
		BundleFusion	0.0192
living_room_traj_3	1-1000	ORB-SLAM2	0.0211
		ElasticFusion	0.105
		BundleFusion	0.0258

TABLE 4: Reconstruction error (m) – BundleFusion, ElasticFusion, and ORB-SLAM2.

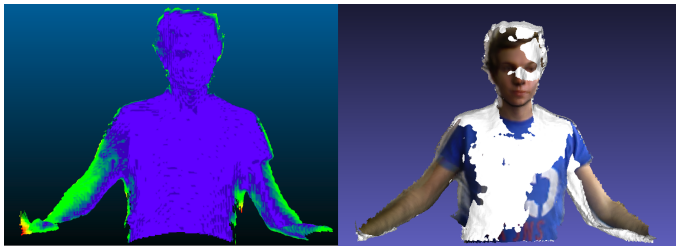


Figure 5: Visualisation of reconstruction error (left). Blue represents negligible error, green is small error while red is large error. DynamicFusion reconstruction (white) overlayed on ground-truth (textured), on the right.

ground-truth is available in the dataset. Table 3 presents the results. Comparison against two datasets shows the ability of each algorithm to generalize, since we have not retrained their neural networks. SemanticFusion was trained on NYU RGB-Dv2 and ORB-SLAM2-CNN on ScanNet. Our experiments have shown that SemanticFusion has a greater capability of generalization, achieving much higher results on a dataset it has not been trained for than ORB-SLAM2-CNN. One of the reasons for this is the generality of the classes, SemanticFusion having general classes such as "object" or "furniture", while in ORB-SLAM2-CNN the classes tend to be more specific.

DynamicFusion — We evaluate our implementation of the DynamicFusion system on the VolumeDeform dataset, using the *reconstruction error* metric [67] for quantitative measurements, as well as qualitatively using an error heat map visualisation. We use a volume size of 256^3 and a sampling decimation of *15mm* in our implementation to extract reconstructions every 100 frames and present the average *reconstruction error* across each sequence.

As illustrated in Table 5, DynamicFusion and VolumeDeform produce similar reconstructions on sequences with simple movement such as *upperbody*, *hoodie* and *boxing*. In sequences such as *calendar* and *shirt*, VolumeDeform performs better thanks to tracking visual features.

	Reconstruction Error
Boxing	0.00748
Calendar	0.01743
Hoodie	0.00946
Minion	0.01097
Shirt	0.02160
Sunflower	0.01185
Umbrella	tracking failure
Upperbody	0.01422

TABLE 5: Dynamic SLAM – VolumeDeform Dataset

FLaME — FLaME is an algorithm designed for low-power devices on MAVs. We use the *rgbd_dataset_freiburg1_room* sequence (TUM RGB-D [59]) to evaluate the accuracy of the depth prediction. The results are shown in Table 6.

Metric	Value
Absolute Relative Difference	0.128747
Accurate depth ($\delta = 1.25$)	40.17%
Accurate depth ($\delta = 1.25^2$)	51.55%
Accurate depth ($\delta = 1.25^3$)	56.04%

TABLE 6: FLaME – Depth estimation quantitative results on TUM RGB-D *rgbd_dataset_freiburg1_room*.

5. Conclusion

We have presented a benchmarking suite that goes beyond traditional SLAM algorithms and integrates relevant algorithms, datasets and metrics for evaluating related problems. SLAMBench 3.0 provides means for evaluating semantic segmentation, depth estimation and non-rigid reconstruction in the context of SLAM. We hope that providing open-source implementations of some of the most recent methods in an unified framework, as well as the first public implementation of a non-rigid real-time reconstruction system will help researchers in performing comparative studies and evaluating their work. In total, SLAMBench 3.0 contains 6 new metrics, 4 new datasets and 5 new algorithms. In future, we plan to include trajectory difficulty metrics such as the metrics described in [54] and [55], and also to extend this paper to include systems that use multiple cameras [56].

Acknowledgments

This research is supported by the EPSRC, grant PAMELA EP/K008730/1 and RAIN Hub EP/R026084/1.

References

[1] M. Abouzahir, A. Elouardi, R. Latif, S. Bouaziz, and A. Tajer. Embedding SLAM algorithms: Has it come of age? *Robotics and Autonomous Systems*, 100:14 – 26, 2018.

[2] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017.

[3] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. Codeslam-learning a compact, optimisable representation for dense visual slam. *arXiv preprint arXiv:1804.00874*, 2018.

[4] B. Bodin, H. Wagstaff, S. Saeedi, L. Nardi, E. Vespa, J. H. Mayer, A. Nisbet, M. Luján, S. Furber, A. J. Davison, et al. Slambench2: Multi-objective head-to-head benchmarking for visual slam. *arXiv preprint arXiv:1808.06820*, 2018.

[5] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.

[6] C. A. Coleman, D. Narayanan, D. Kang, T. Zhao, J. Zhang, L. Nardi, P. Bailis, K. Olukotun, C. Ré, and M. Zaharia. Dawnbench : An end-to-end deep learning benchmark and competition. 2017.

[7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[9] A. Dai and M. Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[10] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *arXiv preprint arXiv:1604.01093*, 2016.

[11] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1052–1067, 2007.

[12] J. Delmerico and D. Scaramuzza. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2502–2509, 2018.

[13] M. Dou, P. Davidson, S. R. Fanello, S. Khamis, A. Kowdle, C. Rhemann, V. Tankovich, and S. Izadi. Motion2fusion: real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6):246, 2017.

[14] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 35(4):114:1–114:13, July 2016.

[15] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[16] S. Elanattil, P. Moghadam, S. Sridharan, C. Fookes, and M. Cox. Non-rigid reconstruction with a single movingrgb-d camera. *arXiv preprint arXiv:1805.11219*, 2018.

[17] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[18] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard. Toward lifelong object segmentation from change detection in dense rgb-d maps. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 178–185. IEEE, 2013.

[19] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[20] W. Gao and R. Tedrake. Surfelwarp: Efficient non-volumetric single view dynamic reconstruction.

[21] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[22] S. Golodetz, M. Sapienza, J. P. Valentin, V. Vineet, M.-M. Cheng, A. Arnab, V. A. Prisacariu, O. Kähler, C. Y. Ren, D. W. Murray, et al. Semanticpaint: A framework for the interactive segmentation of 3d scenes. *arXiv preprint arXiv:1510.03727*, 2015.

[23] W. N. Greene and N. Roy. Flame: Fast lightweight mesh estimation using variational smoothing on delaunay graphs. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 4696–4704. IEEE, 2017.

[24] M. Grupp. Evo. https://github.com/MichaelGrupp/evo, 2018.

[25] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu. Real-time geometry, albedo and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (TOG)*, 2017.

[26] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, pages 1524–1531, Hong Kong, China, May 2014.

[27] E. Herbst, P. Henry, and D. Fox. Toward online 3-d object segmentation and mapping. 2014.

[28] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2631–2638. IEEE, 2014.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[30] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. *arXiv preprint arXiv:1603.08161*, 2016.

[31] O. Kahler, P. V. Adrian, R. C. Yuheng, X. Sun, P. Torr, and D. 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.

[32] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[33] V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. 2011.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[35] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[36] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[37] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger. InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018.

[38] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[39] A. Martinovic, J. Knopp, H. Riemenschneider, and L. Van Gool. 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2015.

[40] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. *arXiv preprint arXiv:1609.05130*, 2016.

[41] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet RGB-D: 5m photorealistic images of synthetic indoor trajectories with ground truth. *CoRR*, abs/1612.05079, 2016.

[42] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *arXiv preprint arXiv:1610.06475*, 2016.

[43] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. Kelly, A. J. Davison, M. Luján, M. F. O'Boyle, G. Riley, et al. Introducing slambench, a performance and accuracy benchmarking methodology for slam. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5783–5790. IEEE, 2015.

[44] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[45] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[46] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011.

[47] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013.

[48] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[49] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *CoRR*, abs/1505.04366, 2015.

[50] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[51] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[53] S. Saeedi, B. Bodin, H. Wagstaff, A. Nisbet, L. Nardi, J. Mawer, N. Melot, O. Palomar, E. Vespa, T. Spink, C. Gorgovan, A. Webb, J. Clarkson, E. Tomusk, T. Debrunner, K. Kaszyk, P. Gonzalez-De-Aledo, A. Rodchenko, G. Riley, C. Kotselidis, B. Franke, M. F. P. O'Boyle, A. J. Davison, P. H. J. Kelly, M. Lujn, and S. Furber. Navigating the landscape for real-time localization and mapping for robotics and virtual and augmented reality. *Proceedings of the IEEE*, 106(11):2020–2039, 2018.

[54] S. Saeedi, E. Carvalho, W. Li, D. Tzoumanikas, S. Leutenegger, P. H. J. Kelly, and A. J. Davison. Characterizing visual localization and mapping datasets. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[55] S. Saeedi, L. Nardi, E. Johns, B. Bodin, P. Kelly, and A. Davison. Application-oriented design space exploration for SLAM algorithms. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5716–5723, 2017.

[56] S. Saeedi, M. Trentini, M. Seto, and H. Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[57] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences.

[58] M. Slavcheva, M. Baust, and S. Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2646–2655, 2018.

[59] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[60] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. In *ACM Transactions on Graphics (TOG)*, volume 26, page 80. ACM, 2007.

[61] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction.

[62] K. Tateno, F. Tombari, and N. Navab. Real-time and scalable incremental segmentation on dense slam. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4465–4472. IEEE, 2015.

[63] D. Thanh Nguyen, B.-S. Hua, L.-F. Yu, and S.-K. Yeung. A robust 3d-2d interactive tool for scene segmentation and annotation. *Computing Research Repository (CoRR)*, 2016.

[64] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics (TOG)*, 34(5):154, 2015.

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

[66] O. Wasenmller, B. Schenkenberger, and D. Stricker. Towards non-rigid reconstruction - how to adapt rigid rgb-d reconstruction to non-rigid movements? In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2017.

[67] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. Elasticfusion: Dense slam without a pose graph. *Proc. Robotics: Science and Systems, Rome, Italy*, 2015.

[68] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *2013 IEEE International Conference on Computer Vision*, pages 1625–1632, Dec 2013.

[69] T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *The IEEE International Conference on Computer Vision (ICCV)*. ACM, October 2017.

[70] Z. Zheng, T. Yu, H. Li, K. Guo, Q. Dai, L. Fang, and Y. Liu. Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus. In *European Conference on Computer Vision (ECCV)*, Sept 2018.