

Probabilistic Object Reconstruction with Online Global Model Correction

Jack Hunt Victor A. Prisacariu Stuart Golodetz
Tommaso Cavallari Nicholas A. Lord Philip H. S. Torr
`{jmhunt,victor,smg,tommaso,nicklord,phst}@robots.ox.ac.uk`
Department of Engineering Science
University of Oxford, Parks Road, Oxford, OX1 3PJ

Abstract

In recent years, major advances have been made in 3D scene reconstruction, with a number of approaches now able to yield dense, globally-consistent models at scale. However, much less progress has been made for objects, which can exhibit far fewer unambiguous geometric/texture cues than a full scene, and thus are much harder to track against. In this work we present a novel probabilistic object reconstruction framework that simultaneously allows for online, implicit deformation of the objects surface to reduce tracking drift and handle loop closure events. Coupled with our probabilistic formulation is the use of a multi subsegment representation of the object, used to enforce global consistency, with segmentation of the object built in to the formulation. Finally, we employ a CRF framework to refine the overall segmentation, defined by a probability field over the object. We present compelling results over the current state-of-the-art object reconstruction work and demonstrate robustness and consistency w.r.t. established dense SLAM frameworks.

1. Introduction

Dense SLAM (Simultaneous Localisation and Mapping) has proven very effective for reconstructing moderately-sized scenes, with much recent research driven by the availability of inexpensive, consumer-grade depth sensing equipment [16, 17, 21]. However, accurate pose estimation in the presence of erroneous measurements and visual aliasing in the scene remains difficult to fully solve. Common approaches [1, 24] exploit geometric/texture cues to estimate the pose of each frame, but tracking can drift – or even fail completely – without reliable features against which to track. Loop closure events are another common source of tracking problems, with explicit handling often required to prevent errors in the model, exacerbating any existing tracking issues. These problems are only magnified when recon-



Figure 1: Chair and Dinosaur Head reconstructed with our system.

structing objects: an object’s surface will generally contain far fewer points than the full scene, and a lack of unambiguous points on the surface can lead to an increase in data association errors when attempting to recover the pose.

In this paper, we introduce a novel approach to reconstruct accurate, globally-consistent object models, introducing a probabilistic rigid-object reconstruction framework based on depth features. The framework facilitates the correction of tracking drift by representing the object as a collection of overlapping subsegments between which transformations may be inferred to maintain alignment, resulting in a consistent overall model. By combining these transformed surfaces, we extract an implicitly deformed surface, optimised for via the probabilistic formulation that follows. We utilise a volumetric representation for each of these object subsegments, with each voxel in a given subsegment having additional appearance posterior information pertaining to the voxel’s membership of the object. Over time, multiple volumes containing both surface and probabilistic appearance information are maintained and manipulated to yield a robust and temporally consistent model. Finally, we optimise for the optimum object segmentation within a CRF (Conditional Random Field) framework.

Recently, robust, globally-consistent, large scale scene reconstruction has been achieved by combining a multi-segment representation with loop closure detection and an online model correction algorithm [11], demonstrating the efficacy of this approach for larger scenes.

We perform both quantitative and qualitative experiments to compare our approach to the state-of-the-art object reconstruction approach of Ren et al. [33], demonstrating compelling improvements in both pose estimation and

model quality. In addition we demonstrate compelling results over the standard KinectFusion pipeline using an open implementation [21] as a benchmark.

2. Related Work

Object Reconstruction. In addition to scene-scale works, there has been much work on object reconstruction and object-centric SLAM. Kolev et al. [12] presented a probabilistic 3D segmentation and surface extraction algorithm based on a variational evolution of a level set representation, but did not handle loop closures and tested their approach only on images with no background noise. Weise et al. [29] presented an explicit, surfel-based reconstruction system for objects rotating in front of a 3D range scanner. They maintain an object topology graph to handle loop closures online, but they only deform the object model when detecting loop closure events, whereas our approach continuously updates the rigid transformations between object segments, making it easier for the user to see what the final model will look like. Ohno et al. [18] presented a robotic system for reconstructing unknown objects in an environment by pushing them and estimating their motion using 3D flow. Krainin et al. [13] present another robotic system that uses Kalman filtering and articulated ICP (Iterative Closest Point) to track both the robot’s manipulator and the object. They perform loop closure in a similar way to [29] and achieve very good surfel models of the object, but their approach requires specialist hardware. Cui et al. [4] presented an object reconstruction system based on ToF (Time of Flight) sensors. They use a super-resolution representation of chunks of raw depth images to reconstruct detailed models. Mihalyi et al. [14] used augmented reality markers to make it possible for untrained users to achieve robust object reconstructions. Their approach works for a range of objects, but the need to add markers to the scene in advance is quite limiting in practice. Narayan et al. [15] combine KinectFusion with visual hull techniques to reconstruct objects with concavities and translucencies. Panteleris et al. [19, 20] reconstruct objects by tracking hand-object manipulations. Their approach runs in real time, but they do not handle loop closures. Tzionas and Gall [28] also make use of hand-object interactions, presenting an elegant system that can reconstruct featureless and highly symmetric objects by tracking contact points between the hand and the object. Their system produces appealing results, but is not real-time and can fail if the fingers slip over the manipulated object. Gupta et al. [9] performed object reconstruction based on monocular, multi-view cues. They segment the objects using graph cuts and track based on geometric/textural cues, but do not handle loop closures and report fluctuating tracking quality caused by illumination variation and specular surfaces. Recently, Slavcheva et al. [26] presented an object reconstruction system that estimates poses by regis-

tering pairs of TSDF volumes. Their system handles loop closures and achieves high-quality results, but at the cost of relying on fiducial markers to improve their tracking and performing loop closure offline as a post-processing step.

The closest approach to ours is that of Ren et al. [33], who presented a probabilistic tracking and reconstruction system that reconstructs objects based on an appearance model, evolving a level set representation for voxels that belong to the object. However, they do not detect loop closures and are prone to tracking drift. Their later work [22] extended [33] to track multiple objects for which an initial model is available at run time. Our system makes no such assumption.

Dense 3D Reconstruction. Much recent work is inspired by the seminal KinectFusion work of Newcombe et al. [16]. This was used to build an implicit, voxel-based TSDF (Truncated Signed Distance Function) representation [5] of a small-scale environment, but could only reconstruct static scenes, and struggled to scale due to inefficient use of memory and difficulties in preventing significant tracking drift in larger-scale scenes. Scalability has progressively been addressed by a moving reconstruction window [23, 30], octrees [34], and sparse methods based on voxel hashing and streaming data to and from the GPU [17, 21]. This has made it possible to reconstruct static scenes whose size is only limited by available system memory, although reconstructing a large scene can still occupy significant space. Tracking drift has also been addressed to some extent, generally by detecting loop closures and either rigidly or non-rigidly deforming parts of the scene [35, 31, 32]. Other approaches exist that do not explicitly detect loop closures [7]. Recently, Dai et al. [6] introduced a system that improves pose estimation for large-scale scenes by considering each previously seen frame within a hierarchical framework and performing sparse feature matching to optimise for the camera pose. However, mismatches between keypoints are reported to have an impact on the robustness of their optimisation procedure. Kähler et al. [11] took a different approach, showing how to combine a multi-segment representation of the scene with loop closure detection and an online model correction algorithm to achieve accurate, globally-consistent scene reconstruction. They reduce drift by tracking only against recent segments of the scene and adjusting the poses between segments online, before refining the final model using pose graph optimisation. In this paper, we extend this latter approach to achieve globally-consistent reconstructions of objects.

3. Method

We divide our object model into subvolumes, each consisting of a TSDF, colour volume and object probability volume, and contains a rigid body transform that specifies its pose relative to the global coordinate frame. At each

new camera frame, we apply our segmentation model to the colour input image to construct an object probability map, and then accumulate the probabilities from this map into the object probability volume of the active subvolume (see Section 3.1). We also update the TSDF and colour volume of the active subvolume, and track against it using ICP. At the end of each frame, we run our novel online model correction algorithm (see Section 3.3), which infers the relative poses between the subvolumes to mitigate tracking drift. Once the reconstruction process is finished, we perform a CRF-based optimisation to refine the resulting object segmentation (see Section 3.5). Our approach is not tied to the use of any one probabilistic model, though in our experiments we use PwP (Pixel Wise Posteriors) [2]. An overview of our object reconstruction pipeline is shown in Figure 2.

3.1. Probabilistic Object Fusion Procedure

The surface map and camera pose are estimated using the standard pipeline of [16, 21]. The surface is represented as the zero level set of a TSDF discretised over voxels, with online weighted-mean fusion of new observations. Pose estimation via ICP is run quasi-simultaneously against the evolving map. Here, inspired by [12], we augment this procedure by estimating the posterior probability, per map voxel, of belonging to the object. This volume of posterior probabilities is updated on each frame, parallel to the fusion process in the mapping and pose estimation subsystems. The representation of the reconstructed object comprises multiple ‘subvolumes’, each pertaining to some patch on the object surface. New subvolumes are started when sufficiently many new voxels have been allocated and have had data integrated. By ensuring overlap between subvolumes, transformations between them can be found and pose inconsistencies addressed, online. Empirically, we define the threshold for starting a new subvolume to be when 50% of the voxels fused in to the current volume are newly observed points.

At each frame, the object probabilities for the visible voxels in the active submap are updated via an appearance-derived probability map for that frame. Under the assumption of conditional independence between frames (for sake of tractability), the posterior probability of a given voxel $\psi \in \Psi$ belonging to the object has the following form (noting that $\Phi \subset \Psi$):

$$P(\psi \in \Phi | \Omega, p) = \prod_{t=0}^{\infty} P(\psi_t \in \Phi | \Omega_t, p_t) \quad (1)$$

where Ψ is the volume of voxels for which measurements are accumulated, Φ is the volume of voxels pertaining to the object, Ω_t is the current image observation at time t and p_t is the currently tracked pose at time t . This encodes the probability of a voxel belonging to the object as the product

of instantaneous appearance-derived pixel-wise conditionals.

3.2. Probabilistic Formulation of Object Reconstruction

Central to the proposed system is a volume of posterior probabilities pertaining to a voxel wise membership of either the object set or the non object set. This allows formulation of the full joint distribution over the object as the Probabilistic Graphical Model of Figure 3(left).

Where Φ is the shape to be reconstructed (represented as a subset of voxels for which surface data has been integrated), u is the appearance model volume, L is the set of consistency constraints for each adjacent sub volume pair in the form of rigid transformations, Ω is the set of RGBD image pixels and p the set of poses over time.

This gives rise to the following factorisation of the distribution given in Figure 3(left)

$$P(\Phi, \Omega, p, u, L) = \prod_{\psi \in \Psi} \prod_{s, s' \in \mathcal{S}} P(\Phi | u_\psi, L_{s, s'}) \prod_{t=0}^{\infty} \prod_{p \in \mathcal{P}} P(u_v | \Omega_{p, t}, p_t) P(L_{s, s'} | \Omega_{p, t}, p_t) P(L_{s, s'}) P(p_t) P(\Omega_{p, t}) \quad (2)$$

where Ψ is the set of voxels across all sub volumes, \mathcal{P} is the set of RGBD pixels for a given frame and \mathcal{S} is the set of sub volumes. Where the notation $s, s' \in \mathcal{S}$ refers to pairs of overlapping subvolumes.

However, if pixel wise independence is assumed in the RGBD observations and temporal independence in the poses, the plate containing Ω and p can be removed as in Figure 3(right).

$$P(\Phi, \Omega, p, u, L) = \prod_{v \in \mathcal{V}} P(\Phi | u_v) \prod_{s, s' \in \mathcal{S}} P(u_v | \Omega, p, L_{s, s'}) P(L_{s, s'} | \Omega, p) P(L_{s, s'}) P(p) P(\Omega) \quad (3)$$

In practice this temporal independence assumption causes no issues. Furthermore, if voxel wise independence is assumed, the plate over voxels can be removed. Finally, assuming $P(p)$ and $P(\Omega)$ are uniform distributions, then we have the simpler distribution given by Figure 3(right).

This simpler distribution can be factorised as follows

$$P(\Phi, \Omega, p, u, L) = \prod_{s, s' \in \mathcal{S}} P(\Phi | u, L_{s, s'}) P(u | \Omega, p) P(L_{s, s'} | \Omega, p) P(L_{s, s'}) \quad (4)$$

The above formalisms describe a probabilistic framework in which online corrections can be made to the reconstructed model to counteract errors incurred by pose tracking inconsistencies. As with scene based dense SLAM systems [16, 21, 17], our system follows a pipeline that consists

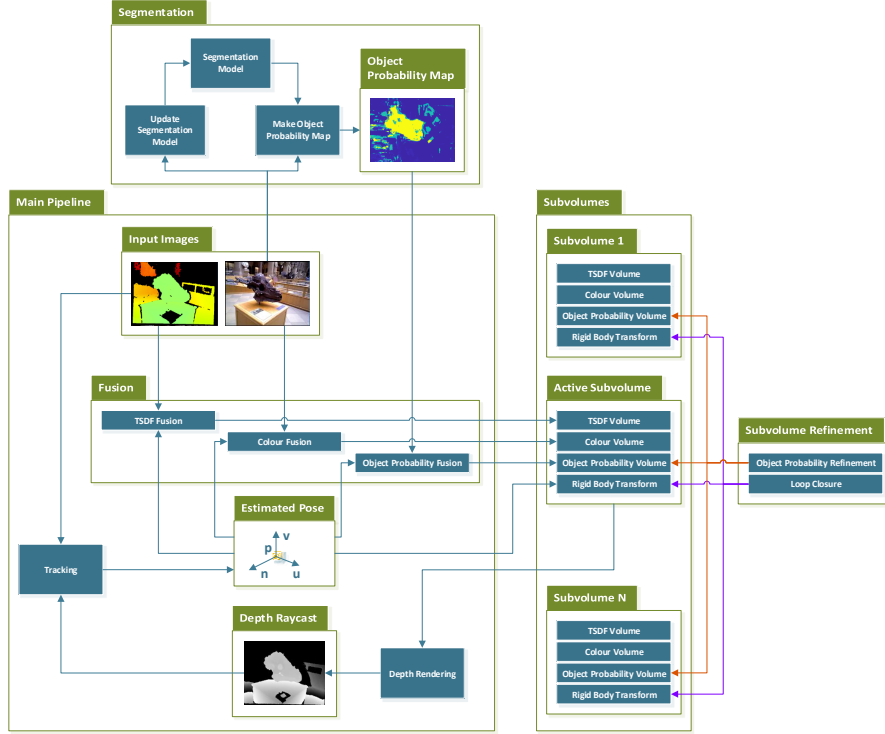


Figure 2: The pipeline of our object reconstruction approach.

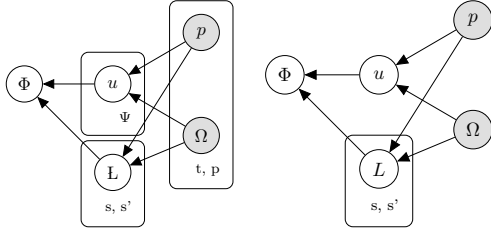


Figure 3: An initial probabilistic formulation of the object reconstruction pipeline (left) and a simpler formulation based on various independence assumptions (right).

of a tracking stage and an integration stage. However, our formulation of this pipeline consists of an additional, novel estimation module that relies on the use of a subvolume representation to correct tracking errors by applying transformations to the subsegments to align them when there are intra subsegment tracking inconsistencies. As inference on the joint distribution of our model is intractable, conditional independence assumptions are made that do not appear to cause any functional issues.

3.3. Online Model Correction

The tracking consistency constraint denoted by the variable L in the graphical models given by Figure 3 can be enforced in terms of minimising the disparity between adjacent subvolumes, such that the poses tracked in each subvolume are consistent. Given instantaneously inferred trans-

forms between subvolumes obtained from tracking results, the objective is to infer a robust, consistent deformation transformation for the subvolume pair. As such, for each pair of visible subvolumes s, s' , the following posterior must be maximised:

$$P(\Omega, p | L_{s,s'}) = \frac{P(L_{s,s'} | \Omega, p) P(\Omega | p) P(p)}{P(L_{s,s'})} \quad (5)$$

The intuition behind the above equation is that the deformation $L_{s,s'}$ applied to the object subvolume Φ_s should increase the probability of observing the current pose p given the current RGBD frame Ω by reducing the variance of the pose estimation result. As such, global tracking variance is reduced by enforcing local consistency, improving the quality of the reconstruction.

It should be noted that in our implementation the $P(\Omega | p)$ term is assumed to be uniform in the case of an RGBD sensor being used, however for applications such as monocular SLAM this term may be replaced with a noise model when there is significant uncertainty about the given depth map at each frame. The following proportionality to the distribution over deformations is made for two overlapping object subsegments $\Phi_s, \Phi_{s'} \in \Phi$, noting again that $\Phi \subset \Psi$. X is the set of valid voxel locations in Φ_s for which the currently accumulated appearance posterior is greater for foreground:

$$P(L_{s,s'} | \Omega, p) \propto P(\Phi_s(X)) | \Phi_{s'}(\Lambda(X)) P(L_{s,s'}) \quad (6)$$

Where the prior distribution over transformations $P(L_{s,s'})$ corresponds to the Gaussian Conjugate Prior $\mathcal{N}(L_{s,s'}|\mathbf{0}, \lambda^{-1})$. The log posterior of the above expression takes the following form:

$$\ln P(\Phi_s(x)|\Phi_{s'}(\Lambda(x)))P(L_{s,s'}) = m \ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{\psi \in \Phi_s} \left[\left(\Phi_s(x_\psi) - \Phi_{s'}(\Lambda(x_\psi)) \right)^2 - \lambda \|\mathbf{x}\|_2 \right] \quad (7)$$

Where $\Phi_s(\cdot)$ is a scalar valued SDF (Signed Distance Function) for the subsegment s , a discretised field of Φ_s (where $\Phi_s \subset \Phi$), as previously described. x is a point represented by a 3-vector and $\Lambda(\cdot)$ is a transformation function taking the following form:

$$\Lambda(x) = R(\rho_1, \rho_2, \rho_3)x + t \quad (8)$$

Where $R(\cdot)$ is a rotation matrix from the Special Orthogonal group $\mathbb{SO}(3)$ parameterised by the three Rodrigues Parameters [25] ρ_1, ρ_2 and ρ_3 . The \mathbf{x} in the log posterior refers to the vector $\mathbf{x} = [t_1, t_2, t_3, \rho_1, \rho_2, \rho_3]$, the translational and rotational parameters. Finally, m is the number of voxel locations for which a valid SDF value has been found for both subsegments. In our experiments, $\sigma = 2$.

Note that the logarithmic form of the posterior is suitable to non-linear least squares optimisation. Gradient-based maximisation of the above posterior to yield an optimal deformation is a highly non-linear optimisation problem. As such, it is suited to second-order gradient-based optimisation. To perform MAP (Maximum A Posteriori) over this posterior using an optimisation routine such as Levenberg-Marquardt, the following gradients must be computed for the rotational component of the deformation:

$$\frac{\partial E}{\partial \rho_n} = \frac{\partial E}{\partial \Psi} \frac{\partial \Psi}{\partial \Lambda} \frac{\partial \Lambda}{\partial \rho_n} \text{ for } n \in \{1, 2, 3\} \quad (9)$$

Similarly for the translational component:

$$\frac{\partial E}{\partial t_d} = \frac{\partial E}{\partial \Psi} \frac{\partial \Psi}{\partial \Lambda} \frac{\partial \Lambda}{\partial t_d} \text{ for } d \in \{x, y, z\} \quad (10)$$

where the gradient $\frac{\partial \Psi}{\partial \Lambda}$ is found via central finite differencing. For notational convenience we define $E \equiv \ln P(\Phi_s(x)|\Phi_{s'}(\Lambda(x)))P(L_{s,s'})$, see Equation 7

3.4. Implicit Surface Deformations

The overall object surface Φ is implicitly deformed by a blending function $\zeta(\Phi)$ over each of the subvolumes for which surface data has been integrated. As such the surface Φ is given by the following:

$$\Phi(\mathbf{x}) = \sum_{\chi \in X} \zeta(\Psi_\chi(\mathbf{x})) \quad (11)$$

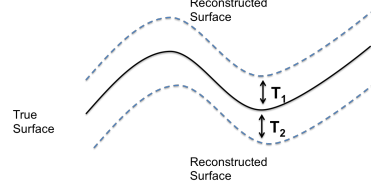


Figure 4: Measured surfaces from different subvolumes being deformed to the true surface.

where X is the set of subvolumes contributing to the surface Ψ . As such, the previously described transformation estimation process that deforms each of the subvolumes, implicitly optimises for the true surface as depicted in Figure 4. In our experiments $\zeta(\cdot)$ is defined as follows

$$\zeta(\psi_\chi) = \begin{cases} \psi_\chi & \text{if } P(\psi \in \Phi|\Omega, p) > P(\psi \notin \Phi|\Omega, p) \\ 0 & \text{if } P(\psi \in \Phi|\Omega, p) < P(\psi \notin \Phi|\Omega, p) \end{cases} \quad (12)$$

At this point the reader is referred back to Equation 1.

3.5. Volumetric Object Segmentation

The final stage in the proposed object reconstruction pipeline is the segmentation of the object voxels from those that have had measurements fused from the background. This segmentation is formulated within a CRF framework, where each node in the CRF represents a set of neighbouring voxels in space, with connections being made between adjacent neighbourhoods. The process of segmentation can be posed as an energy minimisation problem over a cut in downsampled voxel space, such that a segmentation in 3D is obtained. The following energy function consists of the unary posterior probabilities over appearance accumulated during the fusion process for a region in space and an additional pairwise smoothing term representing the physical appearance similarity of the object region represented by the voxel neighbourhoods γ and γ' :

$$E_n = \prod_{t=0}^{\infty} \prod_{\psi \in \Phi_n} P(\psi \in \Psi|\Omega_t, p_t) + P(\mathbb{E}[c]_\gamma | \mathbb{E}[c]_{\gamma'}) \quad (13)$$

where c represents the set of colour measurements fused in to the voxels within a given neighbourhood, for all N subvolumes and \mathbb{E} is the standard Expectation operator. The aforementioned cut in voxel space is achieved by an optimisation of the above equation within the Max-Flow framework [3].

3.6. Explicit Loop Closure Detection

In addition to our online model correction algorithm to counteract tracking drift, we also detect loop closure events such that re-localisation can occur and the tracked pose be adjusted accordingly. Following the approaches of [8, 11] we utilise a keyframe-based loop closure detection system.

For a given depth image, its corresponding keyframe is a subsampled and Gaussian filtered version of the depth image which is then encoded in a Fern Conservatory. Similarity/dissimilarity scores between Ferns are then computed and used to either detect a loop closure event for high scoring matches with existing keyframes, or to spawn a new keyframe if dissimilarity is sufficiently high w.r.t. the last keyframe.

3.7. Surface Fusion Procedure

Volumetric Representation The system we present follows the KinectFusion [16] pipeline for the integration of surface information. In such a formulation the scene or object is represented as the zero level of a level set embedding [5], where the level set is a field of distances to the surface. The surface itself is given by

$$S = \{\psi | \mathcal{D}(\psi) = 0\} \quad (14)$$

where S is the set of surface voxels, ψ is a voxel in the SDF(Signed Distance Function) volume and $\mathcal{D}(\cdot)$ is the SDF value. The SDF volume is truncated to yield a TSDF with truncation region μ .

Pose Estimation The pose estimation method used in this work is the ICP algorithm [1]. The estimation of the camera or object 6-DoF(Degree of Freedom) pose is formulated as a minimisation problem of the form:

$$\arg \min_{R,t} E = \sum_{\omega \in \Omega_d} \left((R\omega + t - \mathcal{V}(\bar{\omega})) \cdot \mathcal{N}(\bar{\omega}) \right)^2 \quad (15)$$

where Ω_d is a depth image, ω is a 3D point extracted from the depth image, R is an $\mathbb{SO}(3)$ rotation matrix, t is a translation vector, \mathcal{V} is a rendered depth map from the SDF model, \mathcal{N} is a rendered normal map from the SDF model and $\bar{\omega}$ is the point ω projected from the coordinate frame of \mathcal{V} and \mathcal{N} to the image plane. At this point it should be highlighted that the tracking algorithm used is separate from the contributions in this paper and as such can be substituted for any suitable pose estimation algorithm.

Surface Integration We utilise a weighted mean to fuse new depth measurements in to the TSDF model, as such, for a new depth measurement η projected to by voxel ψ , the following update to the TSDF volume \mathcal{D} is made

$$\mathcal{D}'(\psi) \leftarrow \frac{w(\psi)\mathcal{D}(\psi) + \min(1, \eta/\mu)}{w(\psi) + 1} \quad (16)$$

where $w(\cdot)$ is a weighting function and μ is the aforementioned TSDF truncation region.

Sequence Name	Our Approach ATE (m)	InfiniTAM ATE (m)
freiburg3_cabinet	0.077903	0.520693
freiburg3_teddy	0.030596	0.048560

Table 1: ATE (Absolute Trajectory Error) results (lower is better) achieved by our approach versus InfiniTAM with object segmentation.

4. Results

To evaluate our system, we perform quantitative experiments on camera pose estimation accuracy, qualitative and quantitative analyses on the obtained reconstructions. Firstly, the pose estimation accuracy is evaluated via an established SLAM evaluation benchmark [27]. We highlight that in traditional dense SLAM systems [21, 17, 16] – for which the benchmark is often employed – the entire contents of the visible scene are used for pose estimation, whereas in our system we rely only on points belonging to the object’s surface. Whilst more challenging, this implicitly allows us to track the camera w.r.t. the object regardless of which of the two is subject to motion. Then, qualitative comparisons are drawn between the reconstructions attained by our system, and those of the method described in [33]. We evaluate our system on multiple frame sequences depicting objects of different sizes. Finally, both qualitative and quantitative comparisons are drawn to an implementation [21] of the KinectFusion pipeline [16].

4.1. Pose Estimation Quality

In this section we present quantitative results of our systems’ robustness in estimating the camera motion, by performing tracking against the reconstruction of a single object, instead of the whole scene. The trajectories estimated by our system demonstrate low tracking drift. We perform such evaluation on two sequences of the RGB-D SLAM Dataset [27] depicting static objects observed by a moving camera. Tracking is performed using purely geometric cues, by matching the current depth frame with a rendering of the reconstructed object using ICP. We compare results against the system of [21] with object segmentation such that the comparison system also only tracks the object.

The tracking accuracy is evaluated via the ATE (Absolute Trajectory Error) metric [27], and is summarised in Table 1.

At this point, it should be highlighted that our proposed system is at a disadvantage when compared to dense SLAM systems that utilise the entire scene geometry for pose optimisation, since we track the sensor/object pose against a subset of the observed scene. Nevertheless, as shown by the results in Figure 5, our system is able to robustly estimate trajectories close to the ground truth whilst using only the objects’ geometric appearance. The cabinet reconstructed in the *freiburg3_cabinet* sequence is lacking in geometric features, as the object is mostly planar, and the small deficit

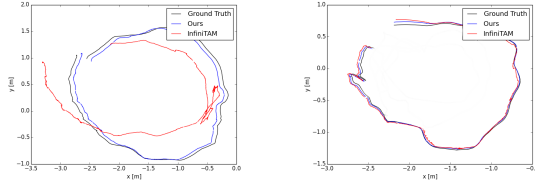


Figure 5: Comparison of the estimated camera trajectory with the ground truth for *freiburg3_cabinet* (left) and *freiburg3_teddy* (right).

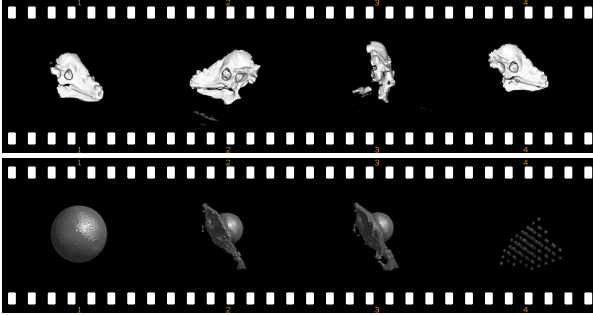


Figure 6: Quarterly interval snapshots of the Dinosaur Head reconstruction using our method (upper), and the one proposed by Ren et al. [33] (lower).

in tracking quality is mostly due to this factor. However, our system remains able to estimate a fairly accurate trajectory. In the *freiburg3_teddy* sequence we determine a trajectory very close to the ground truth. Improvement over the accuracy in *freiburg3_cabinet* is due to the wider availability of geometrical features, such as curves in the teddy’s body and head.

4.2. Qualitative Reconstruction Quality

In this section we present a qualitative comparison of our method against the approach by Ren et al. [33] in the reconstruction of closed object models. Each sequence is run through both systems; to evaluate the obtained results we regularly take snapshots of the reconstruction in the case of our system, and the level set evolutions, in the case of Ren et al. Such snapshots are captured after each quarter of a sequence has been processed.

As depicted in Figure 6, our method is able to successfully reconstruct the Dinosaur Head, whereas the approach by Ren et al. fails to converge towards a feasible shape. In addition, Figure 7 demonstrates that our system is able to generate consistent, closed models (unaffected by camera tracking drift) for a variety of sequences containing several loop closures. Failure of the competing method is also apparent for other sequences evaluated in this work, all presenting failure cases analogous to Figure 6. Another such example may be observed in Figure 8.

The object reconstructions depicted in Figure 9 have

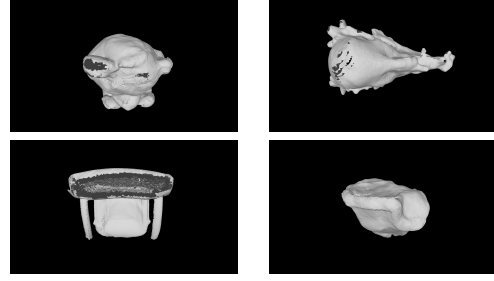


Figure 7: Closed reconstructions of a Teddy, a Dinosaur Head, a Chair and a Rock.

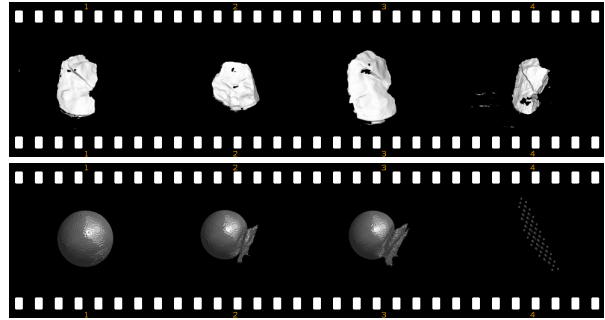


Figure 8: Quarterly interval snapshots of the Museum Rock reconstruction using our method (top row), and the one proposed by Ren et al. [33] (bottom row).

been obtained from sequences in which a camera was moved in a loop around each object in order to generate a closed model.

In addition, we provide a qualitative comparison between the reconstructions produced by the KinectFusion implementation of [21] when tracking the entire scene and manually segmenting the object out as a post processing step against our online segmentation and reconstruction system. This comparison is visible in Figure 9

Finally, we provide an example of a typical failure case when using the standard KinectFusion pipeline vs our approach. The gaps on the object surface in Figure 11 is caused by tracking drift, resulting in the system fusing surface information in to the wrong region of space, as can be seen, this is remedied by our approach.

4.3. Quantitative Reconstruction Quality

In this section we perform a quantitative evaluation of reconstruction quality of our method against an established dense SLAM system [21] following the KinectFusion [16] pipeline. The outputted reconstructions of our model are compared with the reconstruction of the dense SLAM system with the object of interest manually segmented out from the remainder of the scene. To quantify the reconstruction quality we employ the Hausdorff Distance [10] for subsets of metric spaces, where in our case the metric space is Eu-

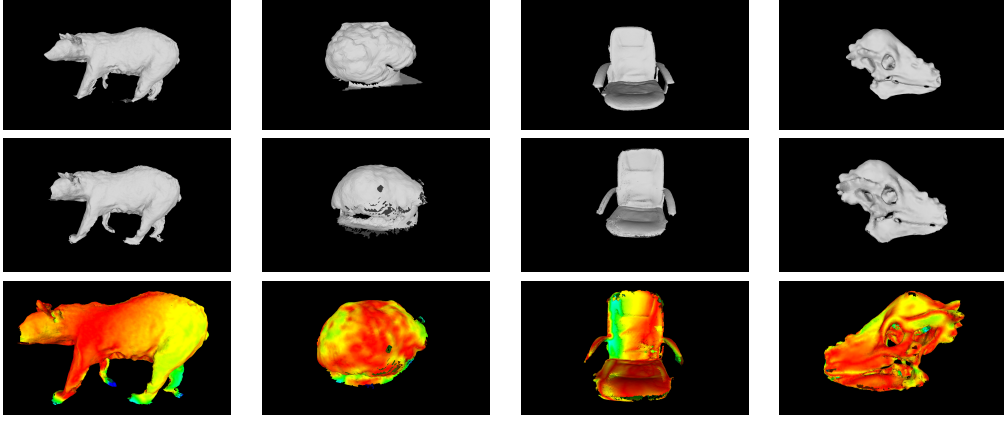


Figure 9: From left to right, Bear, Brain, Chair and Dinosaur Head. Row 1 is InfiniTAM output, row 2 is our system and row 3 is the Hausdorff distance between the two, with the colour scale given in Figure 10.



Figure 10: Hausdorff distance heatmap key. Blue is *Max Dist* and red is *Min Dist* of Table 2.

<i>Sequence</i>	<i>Min Dist</i>	<i>Max Dist</i>	<i>Mean Dist</i>	<i>RMS</i>
Bear	0	0.102777	0.013588	0.019796
Brain	0	0.026465	0.008745	0.011349
Chair	0	0.053441	0.012349	0.016422
Dinosaur Head	0	0.035252	0.007919	0.010676

Table 2: Hausdorff Distance measurements between ground truth mesh and our systems output.

geometrically close to those generated with a dense SLAM system [21] following the KinectFusion [16] pipeline that is modelling and tracking the entire scene.

4.4. Running Times and Performance

We have implemented our system both on the CPU and GPU. With a GPU implementation with NVIDIA CUDA we are able to achieve runtimes of on average *90Hz* with a consumer grade NVIDIA GeForce GTX1060 with 6GB GRAM. With our CPU only implementation we make use of parallelism with OpenMP and achieve runtimes of on average *5Hz* on a consumer grade PC with an Intel Core i5-6600K 3.5GHz CPU and 16GB of RAM. Such online runtimes are possible due to the asynchronous way in which we have implemented the online adjustments combined with GPU acceleration.

5. Conclusion

As has been demonstrated in this paper, our novel approach is effective in 3D object reconstruction. Our system is able to reconstruct closed object models on sequences for which an alternative, state of the art system [33] fails to converge to any reasonable solution. In addition, we show robust odometry on an established SLAM benchmark, despite the difficulty of tracking only the objects surface vs the entire scene. Finally, in spite of our use of ICP for pose estimation with ambiguities, our system is capable of robustly reconstructing a variety of objects that are problematic for existing approaches.

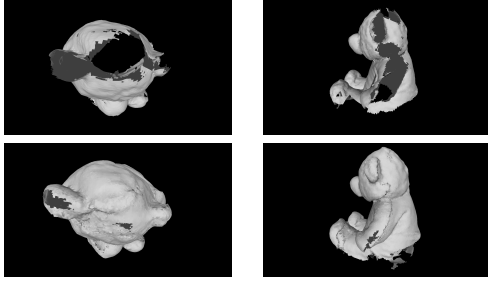


Figure 11: Teddy reconstruction with InfiniTAM (first row) versus our system (second row).

clidean. The Hausdorff Distance is defined as follows

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\} \quad (17)$$

where X is the ground truth dense SLAM reconstruction, Y is the reconstruction outputted by our system and $d(\cdot)$ is the Euclidean distance.

The resultant comparisons may be found in Table 2. In addition, we provide the outputted reconstructions from our system textured on reconstruction quality w.r.t. the Hausdorff Distance. For reference, the colour scale used is given in figure 10 where the left extrema (blue) is given by a sequences *Max Dist* and the right extrema (red) by the sequences *Min Dist*. As can be seen by the similarity measures presented, our system is capable of yielding reconstructions to a high quality despite the much more difficult tracking scenario of a single object rather than an entire scene. It can be seen that our output reconstructions are

References

- [1] P. J. Besl and N. D. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992.
- [2] C. Bibby and I. Reid. *Robust Real-Time Visual Tracking Using Pixel-Wise Posteriors*, pages 831–844. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sept 2004.
- [4] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt. Algorithms for 3d shape scanning with a depth camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(5):1039–1050, May 2013.
- [5] 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, 1996.
- [6] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using online surface re-integration. *arXiv preprint arXiv:1604.01093*, 2016.
- [7] N. Fioraio, J. Taylor, A. Fitzgibbon, L. D. Stefano, and S. Izadi. Large-scale and drift-free surface reconstruction using online subvolume registration. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4475–4483, June 2015.
- [8] B. Glocker, J. Shotton, A. Criminisi, and S. Izadi. Real-time rgb-d camera relocation via randomized ferns for keyframe encoding. *IEEE Trans. Vis. Comput. Graph.*, 21(5):571–583, 2015.
- [9] T. Gupta, D. Shin, N. Sivagnanasadan, and D. Hoiem. 3dfs: Deformable dense depth fusion and segmentation for object reconstruction from a handheld camera. *CoRR*, abs/1606.05002, 2016.
- [10] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, Sep 1993.
- [11] O. Kähler, V. A. Prisacariu, and D. W. Murray. *Real-Time Large-Scale Dense 3D Reconstruction with Loop Closure*, pages 500–516. Springer International Publishing, Cham, 2016.
- [12] K. Kolev, T. Brox, and D. Cremers. Robust variational segmentation of 3D objects from multiple views. In K. F. et al., editor, *Pattern Recognition (Proc. DAGM)*, volume 4174 of *LNCS*, pages 688–697, Berlin, Germany, Sept. 2006. Springer.
- [13] M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in-hand 3D object modeling. volume 30, pages 1311–1327, 2011.
- [14] R.-G. Mihalayi, K. Pathak, N. Vaskevicius, T. Fromm, and A. Birk. Robust 3D object modeling with a low-cost RGBD-sensor and AR-markers for applications with untrained end-users. 66:1–17, 2015.
- [15] K. S. Narayan, J. Sha, A. Singh, and P. Abbeel. Range Sensor and Silhouette Fusion for High-Quality 3D Scanning. pages 3617–3624, 2015.
- [16] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.
- [17] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D Reconstruction at Scale using Voxel Hashing. *ACM Transactions on Graphics*, 32(6):169, 2013.
- [18] K. Ohno, K. Kensuke, E. Takeuchi, L. Zhong, M. Tsubota, and S. Tadokoro. Unknown Object Modeling on the Basis of Vision and Pushing Manipulation. pages 1942–1948, 2011.
- [19] P. Panteleris, N. Kyriazis, and A. A. Argyros. 3d tracking of human hands in interaction with unknown objects. In X. Xie, M. W. Jones, and G. K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 123.1–123.12. BMVA Press, September 2015.
- [20] P. Panteleris, N. Kyriazis, and A. A. Argyros. Recovering 3d models of manipulated objects through 3d tracking of hand-object interaction. In *IEEE International Conference on Computer Vision Workshops (OUI 2015 - ICCVW 2015)*, Santiago, Chile, November 2015. IEEE.
- [21] V. A. Prisacariu, O. Kahler, M. M. Cheng, C. Y. Ren, J. Valentin, P. H. S. Torr, I. D. Reid, and D. W. Murray. A Framework for the Volumetric Integration of Depth Images. *ArXiv e-prints*, 2014.
- [22] C. Y. Ren, V. Prisacariu, O. Kaehler, I. Reid, and D. Murray. 3d tracking of multiple objects with identical appearance using rgb-d input. In *Proceedings of the 2014 2Nd International Conference on 3D Vision - Volume 01, 3DV '14*, pages 47–54, Washington, DC, USA, 2014. IEEE Computer Society.
- [23] H. Roth and M. Vona. Moving Volume KinectFusion. In *British Machine Vision Conference*, pages 1–11, 2012.
- [24] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.
- [25] M. D. Shuster. Survey of attitude representations. *Journal of the Astronautical Sciences*, 41:439–517, Oct. 1993.
- [26] M. Slavcheva, W. Kehl, N. Navab, and S. Ilic. SDF-2-SDF: Highly Accurate 3D Object Reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016.
- [27] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.
- [28] D. Tzionas and J. Gall. 3D Object Reconstruction from Hand-Object Interactions. pages 729–737, 2015.
- [29] T. Weise, T. Wismer, B. Leibe, and L. V. Gool. In-hand scanning with online loop closure. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1630–1637, Sept 2009.
- [30] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald. Kintinuous: Spatially Extended KinectFusion. Technical Report MIT-CSAIL-TR-2012-020, MIT, 2012.

- [31] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald. Real-time large scale dense RGB-D SLAM with volumetric fusion. *34(4-5):598–626*, 2015.
- [32] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. ElasticFusion: Dense SLAM Without A Pose Graph. 2015.
- [33] C. Yuheng Ren, V. Prisacariu, D. Murray, and I. Reid. Star3d: Simultaneous tracking and reconstruction of 3d objects using rgb-d data. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.
- [34] M. Zeng, F. Zhao, J. Zheng, and X. Liu. A Memory-Efficient KinectFusion Using Octree. In *Computational Visual Media*, pages 234–241. Springer Berlin Heidelberg, 2012.
- [35] Q.-Y. Zhou and V. Koltun. Dense Scene Reconstruction with Points of Interest. *ACM Transactions on Graphics*, 32(4):112, 2013.