We present MaskFusion, a real-time, object-aware, semantic and dynamic RGB-D SLAM system that goes beyond traditional systems which output a purely geometric map of a static scene. MaskFusion recognizes, segments and assigns semantic class labels to different objects in the scene, while tracking and reconstructing them even when they move independently from the camera. As an RGB-D camera scans a cluttered scene, image-based instance-level semantic segmentation creates semantic object masks that enable real-time object recognition and the creation of an object-level representation for the world map. Unlike previous recognition-based SLAM systems, MaskFusion does not require known models of the objects it can recognize, and can deal with multiple independent motions. MaskFusion takes full advantage of using instance-level semantic segmentation to enable semantic labels to be fused into an object-aware map, unlike recent semantics enabled SLAM systems that perform voxel-level semantic segmentation. We show augmented-reality applications that demonstrate the unique features of the map output by MaskFusion: instance-aware, semantic and dynamic. Code will be made available.

Index Terms: Visual SLAM—SLAM—Visualization—Tracking; Mapping—Fusion—RGBD—Multi-object Recognition—Context—Semantic—Detection Real-time—Augmented-Reality—Robotics

1 INTRODUCTION

Perceiving the world around us in 3D from image sequences acquired from a moving camera is a fundamental task in fields such as computer vision, robotics, human-computer and human-robot interaction. Visual SLAM (Simultaneous Localisation and Mapping) systems have focused, for decades now, on jointly solving the tasks of tracking the position of a camera as it explores unknown locations and creating a 3D map of the environment. Their real-time capability has turned SLAM methods into the cornerstone of ambitious applications such as autonomous driving, robot navigation and also augmented/virtual reality. Research in Visual SLAM has progressed at a fast pace, moving from early works that reconstructed sparse maps with just a few tens or hundreds of features using filtering techniques [11], to parallel tracking and mapping approaches that could take advantage of computationally expensive batch optimisation techniques for the mapping thread to produce accurate maps with thousands of landmarks [25][30], to contemporary methods that allow instead to reconstruct completely dense maps of the environment [33][34][50]. The impact on augmented reality of this progression towards dense and robust real-time mapping has been...
immense with many SLAM enabled augmented reality applications making their way into consumer products and mobile phone apps. Despite these advances, there are still two areas in which SLAM methods and their application to augmented reality are still very much in their infancy.

Most SLAM methods rely on the assumption that the environment is mostly static and moving objects are, at best, detected as outliers and ignored. Although some first steps have been taken towards non-rigid and dynamic scene reconstruction, with exciting results in reconstruction of a single non-rigid object [12,20,32,53], designing an accurate and robust SLAM system that can deal with arbitrary dynamic and non-rigid scenes remains an open challenge.

The output provided by the majority of SLAM systems is a purely geometric map of the environment. The addition of semantic information is relatively recent [6,8,28,40,44] and is mostly limited to the recognition of a small number of known object instances for which a 3D model is available in advance [28,40,44] or to classify each 3D map point into a fixed set of semantic categories without differentiating object instances [28,44].

Contribution: the novelty of our approach is to make advances towards addressing both of these limitations within the same system. MaskFusion is a real-time capable SLAM system that can represent scenes at the level of objects. It can recognise, detect, track and reconstruct multiple moving rigid objects while precisely segmenting each instance and assigning it a semantic label. We take advantage of combining the outputs of: (i) Mask-RCNN [15], a powerful image-based instance level segmentation algorithm that can predict object category labels for 80 object classes, and (ii) a geometry-based segmentation algorithm, that generates an object edge map from depth and surface normal cues; to increase the accuracy of the object boundaries in the object masks.

Our dynamic SLAM framework takes these accurate object masks as input to track and fuse multiple moving objects (as well as the static background) while propagating the semantic image labels into temporally-consistent 3D map labels. The main advantage of using instance-aware semantic segmentation over standard pixel-level semantic segmentation (such as most previous semantic SLAM systems [6,8,28,40,44,46]) is that it provides accurate object masks and the ability to segment different object instances that belong to the same object category instead of treating them as a single blob.

The additional advantage of MaskFusion over previous semantic SLAM systems [6,8,28,40,44,46] is that it does not require the scene to be static and so can detect, track and map multiple independently moving objects. Maintaining an internal 3D representation of moving objects (instead of treating them as outliers) substantially improves the overall SLAM system by providing a richer map that includes not just the background but also the detailed geometry of the moving objects, and by improving object and camera pose prediction and estimation.

On the other hand, the advantage of MaskFusion over previous dynamic SLAM systems [3,19] is that it enhances the dynamic map with semantic information from a large number of object classes in real time. Not only can it detect individual objects (thanks to the use of Mask-RCNN [15]) and assign semantic labels to their corresponding 3D map points, but it can also accurately segment each individual object instance. Table 1 summarises our contributions in the context of other real-time semantic SLAM and dynamic SLAM systems.

The result is a versatile system that can represent a dynamic scene at the level of objects and their semantic labels, which has numerous applications in areas such as robotics and augmented reality. We demonstrate how the labels of objects can be used for different purposes. For instance, we show that often, being able to detect and segment people allows us to be aware of their presence, ignore those pixels and focus instead on the objects that they are manipulating. We show how this can be useful in object manipulation tasks, as it can improve object tracking even when objects are moved and occluded by a human hand.

2 RELATED WORK

The field of Visual SLAM has a long history of offering solutions to the problem of jointly tracking the pose of a moving camera (see [4] for a recent survey) while reconstructing a map of the environment. The advent of inexpensive, consumer-grade RGB-D cameras – such as the Microsoft Kinect – stimulated further research, and enabled the leap to dense real-time methods [23,24,35].

Dense RGB-D SLAM: Resulting methods are capable of accurately mapping indoor environments and gained popularity in augmented reality and robotics. KinectFusion [33] proved that a truncated signed distance function (TSDF) based map representation can achieve fast and robust mapping and tracking in small environments. Subsequent work [38,51] showed that the same principles are applicable to large scale environments by choosing appropriate data structures.

Surface elements (surfels) have a long history in computer graphics [35] and have found many applications in computer vision [5,9]. More recently, surfel-based map representations were also introduced [18,23] to the domain of RGBD-SLAM. A map of surfels is similar to a point cloud with the difference that each element encodes local surface properties – typically a radius and normal – in addition to its location. In contrast to a TSDF-based map, surfel clouds are naturally memory efficient and avoid the overhead due to switching representations between mapping and tracking that is typical of TSDF-based fusion methods. Whelan et al. [50] presented a surfel-based RGB-D SLAM system for large environments with local and global loop-closure.

Scene segmentation: The computer graphics [7] and vision [9,17,22,25,46] communities have devoted substantial effort to object and scene segmentation. Segmented data can broaden the functionality of visual tracking and mapping systems, for instance, by enabling robots to detect objects. Some methods have proposed to segment RGBD data based on geometric properties of surface normals [9,13,22,45], mainly by assuming that objects are convex. While the clear strength of geometry-based segmentation systems is that they produce accurate object boundaries, their weakness is that they typically result in over-segmentations and they do not convey any semantic information.

Semantic scene segmentation: Another line of work [2,26,52] aims at segmenting 3D scenes semantically, using Markov Random Fields (MRFs). These methods require labelled 3D data, however, which in contrast to labelled 2D image data is not readily available. This is exemplified by the fact that all three works involved manual annotation of training data. Datasets containing isolated RGBD frames, such as NYUv2 [31], are not applicable here and it requires significant effort to build consistent reconstructed datasets for segmentation, as recently shown by Dai et al. [10].

Semantic SLAM: Motivated by the success of convolutional neural networks [15,46,47], Tateno et al. [44] and McCormac et al. [28] integrate deep neural networks in real-time SLAM systems. As inference is solely based on 2D information, the need for 3D annotated data is circumvented. The resulting systems offer strategies to fuse labelled image data into segmented 3D maps. Earlier work by Herman et al. [19] implements a similar scheme, using a randomised decision forest classifier. However, since the systems are not considering object instances, tracking multiple models independently is unattainable.

Dynamic SLAM: There are two main scenarios in dynamic SLAM:
non-rigid surface reconstruction and multibody formulations for independently moving rigid objects. In the first case, a deformable world is assumed \cite{12,32,53} and as-rigid-as-possible registration is performed, while in the second, rigid object instances are identified \cite{40,46} and tracked sparsely \cite{48,54} or densely \cite{39}. Both categories use template- or descriptor-based formulations \cite{40,46,53}, which require pre-observing objects of interest, and template-free methods. In the case when the dynamic parts of the scene are not of interest, it is valuable to recognise them as outliers to avoid errors in the optimisation back-end. Methods for the explicit detection of dynamic regions for static fusion were proposed by Jaimez et al. \cite{21} and Scona et al. \cite{41}.

Table 1 provides an overview of related real-time capable methods comparing them under five important properties.

| Method                      | Model-free | Scene Segmentation | Semantics | Multiple moving objects | Non-Rigid |
|-----------------------------|------------|--------------------|-----------|-------------------------|-----------|
| Static-Fusion \cite{41}     | ✓          | ✓                  | ✓         |                         | ✓         |
| 2.5D is not enough \cite{46} | ✓          | ✓                  | ✓         |                         | ✓         |
| Slam++ \cite{40}            | ✓          | ✓                  | ✓         |                         | ✓         |
| CNN-SLAM \cite{44}          | ✓          | ✓                  | ✓         |                         | ✓         |
| Semantic-Fusion \cite{28}   | ✓          | ✓                  | ✓         |                         | ✓         |
| Non-Rigid RGBD \cite{53}    | ✓          |                    |           |                         |           |
| Dynamic-Fusion \cite{32}    | ✓          |                    |           |                         |           |
| Fusion4D \cite{12}          | ✓          |                    |           |                         |           |
| Co-Fusion \cite{39}         | ✓          | ✓                  | ✓         | ✓                       | ✓         |
| Mask-Fusion                 | ✓          | ✓                  | ✓         | ✓                       | ✓         |

Table 1: Comparison of the properties of MaskFusion with respect to other real-time SLAM systems. In contrast to previous semantic SLAM systems \cite{28,40,44,46}, MaskFusion is both dynamic (it reconstructs objects even when their motion is different from the camera) and segments object instances. Unlike dense non-rigid reconstruction systems \cite{12,32,53}, it can reconstruct the entire scene and adds semantic labels to different objects. Note that while Co-Fusion \cite{39} could use semantic cues to segment the scene, in that case the system was not real-time – only the non-semantic version of Co-Fusion was real-time capable.

Figure 2: High-level overview of the SLAM back-end and masking network, and their interaction.

### 3 System Overview

MaskFusion enables real-time dense dynamic RGBD SLAM at the level of objects. In essence, MaskFusion is a multi-model SLAM system that maintains a 3D representation for each object that it recognises in the scene (in addition to the background model). Each model is tracked and fused independently. Figure 2 illustrates its frame-to-frame operation. Each time a new frame is acquired by the camera, the following steps are performed:

**Tracking:** The 3D geometry of each object is represented as a set of surfels. The six degree of freedom pose of each model is tracked by minimizing an energy that combines a geometric iterative closest point (ICP) error with a photometric cost based on brightness constancy between corresponding points in the current frame and the stored 3D model, aligned with the pose in the previous frame. In order to lower computational demand and increase robustness, only non-static objects are tracked separately. Two different strategies were tested to decide whether an object is static or not: one based on motion inconsistency, similar to \cite{39}, and another that treats objects which are being touched by a person as dynamic.

**Segmentation:** MaskFusion combines two types of cues for segmentation: semantic and geometric cues. Mask-RCNN \cite{15} is used to provide object masks with semantic labels. While this algorithm is impressive and provides good object masks, it suffers from two drawbacks. First, the algorithm does not run in real time and can only operate at a maximum of 5 Hz. Second, the object boundaries are not perfect – they tend to leak into the
background. To overcome both of these limitations, we run a
geometric segmentation algorithm based on an analysis of depth
 discontinuities and surface normals. In contrast to the semantic
instance segmentation, the geometric segmentation runs in real
time and produces very accurate object boundaries (see Figures 3d)
and (e) for an example visualisation of the geometric edge map and
the geometric components returned by the algorithm). On the negative
side, geometry-based segmentation tends to oversegment objects.
The combination of these two segmentation strategies—geometric
segmentation on a per-frame basis and semantic segmentation as
often as possible—provides the best of both worlds, allowing us
to (1) run an overall system in real-time (geometric segmentation
is used for frames without semantic object masks, while the
combination of both is used for frames with object masks) and (2)
obtain semantic object masks with improved object boundaries,
thanks to the geometric segmentation.

**Fusion:** The geometry of each object is fused over time by using
the object labels to associate surfels with the correct model. Our
fusion follows the same strategy as [23, 50].

The rest of the paper is organised as follows. We first describe
the principles of our dynamic RGBD-SLAM method in Section 4;
we provide further details regarding the integration of the semantic and geometric
segmentation results are provided in Section 5. A quantitative
comparison of the proposed approach is presented in Section 6.

### 4 MULTI-OBJECT SLAM

MaskFusion maintains a set of independent 3D models,
\( \mathcal{M}_m \forall m \in \{0..N\} \), for each of the N objects recognised
in the scene and a further model for the background. We adopt the surfel representation popularised by [23, 50],
where a model \( \mathcal{M}_m \) is represented by a cloud of surfels
\( \mathcal{M}_m \in \{p \in \mathbb{R}^3, n \in \mathbb{N}^3, \mathbf{w} \in \mathbb{R}, \mathbf{r} \in \mathbb{R}, t \in \mathbb{R}^2 \} \) \( \forall s < |\mathcal{M}_m| \),
which are tuples of position, normal, colour, weight, radius and
two timestamps. Additionally, models are associated with a class ID
\( e_m \in \{0..80\} \) and an object-label \( t_m = m \forall m \in \{0..N\} \). Finally, for
each time instance \( t \), an is static indicator \( s_m \in \{0, 1\} \) and a rigid pose
\( R_m \in \mathbb{SO}_3, t_m \in \mathbb{R}^3 \) is stored.

#### 4.1 Tracking

Assuming that a good estimate exists for the pose of model \( \mathcal{M}_m \) at
time \( t \)−1, the pose at time \( t \) is inferred by aligning the current depth-
map \( \mathcal{D}_t \) and intensity-map \( \mathcal{J}_t \) with the projection \( \mathcal{P}_m \) \( \forall m \in \mathcal{M}_m \),
which is generated by rendering its surfels using the OpenGL
pipeline. Here, \( \mathcal{D}_t \) and \( \mathcal{J}_t \) are mappings from image-coordinates
\( \Omega \subset \mathbb{N}^2 \) to depth \( \mathcal{D}_t : \Omega \rightarrow \mathbb{R} \) and grey-scale \( \mathcal{J}_t : \Omega \rightarrow \mathbb{N} \),
respectively. \( \mathcal{J}_t \) is derived by weighting RGB channels as follows: \( r, g, b \rightarrow 0.299r + 0.587g + 0.114b \).

The alignment is performed by minimising a joint geometric and
photometric error function [39, 50],
\[
E_m = \min_{\mathbf{R}, \mathbf{t}, \mathbf{c}} \left( E_{mp} + \lambda E_{rgb} \right),
\]
where \( E_{mp} \) and \( E_{rgb} \) are the geometric and photometric error
terms respectively and \( \mathbf{c} \) is the unknown rigid transformation,
expressed in a minimal 6D Lie algebra representation \( se_3 \), which is
subject to optimisation.

The first term in equation (1) is a sum of projective ICP residuals.
Given a vertex \( \mathbf{v}_j \), which is the back-projection of the \( i \)-th vertex in
\( \mathcal{D}_t \); and \( \mathbf{v}_j \) and \( \mathbf{n}_j \), the corresponding vertex and normal in \( \mathcal{P}_m \),
the geometry expressed in the camera coordinate frame at time \( t \)−1,
\( E_{mp} \) is written as:
\[
E_{mp} = \sum_i \left( (\mathbf{v}_j - \exp(\mathbf{c})\mathbf{v}_i) \cdot \mathbf{n}_i \right)^2
\]

The photometric term, on the other hand, is a sum of photo-
consistency residuals between \( \mathcal{I}_t \) and \( \mathcal{I}_{t-1} \), and reads as follows:
\[
E_{rgb} = \sum_{u \in \Omega} \left( \mathcal{I}_t(u) - \mathcal{I}_{t-1}(\pi(\exp(\mathcal{c}_m)\pi^{-1}(u, \mathcal{D}_t))) \right)^2
\]

Here, \( \pi \) performs a perspective projection \( \mathbb{R}^3 \rightarrow \mathbb{R}^2 \), whereas
\( \pi^{-1} \) back-projects from a depth map with 2D coordinate. To
optimise this non-linear least-squares cost we use a Gauss-Newton
solver with a four level coarse-to-fine pyramid scheme. The CUDA
accelerated implementation of the solver builds on the open source
code releases of [50] and [39].

#### 4.2 Fusion

Given \( R_m \) and \( t_m \), surfels for each model \( \mathcal{M}_m \) are updated by
performing a projective data association with the current RGBD
frame. This step is inspired by [23, 50], but a stencilling based on the
segmentation discussed in Section 5 is used to adhere to object
boundaries. As a result, each newly created surfel is part of exactly
one model. Further, we introduce a confidence penalty for surfels
outside the stencil, which is required due to imperfect segmentations.

### 5 SEGMENTATION

MaskFusion reconstructs and tracks multiple objects simultaneously,
maintaining separate models. As a consequence, new data has to be
associated with the correct model before fusion is performed. Inspired
by Co-Fusion [39], instead of associating data in 3D, segmentation
is carried out in 2D and model-to-segment correspondences are
established. Given these correspondences, new frames are
masked and only subsets of the data are fused with existing models.
Masking is based on the semantic instance segmentation labels provided by a DNN [15], in conjunction with geometric segmenta-
tion, which improves the quality of object boundaries. Our
semantic segmentation pipeline provides masks at 30Hz or more.

The design of the pipeline is based on the following observations:
(i) Current semantic segmentation methods are good at detecting
objects, but tend to provide imperfect object boundaries. (ii) The cur-
rent state-of-the-art approach, Mask-RCNN [15], cannot be executed
at frame rate. (iii) The information contained in RGBD frames en-
ables fast over-segmentation of the image, for instance by assuming
object convexity.

The second observation directly implies that to achieve overall
time performance our system must execute instance level seman-
tic segmentation in a parallel thread concurrently to the tracking and
fusion threads. However, executing two programs at different fre-
cuencies concurrently requires a synchronisation strategy. We buffer
new frames in a queue \( Q \) and refer the SLAM system to the head of
the queue, while the semantic segmentation operates on the back of
the queue, as illustrated in Figure 2a. This way, the execution of the
SLAM pipeline is delayed by the worst-case processing time of the
semantic segmentation. In our experiments we picked a queue length
of 12 frames, which involves a delay of approx. 400ms. Whether
this delay can be neglected or not, depends on the use-case of the
system. Even though a latency exists, the system runs at a frame-rate
of 30fps. Furthermore, a semantic segmentation is not available for
most frames due to the lower execution frequency of the masking
component, yet each frame requires a labelling in order to fuse new
data. This issue is solved by associating regions of mask-less frames
with existing models only, as discussed in Section 5.3.

To compensate for inexact boundaries, as mentioned in observa-
tion 1, we make use of observation 3 and map components from
a geometric over-segmentation to semantic masks. This results in
improved masks, due to higher-quality boundaries of the geometric
segmentation.
5.1 SEMANTIC INSTANCE SEGMENTATION

A variety of recently proposed neural network architectures are tackling the problem of instance-level object segmentation. They outperform traditional methods and are capable of handling a large set of object classes. Of these methods, Mask-RCNN [15] is especially compelling, as it provides superior segmentation quality at a relatively high frame-rate of 5Hz. The semantic segmentation pipeline of MaskFusion is based on Mask-RCNN [15], which maps RGB frames to a set of object masks \( \mathcal{Z}_{ts} : \Omega \to \{0,1\} \), bounding boxes \( b_{tn} \in \mathbb{N}^d \) and class IDs \( c_{tn} \in \{0..80\} \), for all \( n \in \{1..N_t^t\} \) of the \( N_t \) instances detected in the frame at time \( t \).

Mask-RCNN achieves this by extending the Faster-RCNN [37] architecture. Faster-RCNN is a two-stage approach that proposes regions of interest first and then predicts an object class and bounding box per region and in parallel. He et al. added a third branch to the second stage, which generates masks independently of class IDs and bounding boxes. Both stages rely on a feature map, which is extracted by a ResNet [10]-based backbone network, and apply convolutional layers for inference.

Figure 3 visualises the output of Mask-RCNN. Note that instances of the same class are highlighted with different colours, and also that masks are not perfectly aligned with object boundaries.

5.2 GEOMETRIC SEGMENTATION

Assuming that objects – especially man-made objects – are largely convex, it is possible to build fast segmentation methods that place edges in concave areas and depth discontinuities. In practice, such methods tend to oversegment data, due to the simplified premise. Moosmann et al. [29] successfully segment 3D laser data based on this assumption. The same principle is also used by other authors to segment objects in RGBD frames [13][22][42][45][47].

Our geometric segmentation method follows this approach and, similarly to [45], generates an edginess-map based on a depth discontinuity term \( \phi_d \) and concavity term \( \phi_c \). Specifically, a pixel is defined as an edge pixel if \( \phi_d + \lambda \phi_c > \tau \), where \( \tau \) is a threshold and \( \lambda \) a relative weight. Given a local neighbourhood \( \mathcal{N} \), \( \phi_d \) and \( \phi_c \) are computed as follows:

\[
\phi_d = \max_{i \in \mathcal{N}} (\mathbf{v}_i - \mathbf{v}) \cdot \mathbf{n} \\
\phi_c = \max_{i \in \mathcal{N}} \begin{cases} 
0 & \text{if } (\mathbf{v}_i - \mathbf{v}) \cdot \mathbf{n} < 0 \\
1 - (\mathbf{n} \cdot \mathbf{n}) & \text{else} 
\end{cases}
\]

Here, \( \mathbf{v} \) and \( \mathbf{v}_i \) indicate vertex positions, while \( \mathbf{n} \) and \( \mathbf{n}_i \) represent normals, obtained by back-projecting \( \mathcal{S} \). Since \( \phi_d + \lambda \phi_c \) depends on a local neighbourhood only, the edginess of a pixel can be evaluated quickly on a GPU. Figure 4d shows the edge map for a frame that was captured with an Asus Xtion RGBD-camera. Edge maps are converted to a geometrical labelling \( \mathcal{Z}_{tg} : \Omega \to \{0..N_t^g\} \), where \( N_t^g \) is the number of extracted components excluding the background, by running an out-of-the-box connected components algorithm, as illustrated in Figure 5.

5.3 MERGED SEGMENTATION

For each frame that is processed by the SLAM system, the pipeline illustrated in Figure 5 is executed. While the geometric segmentation, shown on the left-hand-side, is performed for all frames, geometric labels are mapped to semantic masks only if these are available. In the absence of semantic masks, geometric labels are associated with existing models directly and the following steps are skipped:

1. We are using the Matterport implementation of Mask-RCNN.
To objectively compare MaskFusion with other methods, we evaluate its performance on an established RGBD benchmark dataset.\footnote{We use the value 255, as we represent labels as unsigned bytes and assume a number of models less than that.}
Figure 7: Detecting persons allows MaskFusion to ignore them. In this challenging sequence (fr3_walking_halfsphere), the reconstruction only contains static parts.

Figure 6: Comparison of camera and object trajectories with ground-truth. The AT-RMSEs amount to 2.2cm and 8.9cm for the teddy bear and camera trajectory, respectively. Because the bear occupies a significant proportion of the field of view, tracking it independently affects the quality of the camera pose estimation. Treating the object as part of the background would reduce the camera AT-RMSE to 7.2cm.

this is that other methods label points as dynamic / outlier that would still be beneficial for tracking, and hence show inferior performance.

Making use of context information proves to be especially useful in highly dynamic scenes, or when the beginning of a scene is difficult. These cases can be hard to tackle by energy minimisation, whereas semantic segmentation results are shown to be robust.

Further, we reconstruct and track the teddy bear in sequence β_long_office independently from the background motion. This way it is possible to compare the estimated object trajectory with the ground-truth camera trajectory, as highlighted in Figure 6. The trajectory of the bear is only available for a subsection of the sequence as it is out-of-view otherwise.

### 6.1.2 Reconstruction

We conducted a quantitative evaluation of the quality of the 3D reconstruction achieved by MaskFusion using objects from the YCB Object and Model Set [4], a benchmark designed to facilitate progress in robotic manipulation applications. The YCB set provides physical daily life objects of different categories, which are supplied to research teams, as well as a database with mesh models and high-resolution RGB-D scans of the objects. We selected a ground truth model from the dataset (a bleach bottle), and acquired a dynamic sequence to quantitatively evaluate the errors in the 3D reconstruction. Figure 8 shows an image of the object, the ground truth 3D model, our reconstruction and a heatmap showing the 3D error per surfel. The average 3D error for the bleach bottle was 7.0mm with a standard deviation of 5.8mm (where the GT bottle is 250mm tall and 100mm across).

### 6.1.3 Segmentation

To assess the quality of the segmentation quantitatively we acquired a 600 frame long sequence and provided ground truth 2D annotations for the masks of one of the objects (teddy). Figure 8 shows the intersection over union (IoU) graphs for three different runs. The IoU of the per-frame segmentation masks obtained with MaskRCNN only and MaskRCNN combined with the geometric segmentation are shown in red and blue respectively. The blue curve shows the IoU obtained using our full method, where the object masks are obtained by reprojecting the reconstructed 3D model. This graph shows how combining semantic and geometric cues results in more accurate segmentations, but even better results are achieved when maintaining temporally consistent 3D models over the sequence through tracking and fusion.

### 6.2 Qualitative results

We tested MaskFusion on a variety of dynamic sequences, which show that it presents an effective toolbox for different use cases.

#### 6.2.1 Grasping

A common but challenging task in robotics is to grasp objects. Aside from requiring sophisticated actuators, a robot needs to identify grasping points on the correct object. MaskFusion is well suited to provide the relevant data, as it detects and reconstructs objects densely. Further, and in contrast to most other systems, it continues the tracking during interaction. If the appearance of the actuator is known in advance or if a person interacts with objects, the neural network can be trained to exclude these parts from the reconstruction.

Figure 12 shows a timeline of frames that illustrate a grasping performance. In this example, the first 600 frames were used to detect and model 5 objects in the scene, while tracking the camera. We implemented a simple hand-detector that is used to recognise when an object is touched, and as soon as the person interacts with the spray-bottle, the object is tracked reliably until it is placed back on the table at frame 1100.

#### 6.2.2 Augmented reality

Visual SLAM is a building block of many augmented reality systems and we believe that adding semantic information enables new kinds of applications. To illustrate that MaskFusion can be used for augmented reality applications, we implemented demos that rely on the table at frame 1100.

**Calories demo** This prototype aims at estimating the calories of an object-based on its class and shape. By estimating body volumes, using simple primitive fitting, and providing a database with calories per volume unit ratios for different classes, it is straightforward to augment footage with the desired information. Experiments based on this prototype are shown in Figure 11.

**Skateboard demo** Another demo program presents a virtual character that actively reacts to its environment. As soon as the skateboard appears in the scene the character jumps and remains on it, as depicted in Figure 10. Note that the character stays attached to the board even after a person kicks it and sets it into motion. This requires accurate tracking of the skateboard and camera at the same time.
Figure 8: Comparison of labelling performance over time. Results of Mask-RCNN (MRCNN) and Mask-RCNN followed by our geometric segmentation pipeline (MRCNN+GEOM) are frame-independent and variations in quality are only due to changes in camera perspective. The blue graph (Ours) shows the intersection-over-union correlating ground-truth 2D labels with the projection of the reconstructed 3D model.

Figure 9: Reconstruction of a bleach bottle from the YCB dataset. The average distance of a reconstructed surfel to a point on the ground-truth model is 7.0mm with a standard deviation of 5.8mm.

Figure 10: AR application that shows a virtual character interacting with the scene.

6.3 Performance

The convolutional masking component runs asynchronously to the rest of MaskFusion and requires a dedicated GPU. It operates at 5Hz, and since it is blocking the GPU for long periods of time, we use another GPU for the SLAM pipeline, which operates at >30Hz if a single model is tracked. In the presence of multiple non-static objects, the performance declines and results in a frame-rate of 20Hz for 3 models. Our test system is equipped with two Nvidia GTX Titan X and an Intel Core i7, 3.5GHz.

7 CONCLUSIONS

This paper introduced MaskFusion, a real-time visual SLAM system that utilises semantic scene understanding to map and track multiple objects. While inferring semantic labels from 2D image data, the system maintains independent 3D models for each object instance and for the background. We showed that MaskFusion can be used to implement novel augmented reality applications or perform common robotics tasks.

While MaskFusion makes meaningful progress towards achieving an accurate, robust and general dynamic and semantic SLAM system, it comes with limitations in the three main problems it addresses: recognition, reconstruction and tracking. Regarding the recognition, MaskFusion can only recognise objects from classes on which MaskRCNN [15] has been trained (currently the 80 classes of the MS-COCO dataset) and does not account for miss-classification of object labels. Secondly, although MaskFusion can cope with the presence of some non-rigid objects, such as humans, by removing them from the map, tracking and reconstruction is limited to rigid objects. Thirdly, tracking small objects with little geometric information when no 3D model is available can result in errors. Solving these limitations opens up opportunities for future work.

ACKNOWLEDGMENTS

This work has been supported by the SecondHands project, funded from the EU Horizon 2020 Research and Innovation programme under grant agreement No 643950.
Figure 12: Overview of evaluation sequences.
Figure 13: A series of 6 frames, illustrating the recognition, tracking and mapping capabilities of MaskFusion. While a keyboard (grey), vase (pink), teddy-bear (white) and spray-bottle (orange) were detected from the beginning, the ball (blue) appeared between frame 300 and 600. The right hand side shows the recognition and estimated normals. The spray-bottle was moved by a person between frame 600 and 1000, but MaskFusion explicitly avoided to reconstruct person-related geometry.

REFERENCES

[1] Matterport implementation of mask-rcnn. https://github.com/matterport/Mask_RCNN Accessed: 2018-02-02.
[2] Dragomir Anguelov, B Taskarf, Vassil Chatabalbashiev, Daphne Koller, Dinkar Gupta, Geremy Heitz, and Andrew Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 2, pages 169–176. IEEE, 2005.
[3] Ioan Andrei Barsan. Simultaneous localization and mapping in dynamic scenes. Master’s thesis, ETH, Zurich, 2017.
[4] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. The International Journal of Robotics Research, 36(3):261–288, April 2017.
[5] Rodrigo L Carceroni and Kirilakos N Kutulakos. Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance. International Journal of Computer Vision, 49(2-3):175–214, 2002.
[6] Robert Oliver Castle, Darren J Gawley, Georg Klein, and David W Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In Robotics and Automation, 2007 IEEE International Conference on, pages 4102–4107. IEEE, 2007.
[7] Xiaobai Chen, Alexey Golovinskiy, and Thomas Funkhouser. A benchmark for 3d mesh segmentation. In ACM SIGGRAPH 2009 Papers, SIGGRAPH ’09, pages 73:1–73:12, New York, NY, USA, 2009. ACM.
[8] Javier Civera, Dorian Gálvez-López, Luis Riauzuela, Juan D Tardós, and JMM Montiel. Towards semantic slam using a monocular camera. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 1277–1284. IEEE, 2011.
[9] A. Collet, S. S. Srinivasya, and M. Hebert. Structure discovery in multimodal data: A region-based approach. In 2011 IEEE International Conference on Robotics and Automation, pages 5695–5702, May 2011.
[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2017.
[11] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In Proceedings of the Ninth IEEE International Conference on Computer Vision-Volume 2, page 1403. IEEE Computer Society, 2003.
[12] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. Fanello, A. Kowdle, S. Orts Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. In ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques, 2016.
[13] Ross Finnn, Thomas Whelan, Michael Kaess, and John J. Leonard. Efficient incremental map segmentation in dense rgb-d maps. In IEEE Intl. Conf. on Robotics and Automation, ICRA, (Hong Kong), June 2014.
[14] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. Artificial Intelligence Review, 43(1):55–81, 2015.
[15] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In The IEEE International Conference on Computer Vision (ICCV), Oct 2017.
[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–777, 2016.
[17] P. Henry, D. Fox, A. Bhowmik, and R. Mongia. Patch volumes: Segmenttion-based consistent mapping with rgb-d cameras. In 2013 International Conference on 3D Vision - 3DV 2013, pages 398–405, June 2013.
[18] P. Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. The International Journal of Robotics Research, 31(5):647–663, 2012.
[19] 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, May 2014.
[20] Matthias Inmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In European Conference on Computer Vision, pages 362–379. Springer, 2016.
[21] Mariano Jaimez, Christian Kerl, Javier Gonzalez-Jimenez, and Daniel Cremers. Fast odometry and scene flow from rgb-d cameras based on geometric clustering. In Robotics and Automation (ICRA), 2017 IEEE International Conference on, pages 3992–3999. IEEE, 2017.
[22] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. Object discovery in 3d scenes via shape analysis. In International Conference on Robotics and Automation (ICRA), 2013.
[23] 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, June 2013.
[24] C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2100–2106, Nov 2013.

[25] Georg Klein and David 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.

[26] Hema S. Koppula, Ahbishek Anand, Thorsten Joachims, and Ashutosh Saxena. Semantic labeling of 3d point clouds for indoor scenes. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 24, pages 244–252. Curran Associates, Inc., 2011.

[27] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[28] I. MayCormac, 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, May 2017.

[29] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion. In 2009 IEEE Intelligent Vehicles Symposium, pages 215–220, June 2009.

[30] 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.

[31] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In ECCV, 2012.

[32] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[33] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR, 2011.

[34] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtm: Dense tracking and mapping in real-time. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 2320–2327. IEEE, 2011.

[35] Hanspeter Pfister, Matthias Zwicker, Jeroen van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’00, pages 335–342, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[36] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollr. Learning to refine using local convexity. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 304–311, June 2014.

[37] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 573–580. IEEE, 2012.

[38] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6565–6574, July 2017.

[39] K. Tateno, F. Tombari, and N. Navab. Real-time and scalable incremental segmentation on dense slam. In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015.

[40] K. Tateno, F. Tombari, and N. Navab. When 2.5d is not enough: Simultaneous reconstruction, segmentation and recognition on dense slam. In In. Proc. Int. Conf. on Robotics and Automation (ICRA), May 2016.

[41] Andre Uckermann, Christof Elbrechter, Robert Haschke, and Helge Ritter. 3d scene segmentation for autonomous robot grasping. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 1734–1740. IEEE, 2012.

[42] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. The International Journal of Robotics Research, 26(9):889–916, September 2007.

[43] Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. In-hand scanning with online loop closure. In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 1630–1637. IEEE, 2009.

[44] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. ElasticFusion: Dense SLAM without a pose graph. In Robotics: Science and Systems (RSS), Rome, Italy, July 2015.

[45] T. Whelan, J. B. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard. Kintinuous: Spatially extended kinectfusion. In Workshop on RGB-D: Advanced Reasoning with Depth Cameras, in conjunction with Robotics: Science and Systems, 2012.

[46] Xuehan Xiong and Daniel Huber. Using context to create semantic 3d models of indoor environments. In BMVC, pages 1–11, 2010.

[47] M. Zollhöfer, M. Niessner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an rgb-d camera. ACM Trans. Graph., 33(4), 2014.

[48] Dapeng Zou and Ping Tan. Coslasm: Collaborative visual slam in dynamic environments. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2013.