FullFusion

Citation for published version (APA):
Bujanca, M., Luján, M., & Lennox, B. (2019). FullFusion: A Framework for Semantic Reconstruction of Dynamic Scenes. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*

Published in:
The IEEE International Conference on Computer Vision (ICCV) Workshops

FullFusion: A Framework for Semantic Reconstruction of Dynamic Scenes

Mihai Bujanca Mikel Luján Barry Lennox
The University of Manchester, Manchester, UK

Abstract

Assuming that scenes are static is common in SLAM research. However, the world is complex, dynamic, and features interactive agents. Mobile robots operating in a variety of environments in real-life scenarios require an advanced level of understanding of their surroundings. Therefore, it is crucial to find effective ways of representing the world in its dynamic complexity, beyond the geometry of static scene elements.

We present a framework that enables incremental reconstruction of semantically-annotated 3D models in dynamic settings using commodity RGB-D sensors. Our method is the first to perform semantic reconstruction of non-rigidly deforming objects along with a static background. FullFusion is a step towards enabling robots to have a deeper and richer understanding of their surroundings, and can facilitate the study of interaction and scene dynamics.

To showcase the potential of FullFusion, we provide a quantitative and qualitative evaluation on a baseline implementation which employs specific reconstruction and segmentation pipelines. It is, however, important to highlight that the modular design of the framework allows us to easily replace any of the components with new or existing counterparts.

1. Introduction

One of the important prerequisites for building intelligent embodied systems is creating means of interpreting and organising sensory information. Simultaneous Localisation and Mapping (SLAM) is one of the fundamental problems in modern computer vision, with applications ranging from Augmented Reality (AR) to 3D reconstruction, autonomous driving, robot localisation, and motion capture. In its simplest form, SLAM is the problem of simultaneously building a consistent map and determining the camera location within that map, with no prior information about the initial position or the environment. Recent years have seen SLAM systems evolve to handle a broader number of real-world applications [5][48]. Ultimately, the goal is to provide robots with ways of efficiently representing, understanding, and navigating diverse and challenging environments. To achieve this, researchers started extending SLAM solutions to solve related problems such as 3D reconstruction of non-rigidly deforming objects [28], and incorporating semantic information [26].

Most work on 3D reconstruction assumes the environment to be static. In reality, the world is highly dynamic and interactive. Furthermore, for a wide range of applications, successfully modelling the scene dynamics and interactions is central to decision making and data acquisition. Emerging technologies such as Augmented Reality (AR) headsets (*e.g.* Microsoft Hololens, Magic Leap One, Google Glass) aim to enable virtual telepresence, or “Holoportation”, as well as other applications such as remote inspection. Meanwhile, in autonomous systems, decisions need to be taken based on modelling the present state of the scene and making predictions. In the context of these applications, capturing only the static aspects of a scene is severely limiting.

Semantic labelling of dense reconstructions facilitates a shared understanding of the environment between humans and machines, thus opening up new possibilities for meaningful interaction. For instance, semantic information may be used by humans for queries: “How many students are there in the classroom?”, or providing textual or vocal commands: “Close the rightmost valve”. We show that semantic scene labels not only embed desirable information in 3D reconstruction systems but can additionally be used as priors to select appropriate reconstruction mechanisms depending on the properties of the objects present in the scene.

The problem of reconstructing the geometry of static scenes is well studied, with most solutions adding a dense representation of the map to SLAM systems. One of the crucial processes is estimating the camera pose, which in turn enables data association and fusing data into the map. This requires finding a single transformation that models camera displacement and rotation at every frame. In the case of non-rigid reconstruction, the problem becomes significantly more complex, as it involves estimating not only the camera movement but also the movement in the scene, thus necessitating thousands of transformations to be computed at every frame. Moreover, ambiguities such as occlusions prompt the use of regularisation techniques in order to ensure coherence across frames. Due to the computational demands which arise from such complexity, current non-rigid reconstruction systems suffer from severe scalability limitations.

Until recently, RGB-D SLAM systems could only reconstruct either large, predominantly static spaces, or a single non-rigidly deforming agent. MixedFusion [44], the most similar method to ours to date, incorporates both static and dynamic scene reconstruction by decoupling camera and scene motion estimation. Their approach uses a Sigmoid-based Iterative Closest Point (S-ICP) function, which separates the input into static and dynamic parts. Our method differs in a few aspects: rather than a tightly-coupled system, we propose a generic, modular framework, in order to easily employ different subsystems depending on the application. Secondly, unlike MixedFusion, we use a joint geometric and semantic formulation, which allows us to reliably segment dynamic objects from the first frame they are observed in. Finally, FullFusion not only uses semantics for segmentation, but also integrates the semantic information into the 3D volume.

This paper claims the following contributions:

1. A modular framework that reconstructs the geometric and semantic aspects of dynamic scenes.
2. A segmentation module based on scene semantics and 3D geometry.
3. State-of-the-art performance in trajectory estimation.

2. Related Work

2.1. Static 3D scene reconstruction

Following the release of the Kinect device, KinectFusion [29] introduced the first real-time dense RGB-D reconstruction algorithm. An improvement to this technique is VoxelHashing [30], which proposes a hierarchical hashing approach to store and access voxels. ElasticFusion [42], based on Keller *et al*. [22] is a globally consistent approach that uses fused surfels [32] to represent the scene and does not require a pose graph. Later, algorithms such as InfiniTAM [33] [21] focused on reconstructing large scenes. Recent developments include BundleFusion [8], which performs on-the-fly surface reintegration in real-time. Much of this research has been focused on improving the quality of reconstructions in terms of geometry and texture, as well as allowing the reconstruction of increasingly large spaces using relatively inexpensive hardware. These systems do, however, require that scenes are static. Dynamic elements in the scene can produce artifacts in the reconstruction, as well as high errors in pose estimation, often causing camera tracking failures.

2.2. Robustness to dynamic elements

Due to issues in performing reliable pose estimation, recent research has been increasingly focusing on building systems that are robust to dynamic input. Current SLAM systems approach dynamic movement (non-rigid in particular) either by segmenting out the moving parts or by modelling them explicitly. PoseFusion [45] uses OpenPose to segment out humans by fitting a skeleton. Jaimez *et al*. [20] proposed a joint visual odometry and scene flow (VO-SF) method that segments the scene into rigid clusters and filters out clusters with high registration error. Building on VO-SF, StaticFusion [35] adopts the same segmentation approach and uses ElasticFusion [42] to reconstruct the scene. DynaSLAM [1] uses ORB-SLAM2 [27] along with an approach that combines semantic segmentation with geometry to segment out the dynamic part, showing good improvements in pose estimation. Re-Fusion [31] exploits registration residuals to segment out high-error regions corresponding to scene motion.

Although these approaches show improvements in pose estimation and reconstruction, in many practical applications, dynamic elements are the most important ones in the scene. In particular, humans are widely encountered and are able to change the state of the scene in many ways. As such, in scenarios such as autonomous driving or the deployment of robots in areas with many humans, it would not be appropriate to discard moving elements.

2.3. Non-rigid reconstruction

Building on KinectFusion, DynamicFusion [28] introduced the first real-time non-rigid 3D reconstruction system. VolumeDeform [19] improves on this technique by computing SIFT [25] features to improve frame alignment. Guo *et al*. [17] introduces a pipeline that uses shading information of dynamic scenes to improve the non-rigid registration and temporal correspondences to estimate surface appearance. KillingFusion [36] and SobolevFusion [37] use displacement vectors in voxel space, rather than explicit correspondences. BodyFusion [43] fits a skeleton template for tracking, while HybridFusion [46] uses eight inertial measurement units attached to the reconstructed subject. SurfelWarp [15] employs surfels rather than a TSDF volume and a deformation graph similar to DynamicFusion for computing correspondences. Fusion4D [12] and Dou *et al*. [11] achieve impressive results wielding complex setups that involve four stereo-camera sensors positioned around a moving subject.

While these systems show great potential for applications such as motion capture, they either require complex setups, or do not scale well beyond modelling a single deforming object. As such, they are not fit to be used in applications such as human-robot cooperation, where memory and processing power are restricted, and both static and

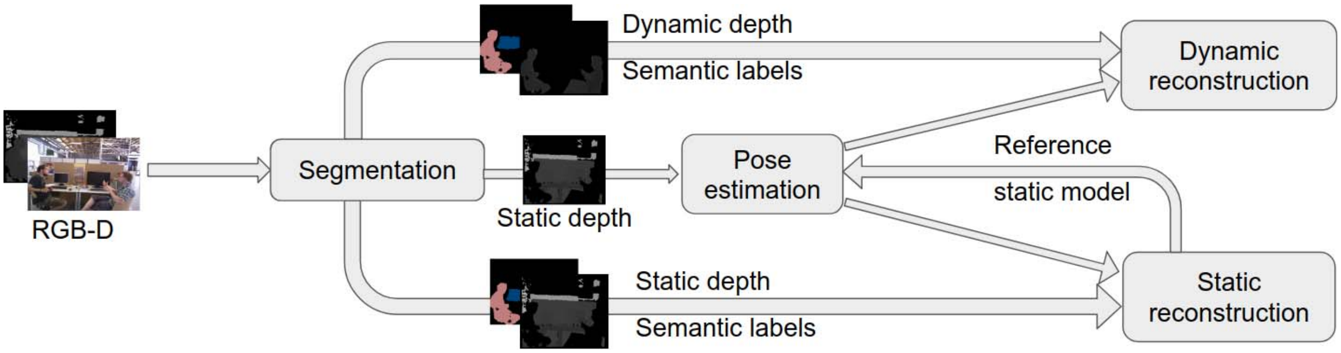

Figure 1. RGB-D input from a sensor such as Microsoft Kinect is divided into a static and a dynamic frame by the *Segmentation module*. The *Pose Estimation* module uses the static frame and a reference frame from the static model to compute the camera position, thus reducing ambiguity between scene dynamics and camera movement. Finally, each of the reconstruction systems receives its processed input, along with the estimated camera pose and semantic labels.

moving elements need to be modelled.

2.4. Semantic scene understanding

Improvements in hardware, and GPU computing in particular, as well as advancements in machine learning, have enabled the development of novel methods for semantic understanding for visual data. Performing online semantic segmentation along with 3D reconstruction has been actively studied in the past few years, and approaches such as SemanticFusion [26], CNN-SLAM [40] and MaskFusion [34] attain excellent results in performing both tasks in real-time. SceneCode [47] recently introduced a code-based learned joint representation of scene semantics and geometry to perform monocular dense semantic reconstruction. ScanComplete [9] uses semantic priors to fill missing information in large-scale scenes.

To the best of our knowledge, the issue of semantic 3D reconstruction with scene labels with both static and non-rigid objects has not been addressed so far.

3. Overview

FullFusion is structured into four loosely-coupled modules, for the following processes:

1. Segmentation
2. Pose estimation
3. Static reconstruction
4. Dynamic reconstruction

We present an overview of our pipeline in Figure 1. The framework receives a registered RGB-D frame pair $\mathcal{F}_t = \{C_t, D_t\}$ at time t defined by a colour image $C_t : \Omega \rightarrow \mathbb{N}^3$ and a depth image $D_t : \Omega \rightarrow \mathbb{N}$ where $\Omega \in \mathbb{N}^2$ is the image plane. The *Segmentation* module produces pairs of frames for the static and dynamic parts of the scene: $\mathcal{F}_t^{static} = \{C_t^{static}, D_t^{static}\}$ and $\mathcal{F}_t^{dynamic} = \{C_t^{dynamic}, D_t^{dynamic}\}$, as well as a label image $L_t : \Omega \rightarrow \mathbb{R}^{|\mathcal{L}|}$ of probabilities with $|\mathcal{L}|$ channels, where $\mathcal{L}$ is the set of labels. The pose estimation module uses $\mathcal{F}_t^{static}$ to compute the pose $T_{lw} \in SE(3)$, representing the 6-DoF transformation from the camera frame to the world frame. Finally, the static and dynamic reconstruction modules receive their respective RGB-D frames, along with the pose and labels.

Our implementation adopts KinectFusion [29] for static reconstruction and DynamicFusion [28] for dynamic reconstruction. Although more advanced systems are currently available, a significant number of publications have been influenced by the ideas presented in KinectFusion and DynamicFusion, and as such, this implementation constitutes a good baseline for future evaluation. We use DeepLabv3+ to perform semantic segmentation.

3.1. API

FullFusion is designed as a generic framework with loosely-connected components. The system is implemented in C++, and only depends on the Eigen library [16], any other dependencies being specific to the implementation of each module. A global configuration file is defined, controlling all hyperparameters for the various modules. Abstract interfaces are defined for Segmentation, Pose estimation, and Reconstruction. The constructor of each interface receives the global configuration and any implementation is expected to acquire its initialisation parameters through the global configuration. All inputs and outputs to functions defined by the interfaces are Eigen matrices (either images or 6-DoF pose in matrix form).

Pseudocode for the abstract interface definitions is provided below:

```
class SegmentationInterface
{
  // Performs segmentation and stores the
  // results to be queried later
```

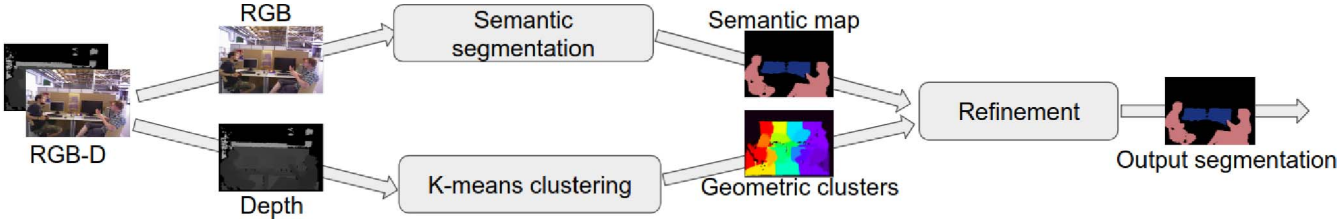

Figure 2. Our implementation of the *Segmentation* module uses DeepLabv3+ trained on the PASCAL-VOC dataset along with a geometric clustering approach to produce geometrically-consistent semantic segmentation

```
  segmentFrame(rgb,depth)
  getStaticFrame() -> static_rgb, static_depth
  getDynamicFrame() -> dynamic_rgb, dynamic_depth
  getSemanticFrame() -> segmentation, probability
}

class ReconstructionInterface
{
    ReconstructionInterface(config)
    processFrame(pose,
                 rgb,
                 depth,
                 segmentation,
                 probability)
    renderModel(pose) -> reference_frame
}

class PoseEstimationInterface
{
    PoseEstimator(config)
    getPose(static_frame, reference_frame) -> pose
}
```

3.2. Segmentation

The segmentation module's job is to provide the other components with appropriate input to increase their performance. Our implementation is based on the observation that since integrating semantic labels in 3D models is desirable for several applications, priors offered by the semantic labels can be used to reason about scene motion. As shown in Figure 2, our implementation combines two approaches to perform semantic segmentation, as well as splitting the input into a static and a dynamic frame. We first use DeepLab v3+ [6][1] trained on the PASCAL-VOC dataset [13] to obtain the label image L_t containing a per-pixel probability distribution over all the recognised classes.

While the semantic segmentation itself is sufficient to segment the scene into static and dynamic parts, the depth input can offer additional geometric priors that can help refine the segmentation. We build on the geometric clustering method introduced by Jaimez *et al*. [20] to segment the scene into K clusters using K-means on the depth image. The refined semantic mask is then obtained by labelling each cluster with the dominant semantic label. We first extract a segmentation map $\mathcal{S} : \Omega \rightarrow \mathbb{N}$ by taking the label with the maximum probability for each pixel: $\mathcal{S} = \{x_i | x_i = argmax_i(y_i), y_i \in \mathcal{L}\}$. Each cluster is th en labelled with the class that occupies the most pixels in the cluster. Finally, neighbouring clusters with the same label are merged to obtain the final segmentation map. The segmentation map is then used as a mask to extract the static and dynamic frame, respectively. Table 1 details the movement labels taken into account when separating the static and dynamic elements.

Static	Non-rigid	Rigid
Background	Bicycle	Aeroplane
Dining table	Bird	Boat
Bottle	Cat	Bus
Chair	Dog	Car
Potted plant	Horse	Motorbike
Sofa	Person	Train
TV/Monitor	Sheep	

Table 1. PASCAL-VOC dataset with movement labels

Since 3D points belonging to different objects may be clustered together, we mitigate this issue by requiring that a class occupies at least 70% of the pixels in a cluster. If this is found not to be the case, we further split the cluster using K-means with $K = 2$. However, we have determined that choosing a reasonably large K, as well as fusing labels from multiple views generally solves the issue.

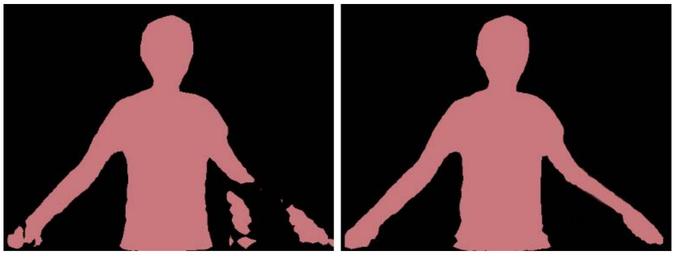

Figure 3. Qualitative comparison of semantic segmentation quality without (left) and with (right) depth clustering

[1] From the Tensorflow GitHub repository

3.3. Pose estimation

The 6-DoF camera pose is estimated using the Iterative Closest Point (ICP) [24] method, as in KinectFusion. ICP aims to minimize the distance between corresponding points in the current depth frame and a reprojection of the current model into the depth frame. Non-rigid motion prevents obtaining good pose predictions using traditional ICP, due to the ambiguity between local changes in non-rigidly moving objects and camera motion, which cannot be explained through a single transformation.

3.4. Reconstruction

Our baseline implementation uses KinectFusion and DynamicFusion for static, and non-rigid reconstruction, respectively. This section provides an outline of the two algorihms and our modifications to include semantic labels, and why neither of the two systems could independently achieve the task FullFusion performs.

KinectFusion and DynamicFusion use voxels to store an implicit volumetric representation as Truncated Signed Distance Function (TSDF) [7], which encodes the distance from the voxel to the closest surface, and is updated at every frame. In order to fuse labels into the scene, we have extended the voxel data structure to store a discrete probability distribution over the set of semantic classes, initialised to a uniform distribution. At any given time, the predicted label of a voxel is simply the one corresponding to the class with the highest probability. Since DynamicFusion is designed to reconstruct individual deforming objects, we store a single semantic label per dynamic model, saving memory and computation time.

We use a running average to update both the TSDF values and the label vector to fuse geometry and semantic labels. As KinectFusion assumes a static environment, any dynamic elements will affect the pose estimation step, as well as data association, which is achieved by rendering the model into the current frame. To solve this issue, DynamicFusion represents the scene using a canonical, rigid model and a coarse warp field. A mesh with vertices $\mathcal{S} \in \mathbb{R}^3$ is extracted using marching cubes from the canonical model at every frame from the TSDF volume. Given the extracted surface and the live depth frame with vertices $\mathcal{T} \in \mathbb{R}^3$, the objective is to transform the canonical model into the live frame $\mathcal{W}(\mathcal{S}) = \mathcal{T}$. This enables alignment between the reconstruction and the current input, which is necessary for data association and fusion.

To estimate the warp field, the following energy function is minimized using a Gauss-Newton non-linear optimization process:

$$E(\mathcal{W}_t, \mathcal{S}, \mathcal{T}) = E_{Data}(\mathcal{W}_{t-1}, \mathcal{S}, T) + \lambda E_{Reg}(\mathcal{W}_{t-1}, \mathcal{S}) \tag{2}$$

The energy function is described by two terms: the data term $E_{Data}(\mathcal{W}_{t-1}, \mathcal{S}, \mathcal{T})$ is a non-rigid ICP function (N-ICP), measuring the difference between the model and the live frame. As non-rigid registration in $\mathbb{R}^3$ is an inherently ill-posed problem [14], an infinite number of solutions can be found, with no guarantee of consistency between frames. To address this issue, an As-Rigid-As-Possible (ARAP) [38] regularisation term $E_{Reg}(\mathcal{W}_{t-1}, \mathcal{S})$ was introduced. E_{Reg} operates as a graph over the set of deformation nodes, with all nodes "pulling" together to promote solutions that deviate the least from the model at $t-1$, and thus ensures smooth deformations. Additionally, it enables the prediction of movement in occluded regions, as there is no E_{Data} associated. λ is a hyperparameter that controls the rigidity of the warp field.

The regularisation graph is responsible for one of the main limitations in DynamicFusion: while E_{Data} is bounded in complexity by the frame size, and is more or less constant between frames, E_{Reg} acts globally, and thus the complexity grows exponentially with the scene size. The vast majority of scenes present far more static objects than non-rigid ones. Even when this is not the case, individual objects deform differently, for unrelated reasons, and have different rigidity properties. As such, global regularisation is not only unnecessary, but a purely distance-based regularisation term such as the one used in DynamicFusion is adverse to the reconstruction quality. As an example, one can imagine a person running their hand over the top of a table. A global regularisation term would cause the surface of the table to "bend" towards the person's hand, due to the deformation nodes pulling together. In such cases, more computation than necessary is performed, however the final result is affected negatively.

4. Experiments

We evaluate our implementation using the SLAMBench framework [3][4]. All experiments were performed on a machine with an Intel Core i7-6700HQ CPU with 16GB of memory, and an NVidia GeForce GTX 960M with 4GB VRAM, running Ubuntu 18.04. Unless otherwise noted, all software was compiled using GCCv6.5.0 and CUDA 9.1. Both KinectFusion and DynamicFusion 1cm voxel sizes and 256^3 volumes. For DynamicFusion, the decimation density used is 25mm, and we use the same hyperparameters recommended in the publication [28]. As no public implementation of either KinectFusion or DynamicFusion is provided by the authors, the evaluation was performed on our own implementation.

4.1. Trajectory accuracy

One of the important assertions of our work is that using only the static component of the scene to estimate the camera pose increases the accuracy. As discussed in Section 2.2, the literature supports this claim.

Setting	Sequence	VO-SF	ElasticFusion	StaticFusion	KinectFusion	FullFusion
Static	fr1/xyz	2.1	**1.9**	2.3	3.0	2.2
	fr1/desk	3.7	**2.9**	3.0	8.2	5.1
	fr1/desk2	5.4	7.2	**5.0**	6.6	7.9
	fr1/plant	6.1	**5.0**	10.4	8.2	12.2
Slightly dynamic	fr3/sit_static	2.4	**0.9**	1.1	2.1	2.1
	fr3/sit_xyz	5.7	**1.6**	2.8	4.4	3.6
	fr3/sit_halfsphere	7.5	17.2	**3.0**	5.7	5.3
Highly dynamic	fr3_walk_static	10.1	26.0	**1.3**	13.8	3.6
	fr3_walk_xyz	27.7	24.0	12.1	tracking lost	**6.0**
	fr3_walk_halfsphere	33.5	20.5	20.7	tracking lost	**7.1**
	fr3_walk_halfsphere*	24.8	16.3	**5.0**	tracking lost	6.3

(a) Comparison of Relative Pose Error (translational RPE-RMSE)

Setting	Sequence	VO-SF	ElasticFusion	StaticFusion	KinectFusion	FullFusion
Static	fr1/xyz	5.1	**1.2**	1.4	1.7	1.3
	fr1/desk	5.6	**2.1**	2.3	3.6	3.7
	fr1/desk2	17.4	5.7	**5.2**	6.0	7.1
	fr1/plant	7.8	**5.3**	11.3	9.2	9.1
Slightly dynamic	fr3/sit_static	2.9	**0.8**	1.3	1.7	1.4
	fr3/sit_xyz	11.1	**2.2**	4.0	3.7	4.3
	fr3/sit_halfsphere	18.0	42.8	4.0	39.6	**3.4**
Highly dynamic	fr3_walk_static	32.7	29.3	**1.4**	79.4	**1.4**
	fr3_walk_xyz	87.4	90.6	12.7	tracking lost	**4.1**
	fr3_walk_halfsphere	73.9	63.8	39.1	tracking lost	**2.9**
	fr3_walk_halfsphere*	48.2	48.6	6.3	tracking lost	**2.7**

(b) Comparison of Absolute Trajectory Error (ATE)

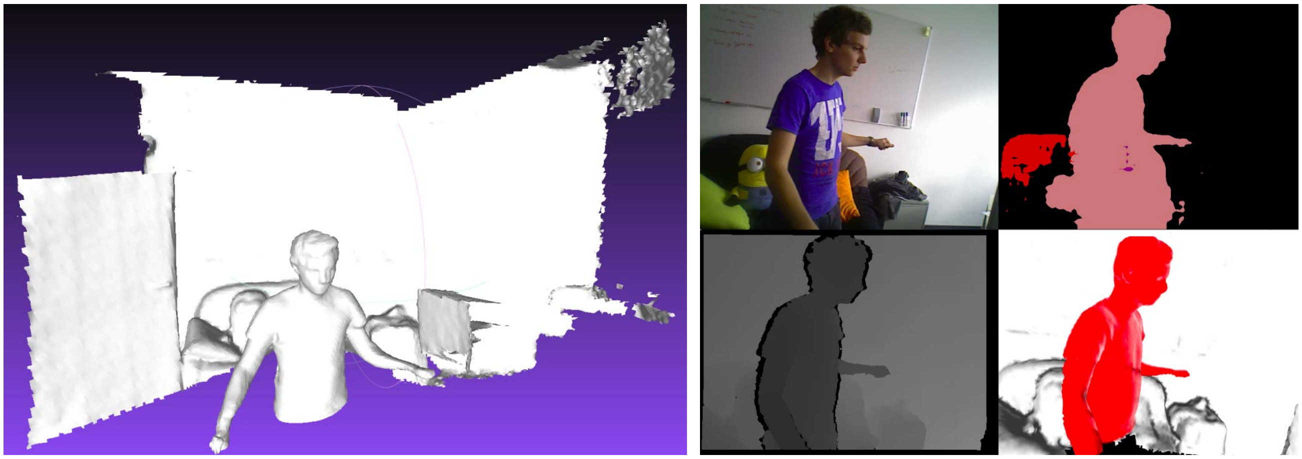

Figure 4. Left: final 3D reconstruction on the upperbody sequence from the VolumeDeform dataset.
Right: in clockwise order, from top-right: RGB frame, Semantic segmentation (without geometric clustering), Depth Frame, Reconstruction (dynamic model highlighted in red)

We evaluate the accuracy of the trajectory estimation using the TUM RGB-D [39] dataset, which provides RGB-D input captured with a Microsoft Kinect device, as well as ground-truth trajectory measurements with static, low dynamic, and highly dynamic sequences. We use the relative pose error (RPE) and absolute trajectory error (ATE) metrics as implemented in SLAMBench. The ATE be computed by directly measuring the absolute distances between the estimated trajectory and the ground-truth trajectory, while the RPE measures the error at each individual pose. For RPE, we report the root-mean square error for the translational component. As seen in Table 3.3, significant improvement can be seen in pose estimation when using only the static part of the scene.

We compare our trajectory accuracy against the following algorithms:

1. VO-SF, a system for visual odometry and scene flow developed by Jaimez *et al*. [20] is a method that computes camera position and is robust to dynamic scenes.
2. StaticFusion [35] builds on VO-SF, additionally performing scene reconstruction using ElasticFusion.
3. ElasticFusion, a surfel-based state-of-the-art method for reconstructing static scenes.
4. KinectFusion, the method we use for static reconstruction and the first real-time RGB-D reconstruction system.

It is worth noting that for VO-SF, ElasticFusion and StaticFusion, we report the results described in the StaticFusion paper [35], and we did not perform the experiments independently. We use SLAMBench to evaluate the results of KinectFusion and FullFusion. In the case of static or slightly dynamic scenes, ElasticFusion tends to perform better than the other systems, as it is designed for high quality static reconstruction. Given the modularity of FullFusion, replacing KinectFusion's ICP with alternative formulations, such as the one employed by ElasticFusion for pose estimation would be straightforward. While in some cases, FullFusion performs worse than the other VO-SF, ElasticFusion, and StaticFusion, we attribute this to our pose estimation module implementation, which is a simple ICP algorithm, whereas ElasticFusion uses a joint ICP and photometric error. We note, however, that almost all cases which contain movement, FullFusion performs at least as well as KinectFusion, thus showing that the segmentation improves pose estimation quality. While FullFusion and KinectFusion use the same pose estimation technique, there are noticeable differences in the results on static scenes. These differences arise due to the sequences containing persons which are being segmented out of the frame when computing the pose. An important highlight of the results is that in the more challenging cases, KinectFusion eventually suffers from tracking failure, while FullFusion performs better than any of the other methods. We believe that the reason FullFusion has better performance then VO-SF and StaticFusion is that relying on semantic priors circumvents the need for an initialization period. For the sake of completeness, we show results on *fr3_walk_halfsphere* with the first 5 seconds skipped (marked as *fr3_walk_halfsphere**), as done in StaticFusion.

4.2. Qualitative evaluation

Figure 4 shows the output of our reconstruction on the *upperbody* sequence from the VolumeDeform [19] dataset. We note that the lack of datasets containing ground-truth reconstructions for both dynamic and static objects is a challenge yet to be addressed.

4.3. MixedFusion: discussion

As MixedFusion is by far the most similar system to ours, it would be ideal to include an evaluation against FullFusion. Unfortunately, due to the lack of a public implementation, or any results on public datasets, we cannot offer any direct comparison. While we fully acknowledge that a quantitative comparison would be superior, we believe that it is necessary to draw a comparison based on our understanding of their work and the provided video[2].

A summary of the differences between FullFusion and MixedFusion is necessary. MixedFusion is a reconstruction system which uses a formulation that jointly computes the camera pose and segments the scene. On the other hand, FullFusion is a framework which ensures the interaction of loosely-coupled subsystems working together to achieve a more complex goal, while also showing improvements in some of the tasks performed by the subsystems. Secondly, rather than using a purely geometric approach to segment the scene, we leverage both geometry and semantics. Finally, we not only use semantics for segmentation, but also fuse the labels into the reconstruction.

An assumption of S-ICP, used for segmentation in MixedFusion, is that a dynamic objects will occupy a small portion of the scene. As shown above, FullFusion shows good performance on cases such as *fr3_walk_halfsphere*, a sequence where there is significant movement from the very beginning, whereas algorithms segmenting the scene solely based on geometry such as StaticFusion require an initialization period, and thus perform poorly. Considering the inherent ambiguities in geometry, we believe any geometry-based method, including that of MixedFusion would produce similar behaviour.

Further to this, the authors of MixedFusion note that one of the limitations of their system is that since their segmentation pipeline is based on geometry connectivity, dynamic objects cannot be accurately segmented if they are connected with the static scene, and suggest that semantic information can help solve the issue. Our results indicate that a joint semantic and geometric segmentation module achieve good performance.

One of the downsides of using semantic priors to predict movement is that exhaustive labelling of all moving classes may not be possible. Moreover, depending on the context, objects might exhibit different behaviour (*e.g*. indoor plants

[2] Available on IEEE Xplore

may be generally static, but outdoor plants will likely be affected by wind). As such, MixedFusion generalises better to any scene movement, as it is not restricted to a finite number of recognised classes. FullFusion could benefit from implementing more robust pose estimation methods such as S-ICP into the segmentation module, or using generic moving object segmentation [41][10] may increase robustness.

5. Limitations and future work

Our system allows overcoming issues presented by 3D reconstruction systems such as KinectFusion and DynamicFusion. Nonetheless, many of the shortcomings of the individual systems affect FullFusion: relocalisation of dynamic models that exit the scene continues to be a challenge, and we do not address the non-functional aspects of 3D scenes, such as producing textured models.

Our current implementation uses a per-voxel probability distribution over the set of labels. This does not scale well with the number of classes, and thus a method similar to SemanticFusion [26] which stores a single label and its probability may be preferable. Additional improvements to the 3D segmentation may include instance segmentation, for example using Mask R-CNN [18] or panoptic segmentation [23], as in the current implementation, two objects of the same class located at a similar distance from the camera could be treated as a single object.

Finally, we see our work in the context of emerging technologies for benchmarking (e.g SLAMBench [4] and hyperparameter tuning (e.g HyperMapper [2]), which provide opportunities to tailor SLAM systems to suit specific applications. In this respect, FullFusion lays groundwork towards finding the right combination of systems for dynamic environments. Considering that KinectFusion was the first real-time RGB-D reconstruction system, and DynamicFusion was the first real-time non-rigid reconstruction system, and both informed a significant number of works, the authors believe that the current implementation constitutes a good baseline for evaluating future solutions.

6. Conclusion

This paper presents FullFusion, a framework for semantic 3D reconstruction of dynamic scenes using RGB-D sensors. We demonstrate our system using KinectFusion [29] for static reconstruction, DynamicFusion [28] for dynamic reconstruction, dense ICP [24] for pose estimation, and a segmentation module based on DeepLabv3+ [6] for semantic segmentation.

In addition to performing a more complex overall task than each of its individual components, FullFusion achieves better performance than its individual modules. Pose estimation is improved by using only the static part of the scene, and the system is able to reconstruct both the static and dynamic scene parts.

Acknowledgments

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

References

[1] B. Bescos, J. M. Fácil, J. Civera, and J. Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.

[2] B. Bodin, L. Nardi, M. Z. Zia, H. Wagstaff, G. S. Shenoy, M. Emani, J. Mawer, C. Kotselidis, A. Nisbet, M. Lujan, et al. Integrating algorithmic parameters into benchmarking and design space exploration in 3d scene understanding. In *2016 International Conference on Parallel Architecture and Compilation Techniques (PACT)*, pages 57–69. IEEE, 2016.

[3] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Luján, S. Furber, A. J. Davison, et al. Slambench2: Multi-objective head-to-head benchmarking for visual slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[4] M. Bujanca, P. Gafton, S. Saeedi, A. Nisbet, B. Bodin, M. O'Boyle, A. J. Davison, P. Kelly, G. Riley, B. Lennox, et al. Slambench 3.0: Systematic automated reproducible evaluation of slam systems for robot vision challenges and scene understanding.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[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, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(3):24, 2017.

[9] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2018.

[10] A. Dave, P. Tokmakov, and D. Ramanan. Towards segmenting everything that moves. *CoRR*, abs/1902.03715, 2019.

[11] 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.

[12] 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.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[14] K. Fujiwara, K. Nishino, J. Takamatsu, B. Zheng, and K. Ikeuchi. Locally rigid globally non-rigid surface registration. In *2011 International Conference on Computer Vision*, pages 1527–1534. IEEE, 2011.

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

[16] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

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

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[19] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*, pages 362–379. Springer, 2016.

[20] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers. Fast odometry and scene flow from rgb-d cameras based on geometric clustering. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3992–3999. IEEE, 2017.

[21] O. Kähler, V. A. Prisacariu, and D. W. Murray. Real-time large-scale dense 3d reconstruction with loop closure. pages 500–516, 2016.

[22] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013.

[23] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[24] K.-L. Low. Linear least-squares optimization for point-to-plane icp surface registration.

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

[26] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017.

[27] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[28] 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.

[29] 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.

[30] 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.

[31] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. Refusion: 3d reconstruction in dynamic environments for RGB-D cameras exploiting residuals. *CoRR*, abs/1905.02082, 2019.

[32] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM Press/Addison-Wesley Publishing Co., 2000.

[33] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. Torr, and D. W. Murray. Infinitam v3: A framework for large-scale 3d reconstruction with loop closure. *arXiv preprint arXiv:1708.00783*, 2017.

[34] M. Runz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20. IEEE, 2018.

[35] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers. Staticfusion: background reconstruction for dense rgb-d slam in dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

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

[37] 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.

[38] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 109–116. Eurographics Association, 2007.

[39] 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.

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

[41] P. Tokmakov, C. Schmid, and K. Alahari. Learning to segment moving objects. *International Journal of Computer Vision*, 127(3):282–301, 2019.

[42] 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.

[43] 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.

[44] H. Zhang and F. Xu. Mixedfusion: Real-time reconstruction of an indoor scene with dynamic objects. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3137–3146, Dec 2018.

[45] T. Zhang and Y. Nakamura. Posefusion: Dense rgb-d slam in dynamic human environments. In *2018 International Symposium on Experimental Robotics*, 2018.

[46] 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.

[47] S. Zhi, M. Bloesch, S. Leutenegger, and A. J. Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11776–11785, 2019.

[48] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer Graphics Forum*, volume 37, pages 625–652. Wiley Online Library, 2018.