EM-Fusion: Dynamic Object-Level SLAM With Probabilistic Data Association

Michael Strecke and Jörg Stückler
Embodied Vision Group, Max Planck Institute for Intelligent Systems
{michael.strecke,joerg.stueckler}@tue.mpg.de

Abstract

The majority of approaches for acquiring dense 3D environment maps with RGB-D cameras assumes static environments or rejects moving objects as outliers. The representation and tracking of moving objects, however, has significant potential for applications in robotics or augmented reality. In this paper, we propose a novel approach to dynamic SLAM with dense object-level representations. We represent rigid objects in local volumetric signed distance function (SDF) maps, and formulate multi-object tracking as direct alignment of RGB-D images with the SDF representations. Our main novelty is a probabilistic formulation which naturally leads to strategies for data association and occlusion handling. We analyze our approach in experiments and demonstrate that our approach compares favorably with the state-of-the-art methods in terms of robustness and accuracy.

1. Introduction

RGB-D cameras are popular devices for dense visual 3D scene acquisition. Most approaches to simultaneous localization and mapping (SLAM) with RGB-D cameras only map the static part of the environment and localize the camera within this map. While some approaches filter dynamic objects as outliers from the measurements, SLAM of multiple moving objects has attracted only little attention so far. In many applications of robotics and augmented reality (AR), however, agents interact with the environment and hence the environment state is dynamic. Approaches that concurrently track multiple moving objects hence have rich potential for robotics and AR applications.

In this paper, we propose a novel approach to dynamic SLAM that maps and tracks objects in the scene. We detect objects through instance segmentation of the images and subsequently perform tracking and mapping of the static background and the objects. In previous approaches [15, 16, 27], data association of measurements to objects is either solved through image-based instance segmentation or by raycasting in the maps. We propose to determine the unknown association of pixels to objects in a probabilistic expectation maximization (EM [3]) formulation which estimates the soft association likelihood from the likelihood of the measurements in our map representation. The probabilistic association provides additional geometric cues and implicitly handles occlusions for object segmentation, tracking, and mapping (see Fig. 1). We represent the object maps by volumetric signed distance functions (SDFs). We augment the maximum likelihood integration of the SDF from depths to incorporate their association likelihood. The probabilistic data association facilitates the direct alignment of the depth maps with the SDF object maps. This avoids projective data association through raycasting which is needed for the ICP algorithm. In our experiments, we evaluate our approach on several datasets and demonstrate superior performance over the state-of-the-art methods. Our results demonstrate that proper probabilistic treatment of data associations is a key ingredient to robust object-level SLAM in dynamic scenes. In summary, we make the following contributions in our work,

- We propose a probabilistic EM formulation for dynamic object-level SLAM that naturally leads to data association and occlusion handling strategies.
- Based on our EM formulation, we approach multi-object tracking as direct alignment of RGB-D images with SDF object representations and evaluate this tracking approach for dense dynamic SLAM.
- Our approach achieves state-of-the-art performance on several datasets for dynamic object-level SLAM.

2. Related work

Static SLAM: Simultaneous localization and mapping (SLAM) with RGB-D sensors has seen tremendous progress quickly after the sensors have become broadly available on the market. KinectFusion [13] is a prominent approach that incrementally tracks the camera motion and maps the environment densely in volumetric signed distance function (SDF) grids. Several other RGB-D SLAM approaches have been proposed that differ in tracking methods such as ICP [13], direct image alignment [10] or SDF

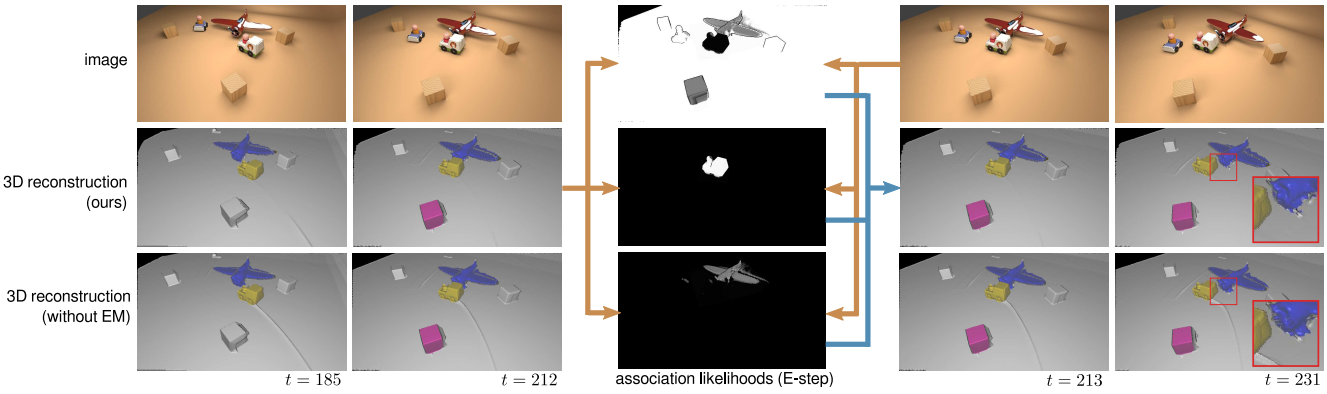

Figure 1. Dynamic object-level SLAM with probabilistic data association. We infer the association likelihood of pixels with objects in an expectation-maximization framework. The probabilistic data association improves accuracy and robustness of tracking and mapping. It implicitly handles occlusions. The E-step estimates the association likelihoods based on the data likelihood of the current image given the latest object maps and poses. In the M-step poses and map are updated with the measurements according to the association likelihoods. Association likelihoods are visualized for the background (top), the train (middle) and the airplane (bottom). The moving train occludes the table and the airplane which is well recovered by the association likelihoods. Without association likelihoods, artifacts are integrated into the map due to wrong data association.

alignment [4], and map representations such as surfels [9] or keyframes [10]. Extensive research has gone into scaling the approaches to large environments [25, 14] or supporting loop-closing [10, 26] to reduce drift. Some approaches also consider the creation of object-level maps [17, 12], but assume the objects to remain static.

Dynamic SLAM: Research on tracking and reconstruction of articulated objects such as human body parts [23, 24] or robots [18, 6] is related to dynamic SLAM. Recently, some RGB-D SLAM methods have been proposed that represent and track moving rigid objects. An early approach extends keyframe-based RGB-D SLAM to object-level dynamic SLAM [20]. The approach segments moving objects between RGB-D frames [21] and builds a keyframe pose graph for associated motion segments in the keyframes. CoFusion [15] extends surfel-based representations for moving objects. It combines geometric with motion segmentation to detect moving objects. Tracking camera motion with respect to the scene background and the objects is based on ICP alignment using geometry and color cues. MaskFusion [16] does not use motion segmentation but fuses geometric with a deep-learning based instance segmentation (Mask R-CNN [7]). MID-Fusion [27] follows a similar approach, but represents the 3D map in volumetric SDFs using octrees. We also represent objects in using SDFs but formulate tracking using efficient but accurate direct SDF alignment [4]. We also propose novel strategies for handling occlusions and disocclusions.

3. Proposed Method

Our dynamic SLAM approach performs incremental tracking and mapping of objects and the static background. We propose a probabilistic formulation for tracking and mapping of multiple objects which naturally leads to a principled method for data association and occlusion handling. We represent the 3D shape of objects and background in volumetric SDF representations which we estimate from depth images. New object instances are initially detected and segmented using a semantic appearance-based deep learning approach (Mask R-CNN [7]).

3.1. Probabilistic Dynamic Tracking and Mapping

We formulate SLAM as maximum likelihood estimation of the camera trajectory and the map from visual observations $\mathbf{z}_t$ (the depth images). The map is composed of separate TSDF volumes $\mathbf{m} := \{\mathbf{m}_i\}_{i=0}^{N}$ for the background ($\mathbf{m}_0$) and N objects. In each camera frame at time t, we track the camera pose with regard to the objects and background with distinct poses $\boldsymbol{\xi}_t := \{\boldsymbol{\xi}_{t,i}\}_{i=0}^{N}$, $\boldsymbol{\xi}_{t,i} \in \mathrm{SE}(3)$. We choose incremental tracking and mapping in which we optimize the joint posterior likelihood of the map and the camera poses in the current frame, given all images so far,

$$\begin{aligned}\arg\max_{\mathbf{m},\boldsymbol{\xi}_t} p(\mathbf{m}, \boldsymbol{\xi}_t \mid \mathbf{z}_{1:t}) = \\ \arg\max_{\mathbf{m},\boldsymbol{\xi}_t} p(\mathbf{z}_t \mid \mathbf{m}, \boldsymbol{\xi}_t)\, p(\mathbf{m} \mid \mathbf{z}_{1:t-1})\, p(\boldsymbol{\xi}_t). \quad (1)\end{aligned}$$

We optimize the posterior separately first for the camera pose, then for the map. By causality, each pixel measurement can only be attributed to one of the objects or the background, such that we also need to find the association of each pixel u to one of the objects. This association is a latent variable $c_t = \{c_{t,u}\}, c_{t,u} \in \{0, \ldots, N\}$ in our probabilistic model which we infer during the tracking and mapping.

3.2. Expectation Maximization Framework

Expectation-maximization (EM) is a natural framework for our problem of finding the latent data association with the map and camera pose estimates. In EM, we treat the map and camera poses as parameters $\boldsymbol{\theta}$ to be optimized. In the E-step, we recover a variational approximation of the association likelihood given the current parameter estimate from the previous EM iteration,

$$q(c_t) \leftarrow \arg\max_{q(c_t)} \sum_{c_t} q(c_t) \ln p(\mathbf{z}_t, c_t \mid \boldsymbol{\theta}). \tag{2}$$

The maximum is achieved for $q(c_t) = p(c_t \mid \mathbf{z}_t, \boldsymbol{\theta})$. For the M-step, we maximize the expected log posterior under the approximate association likelihood

$$\boldsymbol{\theta} \leftarrow \arg\max_{\boldsymbol{\theta}} \sum_{c_t} q(c_t) \ln p(\mathbf{z}_t, c_t \mid \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}). \tag{3}$$

Note that $p(\boldsymbol{\theta}) = p(\mathbf{m} \mid \mathbf{z}_{1:t-1})\, p(\boldsymbol{\xi}_t)$.

In our case the E-step can be performed by evaluating

$$p(c_t \mid \mathbf{z}_t, \boldsymbol{\theta}) = \frac{p(\mathbf{z}_t \mid c_t, \boldsymbol{\theta}) p(c_t \mid \boldsymbol{\theta})}{\sum_{c_t'} p(\mathbf{z}_t \mid c_t', \theta) p(c_t' \mid \boldsymbol{\theta})}. \tag{4}$$

Since we treat data and association likelihood stochastically independent between pixels, the association likelihood can be determined for each pixel individually. Assuming uniform prior association likelihood, we arrive at

$$p(c_t \mid \mathbf{z}_t, \boldsymbol{\theta}) = \frac{p(\mathbf{z}_t \mid c_t, \boldsymbol{\theta})}{\sum_{c_t'} p(\mathbf{z}_t \mid c_t', \boldsymbol{\theta})}. \tag{5}$$

The M-step is solved individually per object by taking into account the association likelihood of the pixels to the objects. We optimize first for the camera poses in the previous map and then integrate the measurement into the map using the new pose estimates. In the following, we detail the steps in our pipeline that implement the EM algorithm.

3.3. Image Preprocessing and Projection

We apply a bilateral filter on the raw depth images to smoothen depth quantization artifacts. From the filtered depth maps D we compute 3D point coordinates $\mathbf{p} = \pi^{-1}(\mathbf{u}, D(\mathbf{u})) \in \mathbb{R}^3$ at each pixel $\mathbf{u} \in \mathbb{R}^2$, where we define $\pi^{-1}(\mathbf{u}, D(\mathbf{u})) := D(\mathbf{u})\, \mathbf{C}^{-1} (u_x, u_y, 1)^\top$ and $\mathbf{C}$ is the camera intrinsics matrix of the calibrated pinhole camera.

3.4. Map Representation

We represent background and objects maps by volumetric SDFs. The SDF $\psi(\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ yields the signed distance of a point $\mathbf{p}$ to the closest surface represented by the SDF. The object surface is determined by the zero level-set $\left\{\mathbf{p} \in \mathbb{R}^3 : \psi(\mathbf{p}) = 0\right\}$ of the SDF. We implement the volumetric SDF through discretization in a 3D grid of voxels. The SDF value at a point within the grid is found through trilinear interpolation. We maintain several SDF volumes: one background volume (resolution 512^3) and several smaller SDF volumes, one for each detected object (initialized with a size of 64^3 and resized as needed, s. Sec, 3.5).

3.5. Instance Detection and Segmentation

For instance detection and segmentation we mostly follow [12], but adapt the approach for dynamic scenes. As in [12] we use Mask R-CNN [7] to detect and segment object instances. The Mask R-CNN detector runs at a lower processing rate (sequentially every 30 frames) than the remaining SLAM pipeline, hence, we only have detections available for a subset of frames. If a detection result is available, we match the detections with the current objects in the map and create new objects for unmatched detections.

Similar to [12] we recursively estimate the foreground probability $p_{fg}(\mathbf{p} \mid i) = Fg_i(\mathbf{p}) / (Fg_i(\mathbf{p}) + Bg_i(\mathbf{p}))$ of points $\mathbf{p}$ through counts in the corresponding voxels. The foreground and background counts $Fg_i(v)$ and $Bg_i(v)$ of each voxel v are updated using the associated segments,

$$\begin{aligned} Fg_i(v) &\leftarrow Fg_i(v) + p_{fg}^{MRCNN}(v) \\ Bg_i(v) &\leftarrow Bg_i(v) + \left(1 - p_{fg}^{MRCNN}(v)\right) \end{aligned} \tag{6}$$

The voxels are projected into the image to determine the segmentation likelihood $p_{fg}^{MRCNN}(v)$ in the associated segment from Mask R-CNN. During raycasting for visualization and generation of model masks, a point $\mathbf{p}$ from object i is only rendered if $p_{fg}(\mathbf{p} \mid i) > 0.5$ and there is no other model along that ray with a shorter ray distance. To account for possible occlusions, we only perform the update in (6) in unoccluded regions, i.e., where the projected mask of the object volume fits the fused segmentation from all objects.

For matching detections with objects, we find the reprojected segmentations of the objects in the map within the current image using raycasting. We determine the overlap of the reprojected segmentations with the detected segments by the intersection-over-union (IoU) measure. Segments are associated if their IoU is largest and above a threshold (0.2 in our experiments). Similar to [12], unmatched segments are used to create new objects by calculating the 10^{th} and 90^{th} percentiles of the pointcloud generated from the depth image masked by a segment and using them to determine the volume center $\mathbf{c}_i$ and size s_i (see [12] for details). We choose a padding factor of 2.0 around these percentiles for the volume size and set the initial volume resolution r_i to 64 along each axis, yielding a voxel size of $v_i = \frac{s_i}{r_i}$. If new detections matched with an existing model fall outside the existing volume, it is resized by determining an increased r_i required to fit the new detection and shifting $\mathbf{c}_i$ by a multiple of v_i so that it is still in the center of the volume.

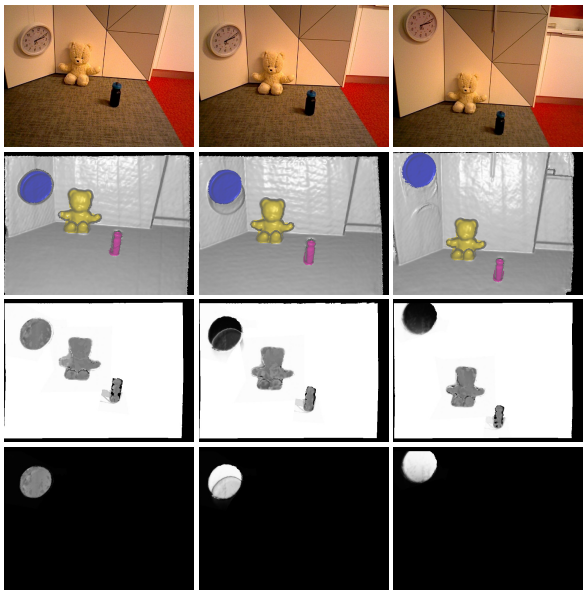

Figure 2. Pixel association likelihood. The E-step of our EM method determines the association likelihood (black: 0, white: 1) for the background (third row) and all objects (fourth row: clock). The association likelihood is determined from the data likelihood of the pixels in all objects given the current pose and map estimates (second row, object segments overlaid by color). Before the clock starts to move, the association weight is equally distributed between the background and the clock model. While the clock moves upwards, the background above the clock becomes occluded and the clock measurements are stronger associated with the object map than with the background.

The new volume is only initialized if its center $\mathbf{c}_i$ is within 5 m from the camera and the volumetric IoU with any other volume is lower than 0.5. Since Mask R-CNN can deliver false detections, we follow [12] and maintain an existence probability $p_{ex}(i) = Ex(i)/(Ex(i) + NonEx(i))$, where for each frame with a Mask R-CNN segmentation available $Ex(i)$ is incremented if the object is matched to a segment and otherwise $NonEx(i)$ is incremented. We delete objects where $p_{ex}(i) < 0.1$.

3.6. Data Association

We associate the pixels $\mathbf{u}$ in the current frame according to Eq. (5). Let $\mathbf{p}_i := \mathbf{T}(\boldsymbol{\xi}_i)\,\overline{\pi^{-1}}(\mathbf{u}, D(\mathbf{u}))$ be the local point coordinate of pixel $\mathbf{u}$ in the coordinate frame of object i, where we denote $\overline{\mathbf{p}} := \left(\mathbf{p}^\top, 1\right)^\top$. We model the data likelihood of a pixel that falls inside the map volume of object c_t with a mixture distribution,

$$p(\mathbf{u} \mid c_t, \boldsymbol{\theta}) = \alpha \frac{1}{2\sigma} \exp\left(-\frac{|\psi_{c_t}(\mathbf{p}_{c_t})|}{\sigma}\right) p_{fg}(\mathbf{p}_{c_t} \mid c_t) + (1-\alpha)\, p_{\mathcal{U}}(\mathbf{p}_{c_t}), \quad (7)$$

where ψ_{c_t} is the SDF of object c_t. The mixture is composed of a Laplace distribution which explains the measurement within the object, and a uniform component $p_{\mathcal{U}}$ that models outlier measurements and objects that are not yet detected and missing in the multi-object map. If the pixel is not within the map volume of object c_t, we set its data likelihood to zero for this object. Hence, the association likelihood is $p(c_t \mid \mathbf{u}, \boldsymbol{\theta}) = \frac{p(\mathbf{u}|c_t,\boldsymbol{\theta})}{\sum_{c'_t} p(\mathbf{u}|c'_t,\boldsymbol{\theta})}$.

Occlusions are implicitly handled by our data association approach. If an object is occluded by another object in the map, the association likelihood will be higher within the occluding object. This results in a lower weight for the measurements in the occluded object for tracking and map integration. Fig. 2 illustrates such a case for a clock which is moved upwards along a wall.

3.7. Tracking

Most existing approaches to dynamic multi-object SLAM employ a variant of the iterative closest points (ICP [2]) algorithm for tracking the camera pose. This requires that a point cloud is extracted from the existing TSDF volume and associations are found between this point cloud and the depth image. A typical approach with SDF map representations is to apply raycasting to determine the zero-crossings along the line-of-sight of the pixels. The point clouds are aligned using non-linear least squares techniques. In this approach, depth measurements are associated projectively with the zero-level surface.

We instead follow the approach in [4] and associate the depth measurements with the closest point on the surface. This is achieved by minimizing the signed distance of the measured points to the surface which is directly given by the SDF function at the points. The main advantage of this strategy is that pixels are associated with the correct part of the implicit surface using only one trilinear interpolation lookup per pixel in each iteration of the algorithm. In ICP, the projective association is only performed once and requires several lookups per pixel until a surface is found.

For the M-step in Eq. (3), we estimate the camera pose with regard to an SDF volume by minimizing

$$E(\boldsymbol{\xi}) = \frac{1}{2} \sum_{\mathbf{u} \in \boldsymbol{\Omega}} q(c_u)\, |\psi\left(\mathbf{T}(\boldsymbol{\xi})\, \overline{\mathbf{p}}(\mathbf{u})\right)|_\delta\,, \quad (8)$$

where $\mathbf{p}(\mathbf{u}) := \pi^{-1}(\mathbf{u}, D(\mathbf{u}))$ and $q(c_u)$ is the association likelihood of pixel $\mathbf{u}$ for the object/background. We use the Huber norm with threshold δ to achieve robustness with regard to outliers.

We optimize Eq. (8) using the iteratively reweighted non-linear least squares (IRLS) algorithm. Since the camera poses are in $\mathrm{SE}(3)$, we optimize Eq. (8) by reformulating it with a local parametrization using the Lie algebra $\mathfrak{se}(3)$. To this end, we apply local increments $\boldsymbol{\delta\xi} \in \mathfrak{se}(3)$ to the current solution for $\boldsymbol{\xi}$ in each iteration which we linearize

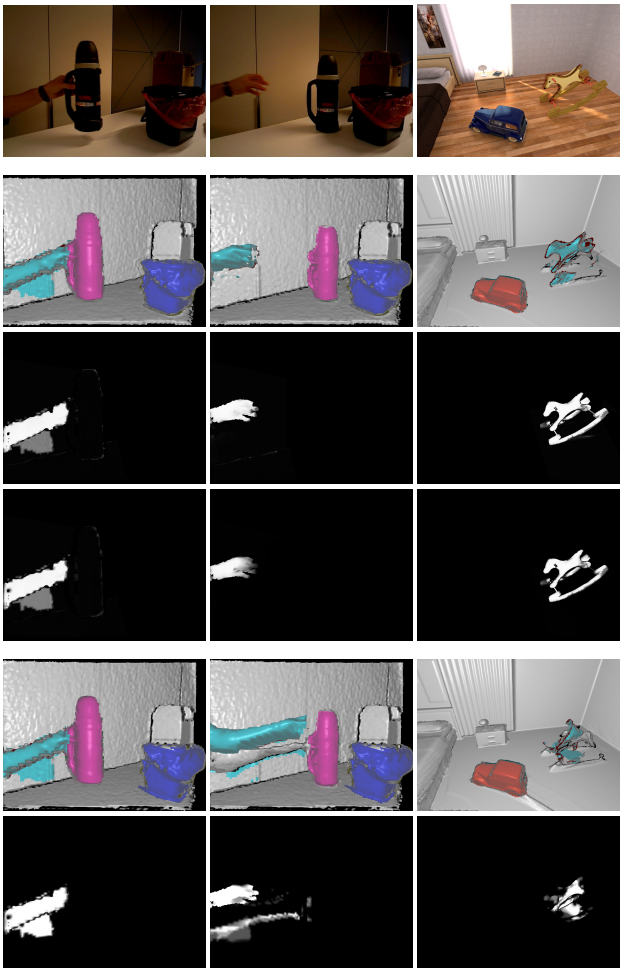

Figure 3. Tracking with association likelihoods. Probabilistic data association helps to overcome inaccuracies of the instance segmentation with geometric cues and makes the tracking more robust. From top to bottom: RGB images, our 3D reconstruction with reprojected object segmentation, association likelihood for the hand/horse object, our total pixel weights for tracking for the hand/horse object, 3D reconstruction with foreground probability instead of the association likelihood, total tracking weights with foreground probability instead of association likelihood.

at $\boldsymbol{\delta\xi} = \mathbf{0}$. Consequently, Eq. (8) becomes

$$E(\boldsymbol{\delta\xi}) = \frac{1}{2} \sum_{\mathbf{u} \in \boldsymbol{\Omega}} q(c_u)\, w_u \left(\psi\left(\mathbf{T}(\boldsymbol{\xi})\mathbf{T}(\boldsymbol{\delta\xi})\, \overline{\mathbf{p}}(\mathbf{u})\right)\right)^2, \quad (9)$$

with weights w_u which are adapted in each iteration to implement the Huber norm. We additionally weigh the individual terms in the sum in (9) by the map confidence $W(\pi^{-1}(\mathbf{u}, D(\mathbf{u})))/\max_{\mathbf{u}' \in \boldsymbol{\Omega}} W(\pi^{-1}(\mathbf{u}', D(\mathbf{u}')))$, where $W(\mathbf{p})$ is the accumulated integration weight (see section 3.8). It quantifies how certain we are about a surface estimate in the model. This robustifies the tracking when large objects enter the frame from the image boundary. The optimization is performed using the Levenberg-Marquardt method. This tracking optimization is first run on the background TSDF to estimate the updated camera pose before recomputing the association probabilities and running the same algorithm on each object TSDF for updating the individual object poses.

Fig. 3 illustrates the effectiveness of using the association likelihood for tracking. We compare our approach with just using the foreground probabilities without geometric cues by replacing $q(c_u)$ with $p_{fg}(\mathbf{p}_{c_u} \mid c_u)$ in Eq. (9). While the foreground probability also provides a segmentation cue, it is not sufficient for robust tracking due to the inaccurate instance segmentations by Mask R-CNN.

3.8. Mapping

Once the new camera poses $\boldsymbol{\xi}_t$ have been estimated, we implement the M-step (Eq. (3)) by integrating the depth maps into the background and object volumes. Following [5], we find the SDF as the maximum likelihood surface fit to the depth images using the recursive integration

$$\begin{aligned} \psi(v) &\leftarrow \frac{W(v)\psi(v) + q(c_u)\, d(v)}{W(v) + q(c_u)}, \\ W(v) &\leftarrow \min(W_{max}, W(v) + q(c_u)), \end{aligned} \quad (10)$$

where $d(v)$ is the measured depth difference of the voxel towards the integrated depth image. For implementing the M-step in Eq. (3), we incorporate the association likelihood $q(c_u)$ of the pixel $\mathbf{u}$ which passes through the voxel for computing the update weight. The cap on $W(v)$ prevents the model from becoming overconfident in the SDF estimate and allows for faster adaptation in case of inaccurate or missing segmentations of dynamic objects. Non-moving objects are initially integrated in the background map as well until the moving object map fits the measurements better. We consider backtracing these insertions too costly. One could reweigh the accumulated weight $W(v)$ with w_u for faster map updates which increases drift though.

4. Experiments

We evaluate the performance of our method qualitatively and quantitatively on datasets containing dynamic scenes published with [15] and the benchmark [22]. We employ the Mask R-CNN implementation of [1]. In our experiments, the truncation distance is chosen to be 10 times the voxel size for each TSDF volume and the parameter δ in (8) is twice the voxel size. In (7), we set $\sigma = 0.02$, $\alpha = 0.8$, and $p_{\mathcal{U}}(\mathbf{p}_{c_t}) = 1.0$. Mask R-CNN detections are only accepted if they are large enough (at least 40×40 pixels) and objects are classified as invisible (tracking and mapping unreliable) and deleted if their projected mask area within a region of 20 pixels from the image boundary is below this threshold. To avoid cluttering the scene with large volumes containing static objects for which Mask R-CNN usually

generates very inaccurate masks, we exclude a list of these object classes (*e.g.*, tables, beds, refrigerators, etc.) from the detections used for instantiating new object volumes. While one could implement a sliding window version for the background TSDF [25], we found that in our experiments a volume size of 5.12m with the camera positioned at the center of one of the sides of the volume usually worked well. The only exception from this strategy is the scene *Room4*, where we increased the volume size to 7.68m and moved the initial camera pose further inside the volume to keep the scene within the volume boundaries.

4.1. Quantitative Evaluation

Tracking of dynamic objects. We perform quantitative evaluation of dynamic object tracking on the synthetic scenes provided by the authors of Co-Fusion [15]. Remarkably, although many objects present in the scenes are not contained in the COCO dataset [11], Mask R-CNN manages to generate detections of most of the moving objects. We compare our method to Kintinuous (KT, [25]), ElasticFusion (EF, [26]), Co-Fusion (CF, [15]), and MaskFusion (MF, [16]). KT and EF are static SLAM systems that treat dynamic objects as outliers. CF uses geometric and motion segmentation for dynamic objects, while MF combines geometric segmentation with Mask R-CNN based instance segmentation. For the publicly available implementation of MF we adjusted the minimum number of pixels required for instantiating a new object model to work well on the sequences. We used the same threshold as in our approach, but MF still failed to instantiate an object instance for the rocking horse in the *Room4* scene.

The results of our evaluation are shown in Table 1. One can see that our method achieves competitive results. Especially for the dynamic objects, our method outperforms the competing dynamic object-level SLAM approaches CF and MF. The large camera tracking error wrt the static background (Static Bg) for MF in the *ToyCar3* scene is caused by a very late detection of one of the moving cars, causing significant drift at the beginning of the trajectory. This shows that the ICP tracking without a robust norm used in MF is sensitive to missing detections. Our robust tracking using direct SDF alignment and the Huber norm, however, manages to keep the trajectory error low.

Robust camera tracking. Similar to experiments performed in MaskFusion [16] and MID-Fusion [27], we can use Mask R-CNN detections with certain labels (*e.g.*, *person*) to exclude these labels from the reconstruction and tracking. In our approach, the association likelihoods already prevent parts of depthmaps projecting into foreground parts of object volumes from being integrated into the background volume used for camera tracking. We thus maintain object volumes for detected people but do not render them during raycasting for visualization. The association likelihood then tends to associate even non-rigidly moving people to the object volumes rather than the background, enabling us to robustly track the camera wrt. background.

We compare our method to five state-of-the-art dynamic SLAM approaches in Table 2. Two of these, joint visual

		KT	EF	CF	MF	**Ours**
ToyCar3	Static Bg	**0.10**	0.59	0.61	20.60	0.95
	Car1	-	-	7.78	1.53	**0.77**
	Car2	-	-	1.44	0.58	**0.18**
Room4	Static Bg	**0.16**	1.22	0.93	1.41	1.37
	Airship	-	-	0.91/ 1.01	13.62/ 2.29/ 3.46	**0.56/** 1.41/ 0.75
	Car	-	-	**0.29**	2.66	2.10
	Horse	-	-	5.80	-	**3.57**

Table 1. AT-RMSEs (in cm) of estimated trajectories for the synthetic sequences from Co-Fusion [15]. The Airship trajectory is split into multiple parts due to separate geometric segments and detections with too little overlap for assignment. Our method achieves competitive results with a static SLAM system (EF) for the static background and outperforms other dynamic SLAM approaches (CF, MF) on the objects.

	VO-SF	SF	CF	MF	MID-F	**Ours**
f3s static	2.9	1.3	1.1	2.1	1.0	**0.9**
f3s xyz	11.1	4.0	**2.7**	3.1	6.2	3.7
f3s halfsphere	18.0	4.0	3.6	5.2	**3.1**	3.2
f3w static	32.7	**1.4**	55.1	3.5	2.3	**1.4**
f3w xyz	87.4	12.7	69.6	10.4	6.8	**6.6**
f3w halfsphere	73.9	39.1	80.3	10.6	**3.8**	5.1

(a) Absolute trajectory (AT) RMSE (in cm)

	VO-SF	CF	SF	MF	**Ours**
f3s static	2.4	1.1	1.1	1.7	**0.9**
f3s xyz	5.7	2.7	2.8	4.6	**2.6**
f3s halfsphere	7.5	**3.0**	**3.0**	4.1	**3.0**
f3w static	10.1	22.4	1.3	3.9	**1.2**
f3w xyz	27.7	32.9	12.1	9.7	**6.0**
f3w halfsphere	33.5	40.0	20.7	9.3	**5.1**

(b) Relative pose (RP) RMSE (cm/s)

Table 2. Comparison of robust camera tracking wrt. the static background in dynamic scenes for different methods. Our approach provides state-of-the-art results and outperforms previous methods in the majority of sequences.

		w/o assoc.	w/o map conf.	Ours
Room4	Static Bg	1.42	**1.37**	**1.37**
	Airship	**0.49/** **1.13/** 1.24	0.73/ 1.47/ **0.75**	0.56/ 1.41/ **0.75**
	Car	**2.01**	2.11	2.10
	Horse	9.12	8.38	**3.57**

Table 3. Ablation study on the synthetic scene *Room4*. We compare AT-RMSE for our approach to not using association likelihoods, and to not using map confidence weights for tracking.

Figure 4. Qualitative evaluation on the real-world datasets published with Co-Fusion [15]. We demonstrate that we can handle fast movement (the second and the third image of the first dataset only 25 frames apart), as well as objects with relative weak geometric cues, such as the clock in the second dataset. Note that the left arm handing over the teddy is not detected in the last dataset. While it initially is integrated into the background it is quickly overriden by actual background depth soon after it moved out of view.

odometry and scene flow (VO-SF, [8]), and StaticFusion (SF, [19]) were designed for reconstructing the static background while ignoring dynamic parts. The remaining ones, CF [15], MF [16], MID-Fusion (MID-F, [27]) were designed for multi-object reconstruction. The latter two of these methods, like our approach, use Mask R-CNN [7] detections for instantiating objects. One can see that our method achieves competitive results in most cases, especially compared to MF [16] and MID-F [27]. Like all these methods, our method might fail if large undetected objects cover the major part of the image. Our results demonstrate that the combination of robust tracking and our data association strategy improves robustness on these sequences. The table rows are ordered approximately by scene difficulty, so the latter rows exhibit large dynamic parts with heavy occlusions. *f3s* abbreviates *freiburg3_sitting* while *f3w* stands for *freiburg3_walking*. MID-F did not report RP-RMSE and thus is not shown in Table 2 (b).

We further compare to MF [16] on the scene *f3_long_office_household* of the benchmark [22]. By exporting the relative trajectory of the teddy bear and the camera, we can compare the object trajectory to the ground truth camera trajectory as was done in [16]. While we achieve slightly worse results on the teddy bear trajectory (3.5cm, while MF achieved 2.2cm), our camera trajectory is more accurate (5.0cm compared to 8.9cm for MF). Note that while MF improved their camera trajectory wrt. the background to 7.2cm AT-RMSE when not tracking the teddy bear, we do not expect a notable change for this case in our approach since the teddy is implicitly reconstructed with partial association likelihood in the background and would be disassociated and removed from it if it started moving.

In Table 3, we do an ablation study to evaluate the contributions of different parts of our method. Since most objects only observe minor changes in their local topology (the Airship moving freely in the air, the car driving on the ground), and there are no large objects moving into view from the edge of the image, the effects of not using association likelihoods or map confidence weights for tracking are numerically negligible for most objects. However, the rocking

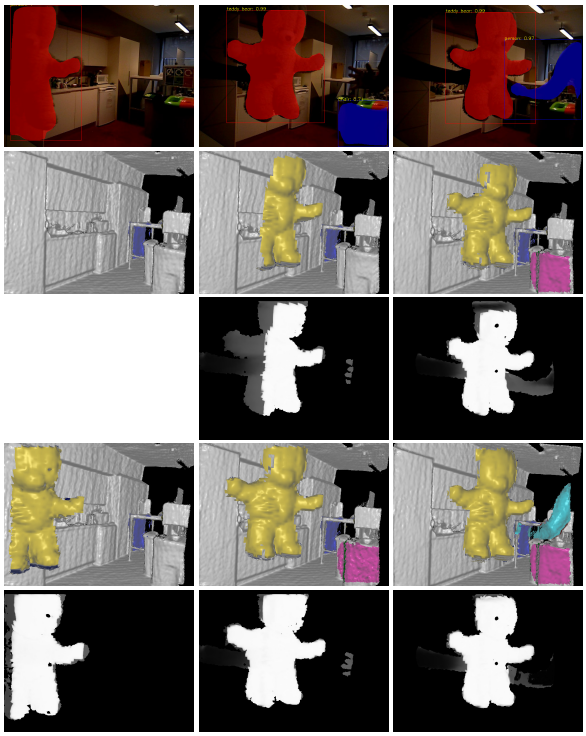

Figure 5. Incremental mask integration. From top to bottom: Masked RGB Frame, model output and association likelihood for teddy before mask integration, model output and association likelihoods after mask integration. One can see that the association likelihoods provide a soft geometric segmentation for the moving geometry inside the volume of the teddy object. It gets stronger for the pixels that actually belong to the object once Mask R-CNN confirms those pixels to belong to that object. Note that the teddy bear is first detected in frame 240 and thus does not have association likelihoods in this frame yet.

horse is subject to topology changes in its surrounding since wall and floor intersect the volume at different angles. We observe a significant improvement for this object in Tab. 3.

Computational performance. While our implementation is not yet tuned for runtime performance (*e.g.*, parallel processing of objects), the average runtime per frame on the CF-datasets [15] ranges from 106ms to 257ms on an Nvidia GTX 1080 Ti GPU with 11GB of memory and an Intel Xeon Silver 4112 CPU with 4 cores and 2.6 GHz. A more detailed analysis separating the runtime on detection frames as well as an ablation study on how varying detection frequencies affect trajectory coverage and accuracy can be found in the supplemental material.

4.2. Qualitative Evaluation

Figure 4 shows a qualitative evaluation on the real-world datasets published with Co-Fusion [15]. One can see, that we manage to reconstruct dynamic and static objects in these scenes if they are detected by Mask R-CNN. Note that some of the objects, like the trashcan in the first sequence are not contained in the set of classes that Mask R-CNN is trained on. Thus, the trashcan is not detected for a large number of frames and deleted because of a low existence probability p_{ex}. The bottle in the clock sequence is deleted after it is classified as "not visible" because it moves out of view and the number of pixels in view is too low.

We show how the incremental integration of foreground probabilities into object volumes improves the object masks in Figure 5. Finally, for a qualitative evaluation of the effect of the association likelihood, we refer to Figure 1, where moving objects leave a visible trace because their depth values are integrated into the background, and Figure 3, which shows that they help to improve the tracking quality by including geometric cues if Mask R-CNN segmentations do not fit the actual object shape.

5. Conclusions

In this paper we propose a novel probabilistic formulation for dynamic object-level SLAM with RGB-D cameras. We infer the latent data association of pixels with the objects in the map concurrently with the maximum likelihood estimates of camera poses and maps. The maps are represented as volumetric signed distance functions. For tracking, our probabilistic formulation facilitates direct alignment of depth images with the SDF representation. Our results demonstrate that proper probabilistic treatment of data associations is a key ingredient to robust tracking and mapping in dynamic scenes. To the best of our knowledge, our approach is the first that considers EM for dynamic object-level SLAM with RGB-D cameras.

Note that our approach treats the detected objects models always as dynamic. While our experiments have shown that their poses are stable in most settings for static objects, in future work an additional classification into static and dynamic objects might be developed to prevent drifting of static objects and to refine the camera pose by tracking it relative to the static object volumes. This might prove beneficial since the object volumes usually exhibit a higher relative resolution. In future work we further plan to integrate information from the RGB image for tracking to further increase the accuracy and robustness of the method in planar surfaces. Furthermore, more efficient data structures and global graph optimization are interesting directions to further scale our approach. Finally, we plan to investigate how our approach could be used on mobile manipulation platforms for the interactive perception of objects.

Acknowledgements. We acknowledge support from the BMBF through the Tuebingen AI Center (FKZ: 01IS18039B) and Cyber Valley. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Michael Strecke.

References

[1] Waleed Abdulla. Mask R-CNN for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`, 2017.

[2] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, Feb 1992.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[4] Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl, and Daniel Cremers. Real-time camera tracking and 3D reconstruction using signed distance functions. In *Proceedings of Robotics: Science and Systems (RSS)*, Berlin, Germany, June 2013.

[5] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM.

[6] Cristina Garcia Cifuentes, Jan Issac, Manuel Wüthrich, Stefan Schaal, and Jeannette Bohg. Probabilistic articulated real-time tracking for robot manipulation. *IEEE Robotics and Automation Letters (RA-L)*, 2(2):577–584, April 2017.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Oct 2017.

[8] Mariano Jaimez, Christian Kerl, Javier Gonzalez-Jimenez, and Daniel Cremers. Fast odometry and scene flow from RGB-D cameras based on geometric clustering. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3992–3999, May 2017.

[9] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3D reconstruction in dynamic scenes using point-based fusion. In *International Conference on 3D Vision (3DV)*, pages 1–8, June 2013.

[10] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual SLAM for RGB-D cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2100–2106, Nov 2013.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[12] John McCormac, Ronald Clark, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level SLAM. In *International Conference on 3D Vision (3DV)*, pages 32–41, Sep. 2018.

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

[14] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013.

[15] Martin Rünz and Lourdes Agapito. Co-Fusion: Real-time segmentation, tracking and fusion of multiple objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478, May 2017.

[16] Martin Rünz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20, Oct 2018.

[17] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. J. Kelly, and Andrew J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359, June 2013.

[18] Tanner Schmidt, Richard Newcombe, and Dieter Fox. DART: dense articulated real-time tracking with consumer depth cameras. *Autonomous Robots*, 39(3):239–258, Oct 2015.

[19] Raluca Scona, Mariano Jaimez, Yvan R Petillot, Maurice Fallon, and Daniel Cremers. StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, May 2018.

[20] Jörg Stückler and Sven Behnke. Hierarchical object discovery and dense modelling from motion cues in rgb-d video. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2502–2509. AAAI Press, 2013.

[21] Jörg Stückler and Sven Behnke. Efficient dense rigid-body motion segmentation and estimation in rgb-d video. *International Journal of Computer Vision (IJCV)*, 113(3):233–245, Jul 2015.

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

[23] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Trans. Graph.*, 35(4):143:1–143:12, July 2016.

[24] Dimitrios Tzionas and Juergen Gall. Reconstructing articulated rigged models from RGB-D videos. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 620–633, Cham, 2016. Springer International Publishing.

[25] Thomas Whelan, Michael Kaess, Maurice F. Fallon, Hordur Johannsson, John J. Leonard, and John B. McDonald. Kintinuous: Spatially extended KinectFusion. In *RSS Work-*

shop on RGB-D: Advanced Reasoning with Depth Cameras, Sydney, Australia, Jul 2012.

[26] Thomas Whelan, Stefan Leutenegger, Renato Salas Moreno, Ben Glocker, and Andrew Davison. ElasticFusion: Dense SLAM without a pose graph. In *Proceedings of Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[27] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. MID-Fusion: Octree-based object-level multi-instance dynamic SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. to appear.