Thesis Title

Department of Engineering Science

University of Oxford

Mr Jack Miles Hunt

Declaration

Acknowledgements

Abstract

Notation and Abbreviations Used

Contents

Chapter 1

Introduction

Chapter 2

Literature Review

2.1 Tracking and Mapping

Besl and McKay [1]

- 3D Shape Registration.
- Full 6DoF pose estimation.
- Requires shape complexity - geometrically distinctive.
- Compute closest point, compute & apply registration, iterate until MSE low.
- Convergence not guaranteed, determined empirically to be rapid over first few iterations.

Curless and Levoy [2]

- Early volumetric reconstruction work.
- Integrates aligned range images in to a volume.
- Volume introduced is SDF - cumulative and weighted.
- Isosurface extracted from SDF - Marching Cubes.
- Gaps filled by tesselation.

Zhou et.al. [3]

- Alternative representation to Curless and Levoy.
- Massively parallel KD-Tree construction.
- Suitable for real-time use.
- Parallelism achieved by building tree with BFS.
- Example use given for Ray Tracing and Photon Mapping.

Censi [4]

- PL-ICP, ICP using point-to-line metric.
- Closed form solution in planar case.
- Quadratic convergence in finite amount of steps.
- New formulation weighted by normal and solved by reducing to quadratic form and introducing Lagrange Multipliers.
- Prior to optimisation, trimming procedure used to remove outliers.

Newcombe et.al. [5]

- Real time mapping of indoor scenes with Kinect sensor.
- Invariance to lighting.
- Observations fused in to SDF like volume of Curless et al - TSDF
- Multilevel ICP(coarse to fine) used to obtain pose.
- Measurement-¿integration-¿isosurface extraction-¿pose update.
- Limited to static scenes.

Neissner et.al. [6]

- Pipeline follows that of KinectFusion.
- Introduce a spatial Hashing data structure.

- TSDF split in to Voxel Blocks which are hashed.
- Low space and time complexity.
- Streaming system to reduce GPU memory usage.

Thomas et.al. [7]

- Represent scene as a set of planes with attributes.
- Motivated by planar nature of objects such as tables and cabinets.
- Attributes are bump(normal) image for geometry encoding, mask image encoding confidence and RGB image.
- Rendering by quadrangulation.
- Tracking as with KF - linearized GICP.

Salas-Moreno et.al. [8]

- Introduce a new "Object Orientated" Dense SLAM paradigm.
- Incorporates prior knowledge that many scenes have repeated structure.
- Scene split in to graph of objects. Pose graph optimisation used.
- ICP run against object renderings, followed by detection and insertion of objects. Finally graph optimisation.
- Graph based re-localisation.
- Requires database of known objects.

Stuckler et.al. [9]

- Uses multiple resolution, probabilistic surfel maps as representation. [10]
- Octree with spatial and appearance statistics at each level.
- Randomised loop closure - graph and key-view based.

- Pose estimation by maximising observation likelihood with uncertainty measure.

Salas-Moreno et.al. [11]

- Exploits planar structure in the scene, like Thomas.
- Focus on detection and modelling of planes, refined over time.
- From generated Surfel maps, planes are segmented and holes filled over time. [10]
- Fern encoding used for relocalisation.
- ICP between measured vertex map and predicted vertex map.

Prisacariu et.al [12] Followed up by tech report [13]

- Open source implementation of Voxel Hashing.
- A number of optimisations to the data structure(allocation & integration) and raycaster.
- IMU data to supplement camera observations for tracking.
- 47Hz NVIDIA Shield tablet, 910Hz Titan X.

Whelan et.al [14]

- Like KF but large scale(hundreds of meters), achieved using GPU cyclic buffers.
- Geometric and photometric camera pose constraints.
- Map updated by frame recognition, as-rigid-as-possible.
- Pose graph based loop closure.

Zhou et.al. [15]

- Uses contour cues to improve camera tracking.
- Contour cues extracted from noisy and incomplete depth images.
- Correspondence constraints using scene geometry enforced on pose estimation.

- Based on KF pipeline.
- Depth image inpainting used before contour extraction.

Kahler et.al [16]

- Based on KF pipeline and InfiniTAM.
- Online submap alignment algorithm for drift correction.
- Inter-submap corrections based on graph optimisation.
- Loop closure detected using fern conservatories.

2.2 Semantic SLAM

Civera et.al. [17]

- Monocular EKF based SLAM.
- Semantics added to points via SURF correspondences with precomputed object descriptors.
- Geometric compatibility is then tested.

Stuckler et.al [18]

- Uses RGB-D, not monocular.
- Fuses only detected objects.
- Object detection via random forests.
- Hand crafted features over object regions.

Valentin et.al. [19] Golodetz et.al. [20].

- Online, real time semantic segmentation using user input.
- Dense reconstruction like KF.

- User physically interacts.
- Voxel Oriented Patch features using normals and appearance in CIELab.
- Streaming Random Forests[21] and Valentin version uses Mean Field [22] optimised by [23].

Bengio et.al. [24] - representation learning. Girshick et.al [25] - feature hierarchies. Handa et.al. [26]

- Real time reconstruction with semantic segmentation.
- Deep autoencoders stacked and trained on synthetic depth data.
- Uses depth only cues.
- KF style reconstruction.

Cavallari et.al. [27]

- Built on top of Voxel Hashing
- RGB frames passed to FCN[28], PMF's out.
- Post rendering, colours determined by argmax over PMF bins.
- Texture(non label colours) trilinearly interpolated.

2.3 Dynamic SLAM, Motion Segmentation and Optical Flow

Tsap et.al. [29]

- Algorithm for nonrigid motion tracking.
- Solve for dense motion vector fields between 3D objects via Finite Element Methods.
- Iteratively analyses difference between actual and predicted behaviour.

- Iterative descent to find optimal parameters of nonlinear FEM.
- Tracking improved by using point correspondences.
- Not scene based.

Chen et.al [30]

- Nonrigid motion tracking applied to human body.
- Surface mesh extracted from multi view video and skinned.
- Hierarchical(w.r.t articulation) Weighted ICP is then applied.
- ICP points weighted by Approximate Nearest Neighbour
- Prior human skeleton fitted.
- Not scene based.

Sun et.al. [31]

- Layered optical flow. Layered over detected moving objects.
- Depth ordered MRF's and Max Flow used for layering.
- Number of layers automatically determined.
- Max Flow used to solve for discretised flow field cost function.
- Motion tracking only, no reconstruction.

Unger et.al. [32]

- Variational formulation for motion estimation and segmentation with occlusion handling.
- Parametric labelling of flow field for each object undergoing motion.
- Labels encoded with an MRF Potts model as with Sun.
- Flow and labels solved for via Primal-Dual optimisation framework.

- Again, motion only. No reconstruction.

Herbst et.al. [33]

- Extension of Optical Flow to 3D scenes.
- Scene flow computed from RGB-D data, i.e. Kinect.
- Approach based on Brox et.al. [34]
- Variational formulation and optimisation framework.

Stuckler et.al. [35]

- EM framework to segment rigid body motion from RGBD data.
- Robust to simultaneous fg and bg motion by treating both with parity.
- Sites of motion as well as motion params are posed as latent variables.
- No reconstruction, but feasibly could be extended.

Keller et.al. [36]

- Dense SLAM system capable of handling scene dynamics.
- Works with RGB-D data.
- Uses ICP outliers to determine dynamic scene components followed by flood fill.
- Surfel representation [10]
- Flat data structure. Does not have advantages of Voxel Hashing.

Perera et.al. [37]

- Motion segmentation in TSDF volumes,
- Based on MAP inference over a CRF on the TSDF.
- Can handle minor and major displacements.

- Motion labels and parameters found w.r.t. live frame and TSDF.
- 256^3 voxel grid does not run real time. Scalability issues.

Newcombe et.al. [38]

- Handles non-rigidly deforming scenes.
- Built on KinectFusion pipeline.
- 6DoF motion field estimated that warps model to live frame.
- Warp field used to fuse new measurements in to canonical model.
- Warp field solved for by Dual Quaternion Blending. [39]
- Limitations, such as lack of robustness to open/closed topology changes(hands). Authors highlight scalability issues.

2.4 Object Reconstruction

Curless and Levoy [2]

- Object reconstruction from different viewpoints but statically.
- No pose tracking in the TAM sense.

Kolev et.al. [40]

- Probabilistic 3D segmentation.
- Rather than direct reconstruction like Curless, the most probable image w.r.t. images is inferred.
- Level set representation used.
- Level set has analogous probability volume(pF,pB)
- Level set is evolved via a variational framework over probability volume.

- Only evaluated on synthetic data.
- Designed for objects with distinct appearance and homogeneous background, though some tolerance to noise is claimed.

Weise et.al. [41]

- 3D in hand scanning system.
- Point cloud representation, Surfel rendering. [10]
- Objects rotated in front of a sensor, suggests limited tracking ability.
- Drift offset in as-rigid-as-possible manner.
- ICP registration. Topology graph used for deformation.
- Uses range scanner. No indication of RGBD capability. May need high quality equipment.

Ren et.al. [42]

- Probabilistic framework for tracking and reconstruction of objects.
- Initialised with a shape prior level set, which is then evolved as per observations.
- Works with RGBD data.
- Pixel-Wise-Posteriors used for segmentation. Prob volume like Kolev. [43]
- Experiments show limitations.

Dou et.al. [44]

- Reconstruction of deformable objects with a Kinect sensor.
- Loop closures automatically detected, distributing drift error over the loop.
- Latent shape and nonrigid deformation solved for with bundle adjustment.
- Surface represented as a triangular mesh.
- Experiments show high quality reconstructions, but with overnight run times.

Gupta et.al. [45]

- 3D reconstruction and segmentation of objects with an RGBD sensor.
- Implicit representation, i.e. volumetric. Fusion with Softmax rather than weighted average like KF.
- Each voxel gets a labelobj, bg, empty. Segmentation with graph cut and alpha expansion.
- Keyframe based loop closure.
- Pose estimation via photometric loss. Authors report drift to be a problem.
- Authors report difficulty in building a granular reconstruction.

2.5 Shape and Pose Prediction

Prisacariu et.al. [46]

- Shape prediction, segmentation(optimisation based) and pose optimisation.
- Hierarchical(early deep?) GPLVM's for Shape Latent Space Embedding. [47]
- Unified energy function.
- Candidate shapes generated as one off regression in latent space.

Dame et.al. [48]

- Dense object reconstruction from monocular image source.
- Shape priors used to aid reconstruction and segmentation, GPLVM.
- Depth maps optimised for with Primal Dual with TV. Poses are known from PTAM.

Toshev et.al. [49]

- Human pose estimation(articulated estimation) with Deep Neural Networks.
- Cascaded DNN regressors.

- No reconstruction, pose estimation only.

Wohlhart et.al. [50]

- 3D object detection and pose recovery.
- CNN descriptors used with Nearest Neighbour Cost for detection and rough pose.
- Problem posed as KNN search in Descriptor Space.
- Object and Pose are coupled in training(i.e. two similar cars with different poses have distant descriptors)

Chang et.al. [51]

- Large scale dataset of 3D shapes.
- Synthetic data with no "source" sequence, i.e. no depth.
- Can be used to learn rich latent space embeddings.

Rock et.al. [52]

- Recovers a complete 3D model from a depth image of an object.
- Input depth image matched to database of objects via a Random Forest.
- Matched shape coarsely matched to depth map, then deformed at a higher granularity.
- Deformation manually optimised for.

Zhou et.al. [**?**]

- Object detection from 3D Point Clouds.
- End-to-end trainable Convolutional, Region Proposal Network.
- Trained on KITTI LIDAR dataset. [53]
- Voxelisation of point cloud used for region detections.

Gwak et.al [54]

- GAN like shape prediction with log-barrier ojective.
- Weakly supervised with sillhouettes and 3D shapes.
- May not work "in the wild"

Chapter 3

Real Time Motion Segmentation for Dense Volumetric Fusion

3.1 Introduction

Progress in Dense Volumetric Fusion has been accelerated in recent years with the availability of comsumer grade RGBD sensors such as the Microsoft Kinect and the Asus Xtion coupled with the increasingly parallel nature of GPU hardware. Systems such as the seminal KinectFusion [5] allow one to build high quality, globally consistent scene models trivially. Applications of such reconstruction pipelines however are limited due to the inability of such systems to handle scenes in which there are dynamics; such systems are unable to yield reliable reconstructions when there is motion in the sensors field of view independent of the sensors own motion. Such a scenario introduces additional error to the Pose Estimation component in the pipeline and as such causes model corruption.

In this chapter an approach to solving the problem of performing robust Dense Volumetric Reconstruction in dynamic scenes is presented. Central to the pipeline is the introduction of a dual scene representation based on the use of an implicit TSDF [2]. The use of a dual TSDF approach allows for the segmentation of moving components in the scene from static components, e.g. segmenting a person getting up from a chair from the chair itself. One of the two scene representations is the "static" scene and the other the "dynamic" scene. Such

separation prevents corruption in the static scene, the reconstruction output of the system.

Without the segmentation of dynamic scene components, when tracking against the current reconstruction the integrated dynamic components may cause artifacts that prevent the finding of ICP correspondences which often causes camera tracking to drift, or completely fail. By tracking against only the stable scene components, this inteference in the ICP pose estimation process can be mitigated.

Once a part of the dynamic model has been stable for a sufficient period, it's volumetric data is integrated in to the static model and it is used for the tracking phase of the pipeline. The use of Volumetric structures in this work is motivated by previous works on Voxel Block Hashing [6], providing efficient, real time lookup operations. The presented approach exploits the abstraction that Voxel Blocks provide; a block of voxels is interpreted as a region of space that can be either static or dynamic. Voxel Block Stability is determined by a confidence measure over the Voxel Blocks in the Dynamic Scene, such that Isosurface information is not transferred to the Static Model(used for Camera Tracking) until there is sufficient confidence in it's stability. From a survey of the literature, it appears that this approach is the first to utilise such a dual representation for the motion segmentation problem.

The remainder of this chapter is structured as follows. Section 3.2 presents an assessment of related and relevant literature. Section 3.3 introduces preliminaries pertaining to the static Fusion pipeline followed by Section 3.4 describing the dynamic Fusion component of the pipeline. Qualitative and Quantitative results are presented in Sections 3.5 and 3.6, respectively. Finally, an application to interactive object recognition is given in Section 3.7.

3.2 Related Work

3.3 Static Volumetric Fusion

The Volumetric Fusion approach taken in this work draws on previous Volumetric integration techniques [2, 5, 6, 12] and shall be introduced in this section as preliminary material and shall be referred to in later chapters. Following this approach, at each frame the camera is tracked against the current scene, after which new data is fused into the scene model which is then

rendered using Raycasting to prepare for tracking in the next frame.

The static Fusion pipeline consists of the following three consecutive steps:

- Camera Tracking.
- Model Integration.
- Rendering.

The approach in this work utilises the TSDF Volumetric data structure which encodes for each voxel in the structure, a signed value and a weight. In the case of 3D environment modelling, the values pertain to distances from surfaces, within some truncation region.

Given a vertex map generated from the back projection of the points in a depth image, the vertices pertain to the zero crossing point with Voxels either side representing distances to the zero crossing point; positive in front of the surface, negative behind. Surface points are known as the Zero Level Set. For a given TSDF $\mathbf{\Phi}$, the Zero Level Set is defined as follows

$$\mathcal{S} = \{v \mid \mathbf{\Phi}(v) = 0\}, \forall v \in \mathbf{\Phi} \tag{3.1}$$

where v denotes a TSDF Voxel.

To facilitate real time fusion, the InfiniTAM framework employs a Voxel Block Hashing mechanism for fast access to scene Voxels [6]. Within this context, Voxel Blocks are collections of $N \times N \times N$ TSDF Voxels, stored in a Hash Table for fast access. As such, each Hash Table entry corresponds to a portion of a global Voxel Block Array, pertaining to a region in the scene. This division of the scene into Voxel Blocks is used for the later process of determining and labelling dynamic regions in the scene.

3.3.1 Camera Tracking

As in previous works [5, 12], the gradient optimisation based ICP algorithm is utilised to register consecutive images to derive the camera pose at time t with respect to time $t-1$, that is to optimise for the Rigid Body Transform $\mathbf{T} \in \mathbb{SE}(3)$ of the camera between the two frames using the Levenberg-Marquardt Nonlinear Least Squares method [55]. The rendering stage of the

pipeline is used to generate the image from the TSDF at time $t-1$ to which a new frame at time t is registered.

The target Transformation $\mathbf{T} \in \mathbb{SE}(3)$ is a member of the Special Euclidean Group

$$\mathbb{SE}(3) = \{\mathbf{R}, \mathbf{t} \mid \mathbf{R} \in \mathbb{SO}(3), \mathbf{t} \in \mathbb{R}^3\} \tag{3.2}$$

and has the following form

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.3}$$

where $\mathbb{SO}(3)$ is the Special Orthogonal Group of Skew Symmetric Rotation Matrices.

Attitude Representation

The Rotation Matrix component of the Transformation $\mathbf{T}$ is Generated by a Rodriguez Paramaterization[56] whereby the $\mathbb{SO}(3)$ Rotation Matrix $\mathbf{R}$ is generated by three rotational parameters, α, β and γ. Each parameter represents a rotation around one of three Principal Axes.

The formulation of the Rodriguez Paramaterization is given by the following [56], with the parameter Vector $\mathbf{p} = [\alpha, \beta, \gamma]^T$

$$\mathbf{R}(\mathbf{p}) = \frac{1}{\|\mathbf{p}\|_2^2}\left[(1 - \|\mathbf{p}\|_2^2)\mathbf{I} + 2\mathbf{p}\mathbf{p}^T + \omega(\mathbf{p})\right] \tag{3.4}$$

where for an arbitrary Vector $\mathbf{v}$, $\omega(\mathbf{v})$ is defined as the Cross Product Matrix operator and is defined as follows

$$\omega(\mathbf{v}) = \begin{bmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{bmatrix} \tag{3.5}$$

Evaluating Equation 3.4 leads to the following form of the Rotation Matrix $\mathbf{R}$

$$\mathbf{R} = \frac{1}{\|\mathbf{p}\|_2^2 + 1}\begin{bmatrix} \alpha^2 - \beta^2 - \gamma^2 + 1 & 2\alpha\beta + \gamma & 2\alpha\gamma - \beta \\ 2\alpha\beta - \gamma & -(\alpha^2 - \beta^2 + \gamma^2 - 1) & \alpha + 2\beta\gamma \\ 2\alpha\gamma + \beta & -(\alpha - 2\beta\gamma) & -(\alpha^2 + \beta^2 - \gamma^2 - 1) \end{bmatrix} \tag{3.6}$$

The Translational component $\mathbf{t}$ of the Transformation $\mathbf{T}$ is given by the following Vector, with each component representing a translation along it's respective axis.

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{3.7}$$

Pose Recovery Formulation

The recovery of the camera pose change between frames t and $t-1$ may be formulated as the following Energy Minimisation problem

$$E(\mathbf{R}, \mathbf{t}, \mathbf{\Omega}, \mathbf{\Phi}) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{p \in \mathbf{\Omega}} \left\| [\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{V}(\bar{\mathbf{x}})]^T \mathcal{N}(\bar{\mathbf{x}}) \right\|_2 \tag{3.8}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the aforementioned Rotation Matrix and Translation Vector of the transformation $\mathbf{T}$. $\mathbf{x}$ is the 3D point extracted from the depth image $\mathbf{\Omega}$ and the point $\bar{\mathbf{x}}$ is the 3D point in the TSDF Volume $\mathbf{\Phi}$ found by Raycasting from $\mathbf{\Omega}$ under the Transformation $\mathbf{T}$. Finally, $\mathcal{N}$ is a Normal map of $\mathbf{\Phi}$ and is defined as follows

$$\mathcal{N} = \frac{\nabla \mathbf{\Phi}}{\|\nabla \mathbf{\Phi}\|_2} \tag{3.9}$$

where $\nabla \mathbf{\Phi}$ is approximated with Central Finite Differencing.

For the Gradient update phase of the ICP algorithm, the Partial Derivatives $\frac{\partial E}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}}$ may be derived as follows in Equation **??**. As a first step, the following definition is made

$$\phi(\mathbf{R}, \mathbf{t}, \mathbf{x}, \bar{\mathbf{x}}) = [\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{V}(\bar{\mathbf{x}})]^T \mathcal{N}(\bar{\mathbf{x}}) \tag{3.10}$$

With this substitution in place, the derivation proceeds as follows

$$\begin{aligned}
\frac{\partial E}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} &= \frac{\partial}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} \sum_{p\in\mathbf{\Omega}} \|\phi(.)\|_2 \\
&= \sum_{p\in\mathbf{\Omega}} \frac{\partial}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} \|\phi(.)\|_2 \\
&= \sum_{p\in\mathbf{\Omega}} \frac{\partial}{\partial \phi(.)} \|\phi(.)\|_2 \frac{\partial \phi(.)}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} \\
&= \sum_{p\in\mathbf{\Omega}} \frac{1}{2} \frac{2\phi(.)}{\sqrt{\phi(.)^T\phi(.)}} \frac{\partial \phi(.)}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} \\
&= \sum_{p\in\mathbf{\Omega}} \frac{\phi(.)}{\|\phi(.)\|_2} \frac{\partial \phi(.)}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} \\
&= \sum_{p\in\mathbf{\Omega}} \frac{\phi(.)}{\|\phi(.)\|_2} \left[\frac{\partial}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} \left[\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{V}(\bar{\mathbf{x}})\right]^T \mathcal{N}(\bar{\mathbf{x}}) + \left[\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{V}(\bar{\mathbf{x}})\right]^T \frac{\partial}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}} \mathcal{N}(\bar{\mathbf{x}}) \right] \\
&= \sum_{p\in\mathbf{\Omega}} \frac{\phi(.)}{\|\phi(.)\|_2} \left[\frac{\partial \mathbf{R}}{\partial\{\alpha,\beta,\gamma\}} \mathbf{x} \right]^T \mathcal{N}(\bar{\mathbf{x}})
\end{aligned} \tag{3.11}$$

The full Partial Derivatives $\frac{\partial \mathbf{R}}{\partial \alpha}$, $\frac{\partial \mathbf{R}}{\partial \beta}$ and $\frac{\partial \mathbf{R}}{\partial \gamma}$ may be found Equations 6, 7 and 8 of Appendix .1.

The Partial Derivatives $\frac{\partial \mathbf{R}}{\partial\{\alpha,\beta,\gamma\}}$ may be multiplied with $\mathbf{x}$ and combined in to the following Rotation Jacobian

$$\begin{aligned}
\mathbf{J}_R &= \left[\frac{\partial \mathbf{R}}{\partial\{\alpha,\beta,\gamma\}} \mathbf{x} \right]^T \Bigg|_{\alpha,\beta,\gamma=0} \\
&= \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}
\end{aligned} \tag{3.12}$$

Note that the derivation for the Partial Derivatives $\frac{\partial E}{\partial \mathbf{t}_{\{x,y,z\}}}$ is analogous with that of Equation 3.11, with the result given as follows

$$\frac{\partial E}{\partial \mathbf{t}_{\{x,y,z\}}} = \sum_{p\in\mathbf{\Omega}} \frac{\phi(.)}{\|\phi(.)\|_2} \left[\frac{\partial \mathbf{t}}{\partial\{x,y,z\}} \right]^T \mathcal{N}(\bar{\mathbf{x}}) \tag{3.13}$$

The Translation Partial Derivatives $\frac{\partial \mathbf{t}}{\partial\{x,y,z\}}$ may also be combined in to the following Trans-

lation Jacobian

$$\mathbf{J}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

The overall combined Jacobian for the Energy function defined in Equation 3.8 as follows

$$\mathbf{J} = \frac{\phi(.)}{\|\phi(.)\|_2} \begin{bmatrix} \mathbf{J}_R & \mathbf{J}_t \end{bmatrix}^T \mathcal{N}(\bar{\mathbf{x}}) \tag{3.15}$$

Pose Recovery Optimisation

With the gradient derivations in place, this section will now detail the Pose Recovery procedure. As highlighted, the optimisation routine used is the Levenberg-Marquardt[55] algorithm for solving Nonlinear Least Squares problems. The Gradient update equation for the Levenberg-Marquardt algorithm is given as follows

$$\theta_{t+1} = \theta_t - (\mathbf{H} + \lambda \text{diag}(\mathbf{H}))^{-1}\mathbf{J} \tag{3.16}$$

where $\theta_t = [\alpha, \beta, \gamma, t_x, t_y, t_z]^T$ is the Parameter Vector of $\mathbf{T}$ at time t, $\mathbf{J}$ is the Jacobian introduced in Equation 3.15 and $\mathbf{H}$ is the Hessian, approximated by $\mathbf{H} = \mathbf{J}^T\mathbf{J}$. The parameter λ controls the influence of the Gradient on the update step and is adjusted according to the change in error, as shall be evident in the Algorithm that follows.

Algorithm 1 ICP with Levenberg-Marquardt

procedure ICP($\mathcal{D}_l, \mathcal{D}_m, \mathcal{V}, \theta_{t-1}$)
  $\lambda \leftarrow \lambda_{init}$
  $\theta_{tmp} \leftarrow \theta_{t-1}$
  $\theta_t \leftarrow \theta_{tmp}$
  $\epsilon_{old} \leftarrow \inf$
  $\epsilon \leftarrow \epsilon_{old}$
  while $\epsilon >= \tau$ **do**
    $\epsilon \leftarrow E(.)$ ▷ Evaluate Equation 3.8
    if $\epsilon \leq \epsilon_{old}$ **then**
      $\lambda \leftarrow 10\lambda$
      $\theta_t \leftarrow \theta_{tmp}$
    else
      $\lambda \leftarrow \frac{\lambda}{10}$
      $\epsilon_{old} \leftarrow \epsilon$
      $\theta_{tmp} \leftarrow \theta_t$
    end if
    $\mathbf{J} \leftarrow \nabla E$ ▷ Evaluate Equation 3.15
    $\mathbf{H} = \mathbf{J}^T\mathbf{J}$
    $\mathbf{C} = \text{chol}(\mathbf{H} + \lambda\text{diag}(\mathbf{H}))$
    $\delta = \text{backsub}(\mathbf{C}, \mathbf{J})$
    $\theta_{tmp} \leftarrow \theta_{tmp} - \delta$
  end while
return θ_t
end procedure

3.3.2 Volumetric Integration

The second phase in the Scene Reconstruction pipeline is Volumetric Integration. That is, the integration of observed depth images in to a consistent, Implicit Volumetric representation, in this case the existing TSDF model of the scene, providing the basis for an updated rendering to be used in the ICP procedure for Tracking at the next time step.

As previously outlined, the TSDF may be defined as a volume of distances to an Isosurface, with the Isosurface itself being given by the Zero Level Set, as defined in Equation 3.1. A graphical representation is given in Figure 3.3.2.

TODO: Make figure.

Figure 3.1: An example of a Two Dimensional Signed Distance Function.

As with KinectFusion [5] the global (scene) location $\mathbf{x}_v$ of each voxel $v \in \mathbf{\Phi}$ that is visible in

the current view frustum is transformed into the camera's coordinate frame via the following Transformation(noting that $\mathbf{x}_v$ is in Homogeneous form)

$$\mathbf{x}_\Omega = \mathbf{K}\mathbf{T}_i^{-1}\mathbf{x}_v \tag{3.17}$$

where $\mathbf{K}$ is the Camera's Intrinsic Calibration Matrix, $\mathbf{T}$ is the Transformation optimised for at time t, with the form given in Equation 3.3 and $\mathbf{x}_\Omega$ is the resultant projected coordinates.

The Integration of new data points into the TSDF Volume is achieved by computing running averages; each voxel contains a running average of its SDF value over time.

Projecting to the depth image $\mathbf{\Omega}$ coordinates as in Equation 3.17 to perform a depth lookup in $\mathbf{\Omega}$ and subtracting the z component of $\mathbf{x}_v$ from the resulting value yields the depth offset from the surface, as follows

$$\eta = \mathbf{x}_\Omega - \mathbf{x}_v^z \tag{3.18}$$

If $\eta \geq -\mu$, that is that the depth of the point is not beyond the truncation band behind the Isosurface(Zero Level Set), where μ is half the width of the truncation band, then the TSDF depth measurement update proceeds as follows for a Voxel $\mathbf{x}_v \in \mathbf{\Phi}$.

$$\mathbf{\Phi}(\mathbf{x}_v)_t = \frac{1}{\phi(\mathbf{x}_v)_{t-1} + 1}\left[\phi(\mathbf{x}_v)_{t-1}\mathbf{\Phi}(\mathbf{x}_v)_{t-1} + \min\left(1, \frac{\eta}{\mu}\right)\right] \tag{3.19}$$

In addition, the weight $\phi(\mathbf{x}_v)$ is updated as follows.

$$\phi(\mathbf{x}_v)_t = \phi(\mathbf{x}_v)_{t-1} + 1 \tag{3.20}$$

3.3.3 Rendering

Following the Integration process outlined in Section 3.3.2 is the rendering stage in which an image of the scene image is generated to provide an updated view for the tracking stage outlined in Section 3.3.1, at the next time step.

Rendering in the pipeline is achieved by Raycasting, the process of 'casting' a ray from the Camera Frame into the scene, to find intersections with the scene Isosurface.

3.4 Volumetric Fusion with Dynamic Scenes

To adapt the conventional approach described in Section 3.3 to handle dynamic environments, a dual-volume representation of the scene is introduced, consisting of a *Static Model* and a *Dynamic Model* (both of which are TSDF's).

There are two additional stages in the dynamic pipeline to handle integration in to the Static Model. The first updates stability values for all of the Voxel Blocks in the current view frustum at each frame. The second Integrates blocks whose stability values are above a certain threshold into the static model. Camera Tracking is performed against the Static Model as soon as it contains valid Isosurface data to track against, preventing moving objects in the scene from contributing to tracking drift.

An overview of the Pipeline is given in Figure 3.4

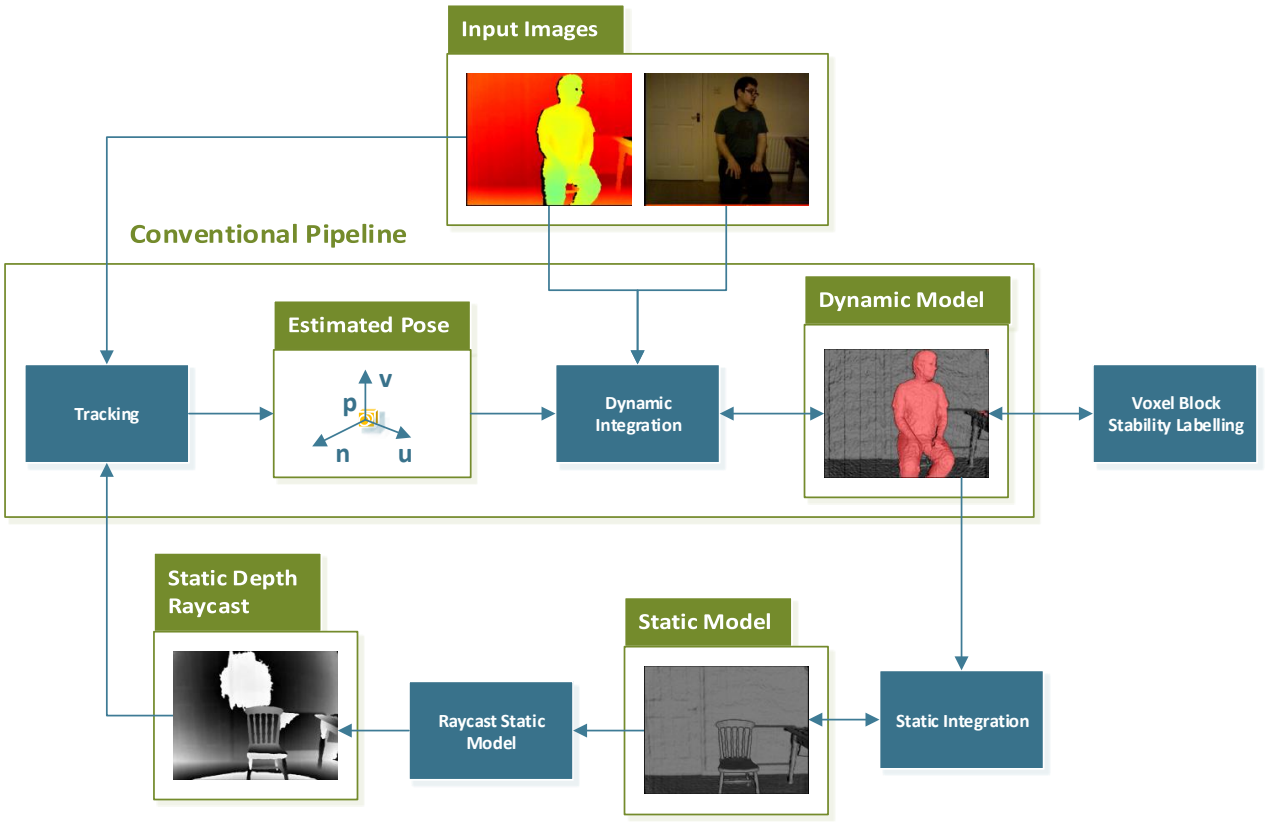


Figure 3.2: The Motion Segmentation pipeline. Note the additional Voxel Block Labelling stage with a feedback to the Model Integration stage.

3.4.1 Stability Labelling

The purpose of the stability labelling section of the pipeline is to distinguish between Stable and Unstable Voxel Blocks in the Dynamic Model, resulting in parts of the scene that are moving being excluded from the Static Model.

For each Voxel Block in the Dynamic Model, a *Stability Value* is maintained, representing the extent to which the instantaneous TSDF values for the Voxels in a given Voxel Block(visible in the current View Frustrum under the current Pose) have remained similar to the existing TSDF values for those voxels over time.

The stability value for each voxel block in the scene is initialised to nought. At each frame, the instantaneous TSDF values for the Voxels in each visible Voxel Block are computed. For each Voxel Block, the Mean Absolute Difference τ, between the instantaneous and existing TSDF values is computed as follows.

$$\tau_{\mathcal{V}} = \left[\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \left| \mathbf{\Phi}^d(v) - \min\left(1, \frac{\eta}{\mu}\right) \right| \right] \tag{3.21}$$

The Stability Label $l_{\mathcal{V}} \in \{\text{stable}, \text{unstable}\}$ for a given Voxel Block $\mathcal{V}$ is determined by thresholding on ξ (empirically set to $0.3m$) as follows.

$$l_{\mathcal{V}} = \begin{cases} \text{stable} & \textbf{if } \tau \leq \xi \\ \text{unstable} & \textbf{if } \tau > \xi \end{cases} \tag{3.22}$$

If the label $l_{\mathcal{V}}$ is "stable", the implication is that the Voxel Block contains few disparities between the current scene and the stored model. In this case, it's stability value is incremented. If however $l_{\mathcal{V}}$ has the label "unstable", the implication is that the contents of the Voxel Block are changing, so it's stability value is reset to nought, as follows.

$$\tau_{\mathcal{V}} = \begin{cases} \tau_{\mathcal{V}} + 1 & \textbf{if } l_{\mathcal{V}} = \text{stable} \\ 0 & \textbf{if } l_{\mathcal{V}} = \text{unstable} \end{cases} \tag{3.23}$$

Voxel Blocks that are observed to be stable over a sufficiently long period of time (empirically

set to 40 frames) will be Integrated into the Static Model, as follows in Section 3.4.2.

3.4.2 Integration into Static Model from Dynamic Model

For a time step t, each Voxel Block in the Dynamic Model that has assigned to it a stable label has the entirety its Voxel TSDF values Integrated in to the static model. In a similar formulation to that of Equation 3.19 in Section 3.3.2, the update comprises Integration of new data in to a running average.

The weight update for a given Voxel v in the Stable Model $\mathbf{\Phi}^s$ with respect to a Voxel(belonging to a Voxel Block labelled "stable") $\bar{v}$ in the Dynamic Model $\mathbf{\Phi}^d$ is given as follows.

$$\phi_t^s(v) = \phi(v)_{t-1}^s(v) + \phi(\bar{v})_{t-1}^d(\bar{v}) \tag{3.24}$$

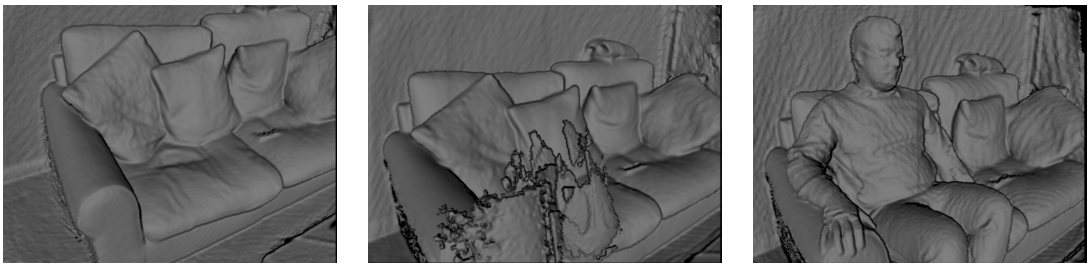
Similarly, the update for the TSDF values is as follows.

$$\mathbf{\Phi}_t^s(v) = \frac{1}{\phi_t^s(v)} \left[\phi_{t-1}^s(v) \mathbf{\Phi}_{t-1}^s(v) + \phi_{t-1}^d(\bar{v}) \mathbf{\Phi}_{t-1}^d(\bar{v}) \right] \tag{3.25}$$

3.5 Qualitative Results

Empirically, the proposed Motion Segmentation system is capable of retaining globally consistent tracking within the Dense SLAM framework for a range of scenarios that would prove to be problematic for other, static Dense SLAM systems. The experiments performed demonstrate a robustness to dynamics in various scenes, both in terms of tracking and noise artefacts in the static model. In addition, the system is robust to the addition and removal of scene components and is robust to short term occlusions, such as a person walking in front of the camera. An example of a person moving in to the the View Frustrum and sitting on a Setee is given in Figure 3.5.

In addition to the ability to reconstruct a moving object that becomes static in the scene as shown in Figure 3.5, the proposed system is also capable of segmenting dynamic objects in a scene that are undergoing nonrigid motion. Whilst these objects are segmented and labelled as "dynamic" they are not used for Camera Pose Estimation. An example of this behaviour



(a) (b) (c)

Figure 3.3: A qualitative comparison between the proposed system and InfiniTAM [12]. (a) A static scene containing a Setee is reconstructed. (b) When a person enters the scene using standard InfiniTAM, the tracking fails, leading to a corrupted scene model. (c) Using the proposed system, tracking is maintained and the person is Integrated successfully into the scene.

may be observed in Figure 3.5.

3.6 Quantitative Results

In this section, a quantitative evaluation of the system's efficacy in terms of and Camera Tracking ability is provided.

For the quantitative analysis, the system has been evaluated on the Dynamic Objects subset of the TUM RGBD dataset [57] with respect to trajectory quality. The scenes provided in the dataset contain a range of dynamic components ranging from arm movement to people walking around, occluding parts of the scene. Comparison is drawn against the standard InfiniTAM framework on which the proposed system is based, using standard, static InfiniTAM as a baseline.

Given in Table 3.6 is the Absolute Trajectory Error measures for each of the TUM Dynamic Scenes. The Absolute Trajectory Error utilises the method of Horn [58] to solve for the error incurred by mapping the trajectory of the proposed system on to the ground truth trajectory of the TUM sequence, for a given TUM Dynamic Objects sequence.

The results of Table 3.6 are visualised in Figure 3.6.

In addition to evaluating quantitatively in terms of Absolute Trajectory Error, additional results are provided in terms of Relative Trajectory Error [57]. RTE measures the relative pose error over a fixed time interval, with the final score being the RMSE over all time windows.

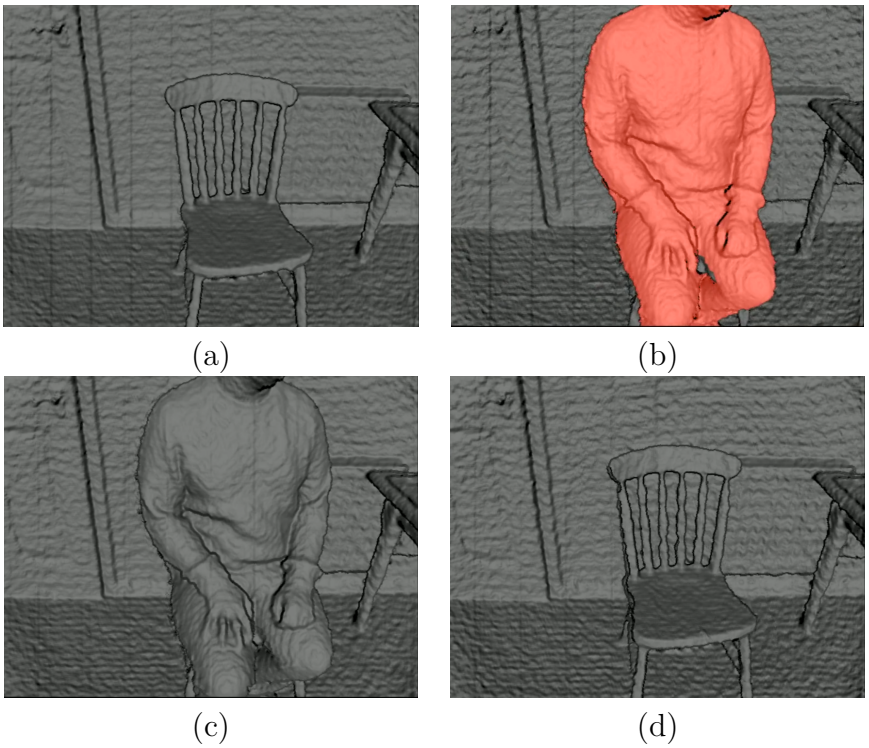


Figure 3.4: A qualitative example of the proposed systems ability to segment dynamic scene components. (a) A static scene containing a Chair is reconstructed. (b) A person enters the scene, is marked as "dynamic" and sits down. (c) After gaining a sufficient confidence score, the person is labelled "static" and is integrated in to the Static Model. (d) The person gets up from the Chair and leaves the scene. The original reconstruction of the chair from (a) is in tact.

RTE results are provided in Table 3.6 and visualised in Figure 3.6.

3.7 Application to Semantic Scene Understanding

The dynamic scene handling approach described in 3.4 can be used to prevent moving objects from being integrated into the static scene model, but for many applications, e.g. mobile robotics, there is an additional need to understand what objects are present in the scene and where they are.

In this section, it is therefore show that classifiers may be trained for the moving objects and used to recognise new instances of those objects as they enter the scene.

The Voxel Blocks in the Dynamic Model that were identified as "unstable" provide a natural representation of the dynamic parts of the scene. Where multiple dynamic objects are present, they can be separated by finding the connected components of these Voxel Blocks. For each

TUM Standard Sequence Name	*MoSeg* ATE	*Baseline* ATE
fr2-desk-with-person	**0.158 ±0.091**	0.297 ±0.193
fr3-sitting-static	0.014 ±0.008	**0.012 ±0.007**
fr3-sitting-xyz	0.064 ±0.031	**0.053 ±0.029**
fr3-sitting-halfsphere	0.142 ±0.063	**0.115 ±0.049**
fr3-sitting-rpy	**0.056 ±0.033**	0.081 ±0.051
fr3-walking-static	**0.294 ±0.153**	0.999 ±0.178
fr3-walking-xyz	**0.385 ±0.271**	0.544 ±0.343
fr3-walking-halfsphere	**0.539 ±0.360**	0.762 ±0.367
fr3-walking-rpy	**0.662 ±0.335**	0.843 ±0.365

Table 3.1: The Bbsolute Trajectory Error (ATE) results (in metres, lower is better) achieved by the proposed approach in comparison to the baseline InfiniTAM [12] framework on a variety of the standard sequences from the TUM RGBD dataset [57]. Results are in the format mean ± standard deviation. The better result (by mean) on each sequence is highlighted in bold.

TUM Standard Sequence Name	*MoSeg* RTE	*Baseline* RTE
fr2-desk-with-person	**0.023 ±0.030**	0.026 ±0.037
fr3-sitting-static	0.010 ±0.008	**0.010 ±0.008**
fr3-sitting-xyz	0.028 ±0.017	**0.028 ±0.017**
fr3-sitting-halfsphere	**0.031 ±0.033**	0.032 ±0.029
fr3-sitting-rpy	0.073 ±0.061	**0.067 ±0.065**
fr3-walking-static	**0.082 ±0.140**	0.163 ±0.308
fr3-walking-xyz	0.410 ±0.262	**0.300 ±0.252**
fr3-walking-halfsphere	**0.245 ±0.320**	0.305 ±0.374
fr3-walking-rpy	0.482 ±0.456	**0.406 ±0.364**

Table 3.2: The Relative Trajectory Error (RTE) results (in metres, lower is better) achieved by the proposed approach in comparison to the baseline InfiniTAM [12] framework on a variety of the standard sequences from the TUM RGBD dataset [57]. Results are in the format mean ± standard deviation. The better result (by mean) on each sequence is highlighted in bold.

object, a one-class Support Vector Machine (SVM) with a Polynomial Kernel is trained and used to recognise new instances of the object.

On seeing a new object, the system first tries to classify the object as one that has already been seen using all of the existing SVMs. If this fails, the system generates a label for the object and trains a new SVM for it. To make the training examples for the object, points are uniformly sampled from the object's surface (Voxels in the object's component that belong to the Zero Level Set) and compute Fast Point Feature Histogram (FPFH) descriptors [59] at those points. FPFH descriptors are geometric features that provide a per-point statistic of

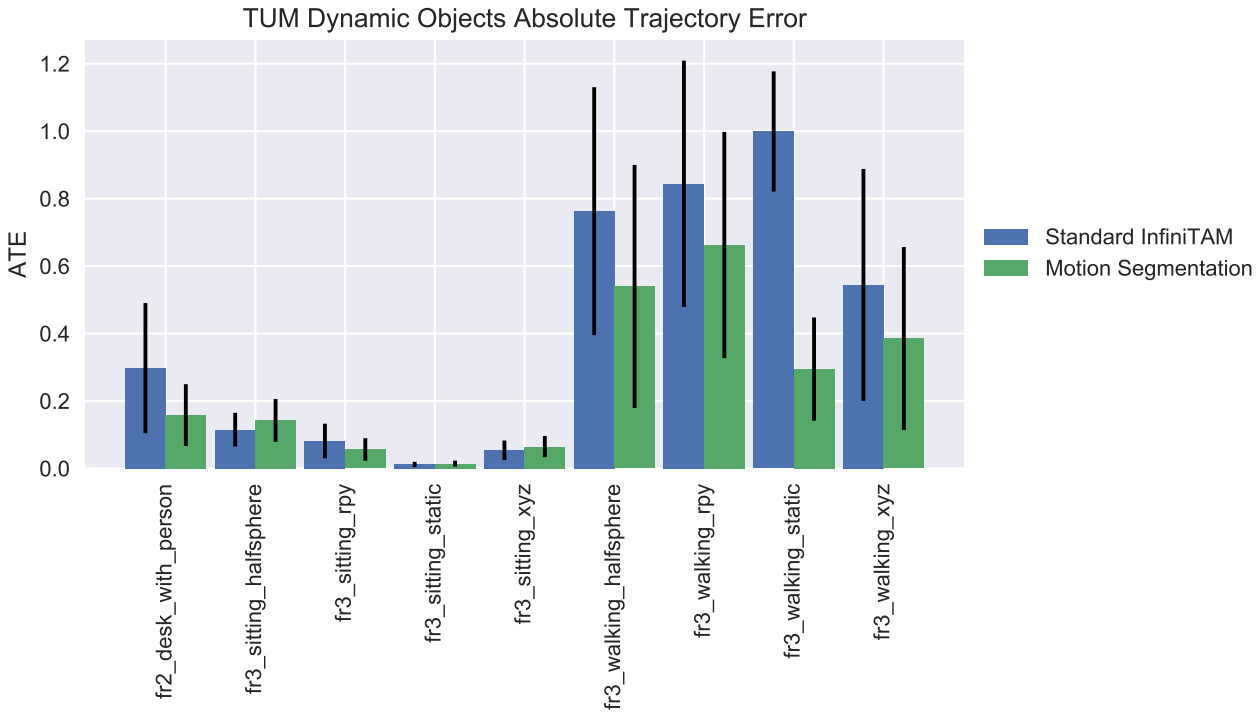


Figure 3.5: Absolute Trajectory Error for the TUM Dynamic Scenes dataset.

the curvature within some neighbourhood of the point (empirically, a 2.5cm radius around the point was used).

Figure 3.7 shows an example of this training and prediction process for dynamic objects.

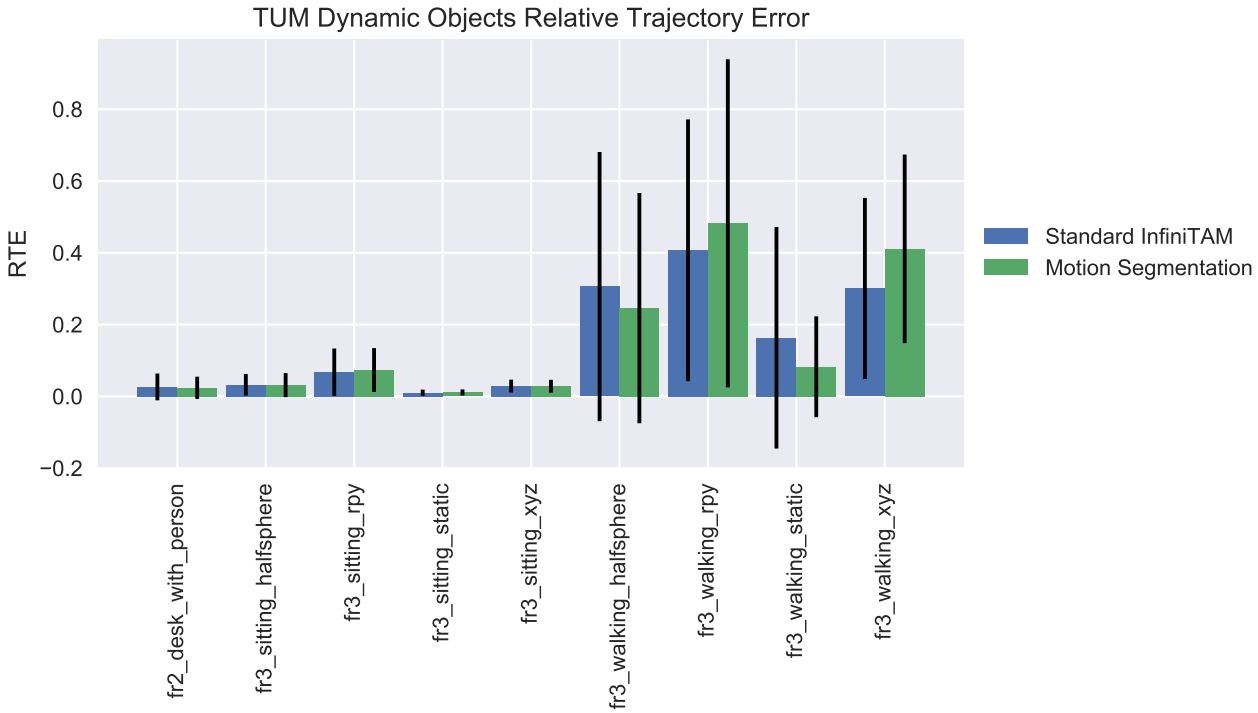


Figure 3.6: Relative Trajectory Error for the TUM Dynamic Scenes dataset.

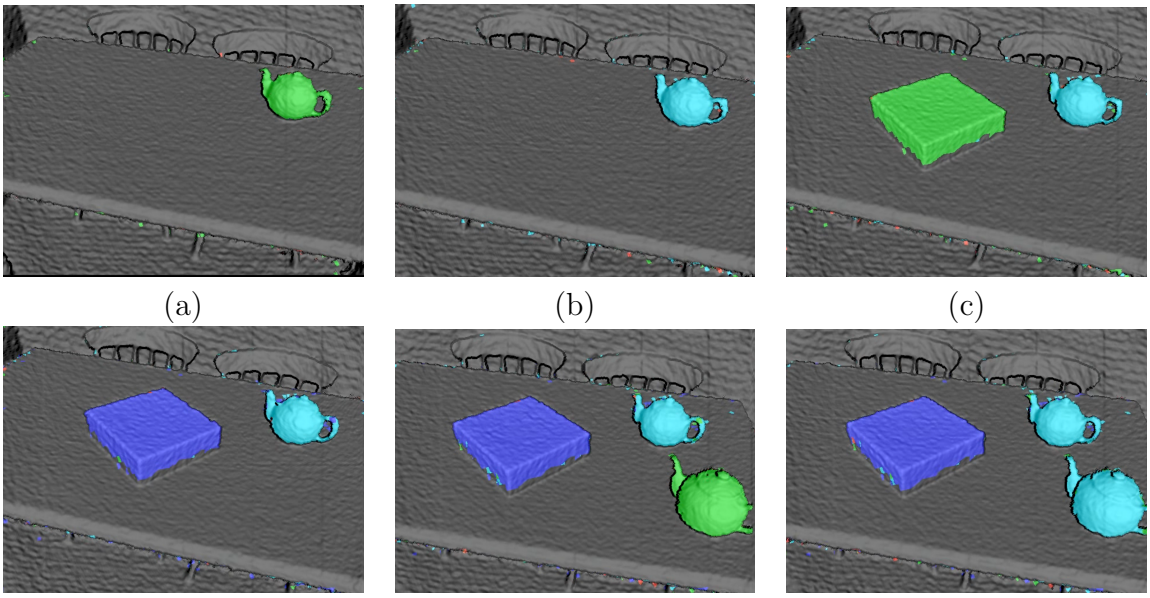


Figure 3.7: An example of the training and prediction process for dynamic objects: (a) a teapot is placed in the scene; (b) the teapot is recognised as a new object and an SVM is trained for it; (c) a box is placed in the scene; (d) the box is recognised as being distinct from the teapot, and a separate SVM is trained for it; (e); another teapot is placed in the scene; (f) the new teapot is recognised as being a teapot rather than a box, and is labelled accordingly.

Chapter 4

Probabilistic Object Reconstruction with Online Drift Correction

4.1 Introduction

Dense SLAM (Simultaneous Localisation and Mapping) has proven to be an effective paradigm for the reconstruction of scenes of moderate size, with much research on the topic driven by the availability of consumer grade depth sensing equipment. However, there is a heavy reliance on descriptive geometry in the scene when there is a lack of texture. Less descriptive geometry leads to an increase in camera tracking error and causes model inconsistencies, especially when a loop closure event occurs.

As object reconstruction can be seen as a smaller scale equivalent of the scene based dense reconstruction problem, it too is prone to the tracking drift and loop closure problem, sometimes to a prohibitive level. Often it may be desirable to perform object reconstruction in an interactive way, for example, as a component of a scene understanding system, or to procure training data for the object in question.

With a high level of interaction comes an exacerbation of the aforementioned shortcomings of dense SLAM, particularly due to the potential for frequent, repetitive motion. This is the problem that is addressed in this chapter.

In this chapter, a probabilistic object reconstruction framework is presented for the recon-

struction of rigid objects based on object appearances. The framework facilitates the correction of camera tracking drift by representing the object to be reconstructed as a collection of overlapping subsegments, such that deformations may be inferred to keep the subsegments aligned, resulting in a consistent overall model. The system utilises a volumetric representation for each of these object subsegments, as with many larger scale reconstruction systems. Each voxel in the subsegments has additional appearance posterior information pertaining to the voxels 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, the optimum object shape is optimised for within a CRF (Conditional Random Field) framework.

The proposed system is inspired by[40] in that the representation used for the shape of the object to be modelled is a volume of probabilities, pertaining to posteriors over a voxels assignment to being either on the objects surface or not. In the proposed system this volume of posterior probabilities is "fused" into with each frame, much like the fusion process in systems such as KinectFusion[5] and InfiniTAM[12].

The probabilities that are "fused" into the volume are generated from an appearance model, initialised prior to reconstruction by a Maximum Likelihood procedure over the first frame of the RGB image. There are two appearance models, one for the foreground object and one for the background, with the foreground object indicated by a bounding box on the first RGB frame. A normal distribution is fitted over the colour features of each class, foreground and background. During the fusion process, the PDF's of these distributions are evaluated on the latest colour observation for a voxel and posterior's are computed and updated accordingly in the probability volume. Only those voxels with a posterior higher for the foreground are rendered.

4.2 Related Work

4.3 Algorithm Overview

In the proposed system, the object model is divided into Subvolumes, each consisting of a TSDF, colour volume and object Probability Volume. Additionally, each has associated with it a Rigid Body Transform that specifies its pose relative to the global coordinate frame.

At each time step, a segmentation model is applied to the RGB input image to generate an object *Probability Map* defining the segmented region to be the object of interest and the remainder the background, to be discarded. Using these generated Probability Maps, the system accumulates the probabilities into the object *Probability Volume* of the active Subvolume.

As with the Dense SLAM system outlined earlier in this work, the proposed system also has *Integration*, *Tracking* and *Rendering* stages in it's pipeline(all of which are run at each time step). However, in the proposed system, there are an additional two stages to the pipeline; *Online Model Correction* and *CRF Based Segmentation*.

At the end of each frame, the online model correction algorithm is run, which infers the relative poses between the subvolumes, mitigating tracking drift. Once the reconstruction process is finished, we perform a CRF-based optimisation to refine the resulting object segmentation over all Subvolumes.

The proposed approach is not tied to the use of any one probabilistic model, though in the presented experiments PwP (Pixel Wise Posteriors) are used [43]. An overview of the object reconstruction pipeline is shown in Figure 4.3.

4.4 Probabilistic Formulation of Object Reconstruction

The surface map and camera pose are estimated using the standard KinectFusion like pipeline of [5, 12]. The surface is represented as the Zero Level Set of a TSDF discretised over voxels, with the Isosurface built by a weigted mean of new observations, as outlined in Equations 3.19 and 3.20. Camera Pose Estimation is performed with ICP, as outlined in Section **??** and is run quasi-simultaneously against the evolving map. Here, inspired by [40], this procedure is

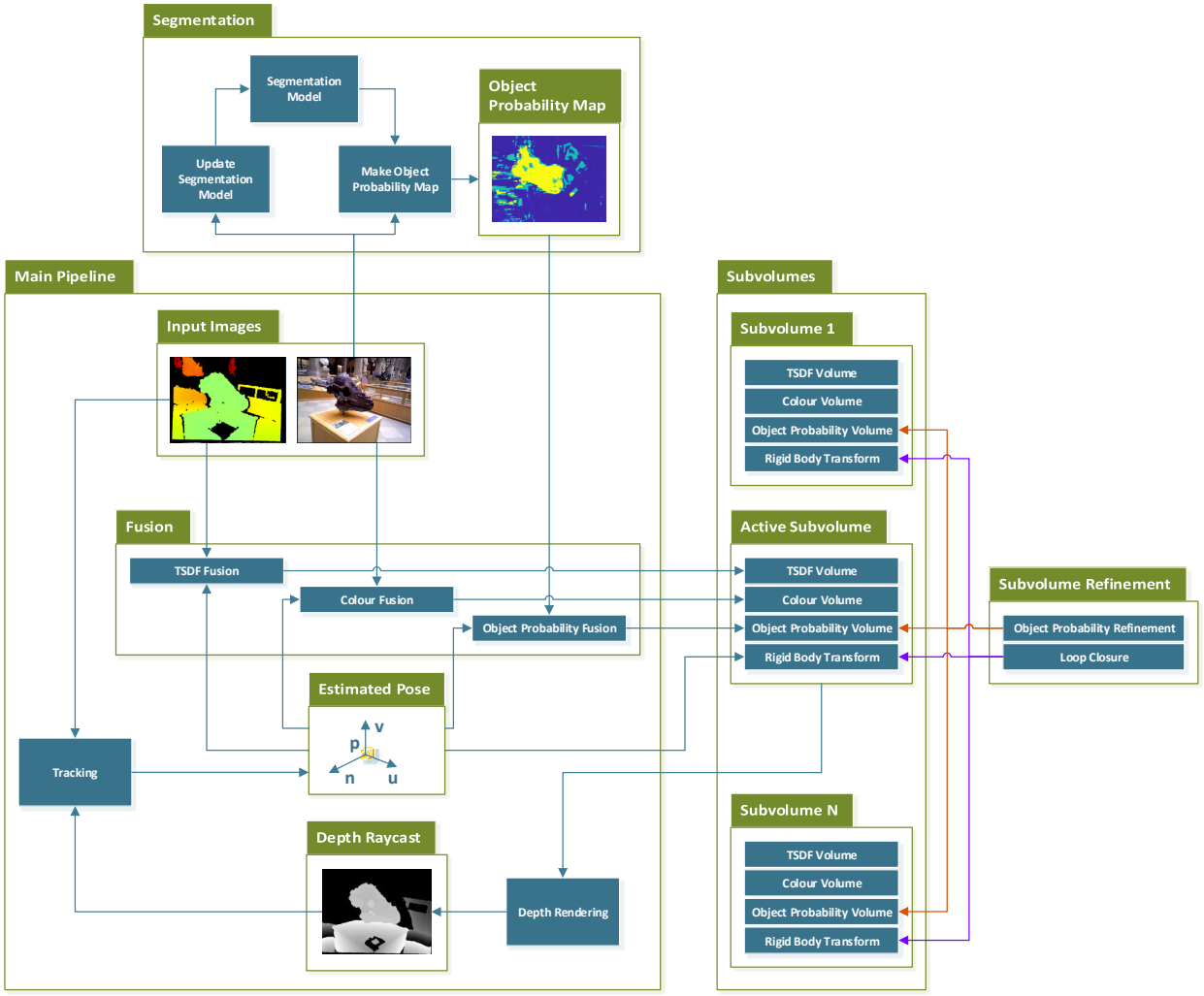


Figure 4.1: The pipeline of the proposed Object Reconstruction approach.

augmented by estimating the Posterior Probability, per map Voxel, of belonging to the object of interest. This volume of Posterior Probabilities is updated at each time step, in parallel to the fusion process in the mapping and pose estimation components of the Pipeline. The representation of the reconstructed object comprises multiple "subvolumes", each pertaining to some patch on the object surface. New subvolumes are created when sufficiently many new Voxels have been allocated and have had SDF data integrated. By ensuring overlap between the subvolumes, transformations between them can be found and pose inconsistencies addressed, online. Empirically, the threshold for starting a new subvolume is defined as the event when 50% of the Voxels fused in to the current volume are newly observed points.

4.4.1 Volumetric Appearance Model

At each observed RGBD frame, the object Posterior 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 $\Phi \subset \Psi$):

$$P(\psi \in \Phi | \Omega, p) = \prod_{t=0}^{\infty} P(\psi_t \in \Phi | \Omega_t, p_t) \tag{4.1}$$

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 RGBD 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 of interest as the product of instantaneous appearance-derived pixel-wise conditionals. Note that in the above, Φ is a discretisation of the continuous Φ in the probabilistic formulation that follows.

4.4.2 Full Joint Definition

Central to the proposed system is the aforementioned Volume of appearance based Posterior Probabilities pertaining to a Voxel wise membership of either the object Voxel set or the non object(background) Voxel set. This allows formulation of the full joint distribution over the object as the Probabilistic Graphical Model of Figure **??**.

Where Φ is the shape of the object to be reconstructed (represented as a subset of Voxels for which surface data has been Integrated into the relevant TSDF), u is the appearance model volume(aforementioned appearance Posteriors), 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.

The PGM given in Figure 4.4.2 leads to the following factorisation over the full Joint

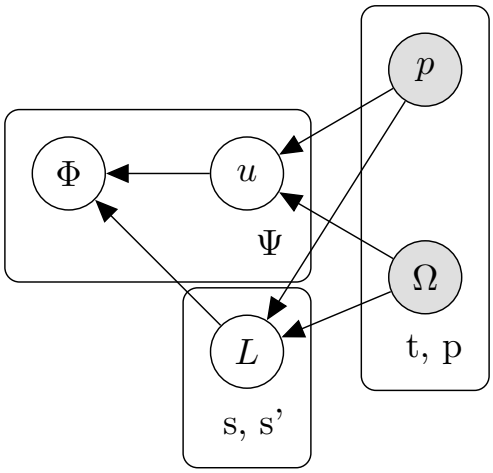


Figure 4.2: Probabilistic Graphical Model representing the full Joint Distribution over the shape $\mathbf{\Phi}$ of the object of interest.

Distribution.

$$P(\Phi, \Omega, p, u, L) = \prod_{\psi \in \Psi} \prod_{s,s' \in \mathcal{S}} P(\Phi | u_\psi, L_{s,s'}) \prod_{t=0}^{\infty} \prod_{p \in \mathcal{P}} P(u_\psi | \Omega_{p,t}, p_t) P(L_{s,s'} | \Omega_{p,t}, p_t) P(L_{s,s'}) P(p_t) P(\Omega_{p,t}) \tag{4.2}$$

Where in Equation 4.2, Ψ is the set of Voxels across all subvolumes, $\mathcal{P}$ is the set of RGBD pixels for a given frame and $\mathcal{S}$ is the set of subvolumes. Note that the notation $s, s' \in \mathcal{S}$ refers to pairs of overlapping subvolumes.

If pixel-wise independence is assumed in the RGBD observations and temporal independence is assumed in the poses, the plate containing Ω and p can be removed as shown in Figure 4.4.2.

The simplifications transforming the PGM of Figure 4.4.2 in to that of Figure 4.4.2 lead to the following Factorisation of the Joint Distribution over $\mathbf{\Phi}$.

$$P(\Phi, \Omega, p, u, L) = \prod_{\psi \in \mathbf{\Psi}} P(\Phi \mid u_\psi) \prod_{s,s' \in \mathcal{S}} P(u_\psi \mid \Omega, p, L_{s,s'}) P(L_{s,s'} \mid \Omega, p) P(L_{s,s'}) P(p) P(\Omega) \tag{4.3}$$

The formalisms defined in Figures 4.4.2 and 4.4.2 and Equations 4.2 and 4.3 describe a probabilistic framework in which online corrections can be made to the reconstructed model(piecewise over subvolumes) to counter errors caused by pose tracking inconsistencies. As with scene scale dense SLAM systems[5, 12**?**], the presented system follows a pipeline that consists of a tracking

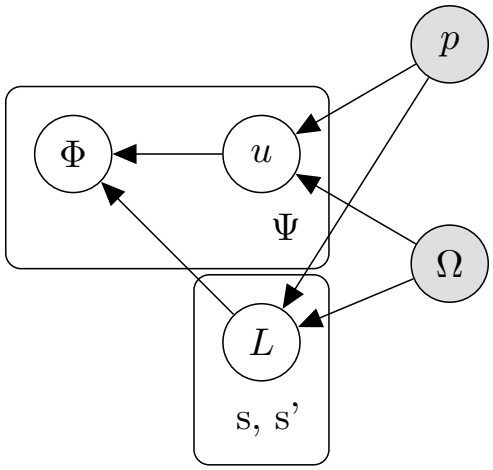


Figure 4.3: Probabilistic Graphical Model representing the simplified Joint Distribution over the shape $\mathbf{\Phi}$ of the object of interest.

stage and an integration stage, as outlined in Section 3.3.

However, the presented 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 Rigid Body Transformations to the subsegments of the reconstructed shape(the subvolumes) to correct their alignment when there are intra subsegment tracking inconsistencies. As inference on the Joint Distribution of the presented Probabilistic Model model is intractable, Conditional Independence assumptions are made that empirically do not appear to cause any functional issues.

4.4.3 Appearance Marginal

Continuing on from the formulation given in Equation 4.3, the appearance model u may be Marginalised as follows.

$$\begin{aligned} P(\Phi, \Omega, p, L) &= \int_{-\infty}^{\infty} \left[\prod_{\psi \in \mathbf{\Psi}} P(\Phi \mid u_\psi) \prod_{s,s' \in \mathcal{S}} P(u_\psi \mid \Omega, p, L_{s,s'}) P(L_{s,s'} \mid \Omega, p) P(L_{s,s'}) P(p) P(\Omega) \right] \mathrm{d}u \\ &= \prod_{\psi \in \mathbf{\Psi}} \int_{-\infty}^{\infty} \left[P(\Phi \mid u_\psi) \prod_{s,s' \in \mathcal{S}} P(u_\psi \mid \Omega, p, L_{s,s'}) P(L_{s,s'} \mid \Omega, p) P(L_{s,s'}) P(p) P(\Omega) \right] \mathrm{d}u \\ &= \prod_{s,s' \in \mathcal{S}} P(L_{s,s'} \mid \Omega, p) P(L_{s,s'}) P(p) P(\Omega) P(\mathbf{\Phi}) \end{aligned} \tag{4.4}$$

Note that the Appearance Posterior Volume outlined in Section 4.4.1 is reintroduced later in this work in Section **??** for the purposes of subvolume alignment and determining the subset of Voxels $\mathbf{\Phi} \subset \mathbf{\Psi}$ that determine the target object shape.

Further details pertaining to the inference procedure for the per-subvolume deformations is provided in Section 4.5.

4.5 Online Model Correction

The tracking consistency constraints denoted by the variables $L_{s,s'}$ such that $s, s' \in \mathcal{S}$ with $\mathcal{S}$ being the set of overlapping subvolume pairs s, s' in the Probabilistic Graphical Models given by Figures 4.4.2 and 4.4.2 can be enforced in terms of minimising the disparity between each pair of adjacent subvolumes. The effect of this minimsation being that consistency in the Pose Estimation phase of the Pipeline outlined in Figure 4.3 is enforced. The objective of this procedure is to infer a robust and consistent deformation transformation for the subvolume pair.

4.5.1 Alignment MAP Estimate

Referring back to the joint distribution of Equation 4.3, to achieve the aforementioned minimisation of disparity between overlapping subvolumes, a Maximum a Posteriori (MAP) estimate is desirable. As such, a MAP estimate over $L_{s,s'}$ in Equation 4.3 for a given subvolume pair s, s' may be derived as follows.

$$\begin{aligned} P(\Omega, p|L_{s,s'}) &\propto \frac{P(L_{s,s'}|\Omega, p)P(\Omega|p)P(p)P(L_{s,s'})}{\int_{-\infty}^{\infty} P(L_{s,s'} \mid \Omega, p)\mathrm{d}L_{s,s'}} P(\Phi) \\ &\propto P(L_{s,s'}|\Omega, p)P(\Omega|p)P(p)P(L_{s,s'})P(\Phi) \\ &\propto P(L_{s,s'}|\Omega, p)P(L_{s,s'})P(\Phi) \end{aligned} \tag{4.5}$$

Note that in the third step of Equation 4.5 the distributions $P(\Omega|p)$ and $P(p)$ are taken to be Uniform and as such may be omitted whilst retaining proportionality. The distribution $P(L_{s,s'})$ is Conjugate to $P(L_{s,s'}|\Omega, p)$ and is of the form of a Multivariate Gaussian Distribution $\mathcal{N}(\mathbf{0} \,|\, \mathbf{I})$

over the $\mathbb{SE}(3)$ deformation parameters. The choice of such a Prior Distribution is motivated by the assumption that motion between consecutive frames is minor, thus the given Prior will have the effect of constraining the $\mathbb{SE}(3)$ transformation as such. The Prior distribution $P(\Phi)$ serves as a *Surface Prior* to mitigate the effect of noise introduced in to the TSDF volumes. The form of $P(\Phi)$ shall be discussed in Section 4.5.2.

The rationale of Equation 4.5 is that the deformation $L_{s,s'}$ applied to the subvolume s maximises the Posterior Probability of observing the current pose p given the current RGBD frame Ω by reducing the variance of the result of the Pose Estimation phase of the Pipeline. As such, global tracking variance (quantified by the proportion of outliers in the result of the ICP component of the Pipeline) is reduced by enforcing local consistency, also improving global consistency and thus the quality of the resultant reconstruction.

4.5.2 Analytic Form of Alignment MAP Estimate

With the Probabilistic framework now outlined, the analytic form of the Posterior given in Equation 4.5 may be explored. The Likelihood term in Equation 4.5 quantifies the ability of the constraint $L_{s,s'}$ to maximise consistency between subvolumes s and s', with respect to observed RGBD Frame Ω and Pose p. By quantifying the Likelihood only for Voxels in s and s' that are visible in the current view frustrum at time t, the Posterior $P(\Omega, p|L_{s,s'})$ is given. As outlined in Section 4.5.1, the Prior on the constraints $L_{s,s'}$ has the form of a Multivariate Gaussian Distribution of the form $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

The form of the

4.5.3 Optimisation for MAP Inference

4.6 Volumetric Segmentation and Explicit Loop Closure Detection

4.7 Qualitative Results

4.8 Quantitative Results

Appendices

.1 Mathematical Appendices

.1.1 Rodriguez Paramaterisation Partial Derivatives

In this section the full Partial Derivatives of a Rotation Matrix $\mathbf{R}$ generated by the Formulation of Equation 3.4 in Section 3.3.1 are given as follows.

$$\frac{\partial \mathbf{R}}{\partial \alpha} = \begin{bmatrix} -\frac{2\alpha\left(\alpha^2-\beta^2-\gamma^2+1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1} & -\frac{2\alpha(2\alpha\beta+\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1} & -\frac{2\alpha(2\alpha\gamma-\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} \\ -\frac{2\alpha(2\alpha\beta-\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\alpha\left(\alpha^2-\beta^2+\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1} & -\frac{2\alpha(\alpha+2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{1}{\alpha^2+\beta^2+\gamma^2+1} \\ -\frac{2\alpha(2\alpha\gamma+\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\alpha(\alpha-2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{1}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\alpha\left(\alpha^2+\beta^2-\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1} \end{bmatrix} \tag{6}$$

$$\frac{\partial \mathbf{R}}{\partial \beta} = \begin{bmatrix} -\frac{2\beta\left(\alpha^2-\beta^2-\gamma^2+1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1} - \frac{2\beta(2\alpha\beta+\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} & -\frac{2\beta(2\alpha\gamma-\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{1}{\alpha^2+\beta^2+\gamma^2+1} \\ \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1} - \frac{2\beta(2\alpha\beta-\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} & \frac{2\beta\left(\alpha^2-\beta^2+\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1} & -\frac{2\beta(\alpha+2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} \\ -\frac{2\beta(2\alpha\gamma+\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{1}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\beta(\alpha-2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\beta\left(\alpha^2+\beta^2-\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1} \end{bmatrix} \tag{7}$$

$$\frac{\partial \mathbf{R}}{\partial \gamma} = \begin{bmatrix} -\frac{2\gamma\left(\alpha^2-\beta^2-\gamma^2+1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} & -\frac{2\gamma(2\alpha\beta+\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{1}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1} - \frac{2\gamma(2\alpha\gamma-\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} \\ -\frac{2\gamma(2\alpha\beta-\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{1}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\gamma\left(\alpha^2-\beta^2+\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} - \frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1} - \frac{2\gamma(\alpha+2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} \\ \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1} - \frac{2\gamma(2\alpha\gamma+\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} & \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1} + \frac{2\gamma(\alpha-2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} & \frac{2\gamma\left(\alpha^2+\beta^2-\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} + \frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} \end{bmatrix} \tag{8}$$

.2 Motion Segmentation Results Appendices

.2.1 Motion Segmentation additional ATE results

In this section, additional results for the Motion Segmentation system to complement those outlined in Section 3.6 are given. The results in this section assess Absolute Trajectory Error on the TUM Dynamic Objects *Validation* set. Quantitative results are given in Table .2.1 and visualised in Figure .2.1.

TUM Standard Sequence Name	*MoSeg* ATE	*Baseline* ATE
fr3-sitting-static	0.046 ±0.021	**0.030 ±0.014**
fr3-sitting-xyz	0.048 ±0.027	0.048 ±0.027
fr3-sitting-halfsphere	0.029 ±0.013	**0.028 ±0.012**
fr3-sitting-rpy	0.047 ±0.022	**0.044 ±0.020**
fr3-walking-static	**0.163 ±0.191**	0.466 ±0.252
fr3-walking-xyz	**0.092 ±0.075**	0.633 ±0.429
fr3-walking-halfsphere	**0.412 ±0.271**	0.525 ±0.325
fr3-walking-rpy	**0.082 ±0.042**	0.561 ±0.182

Table 1: The Absolute Trajectory Error (ATE) results (in metres, lower is better) achieved by the proposed approach in comparison to the baseline InfiniTAM [12] framework on a variety of the standard sequences from the TUM RGBD *Validation* dataset [57]. Results are in the format mean ± standard deviation. The better result (by mean) on each sequence is highlighted in bold.

.2.2 Motion Segmentation additional RTE results

In this section, additional results for the Motion Segmentation system to complement those outlined in Section 3.6 are given. The results in this section assess Relative Trajectory Error on the TUM Dynamic Objects *Validation* set. Quantitative results are given in Table .2.2 and visualised in Figure .2.2.

TUM Standard Sequence Name	*MoSeg* RTE	*Baseline* RTE
fr3-sitting-static	0.013 ±0.007	**0.011 ±0.007**
fr3-sitting-xyz	**0.033 ±0.021**	0.034 ±0.021
fr3-sitting-halfsphere	0.022 ±0.013	0.022 ±0.012
fr3-sitting-rpy	0.05 ±0.048	**0.048 ±0.043**
fr3-walking-static	**0.099 ±0.240**	0.163 ±0.308
fr3-walking-xyz	**0.055 ±0.039**	0.285 ±0.337
fr3-walking-halfsphere	**0.171 ±0.324**	0.211 ±0.233
fr3-walking-rpy	**0.139 ±0.067**	0.194 ±0.182

Table 2: The Relative Trajectory Error (RTE) results (in metres, lower is better) achieved by the proposed approach in comparison to the baseline InfiniTAM [12] framework on a variety of the standard sequences from the TUM RGBD *Validation* dataset [57]. Results are in the format mean ± standard deviation. The better result (by mean) on each sequence is highlighted in bold.

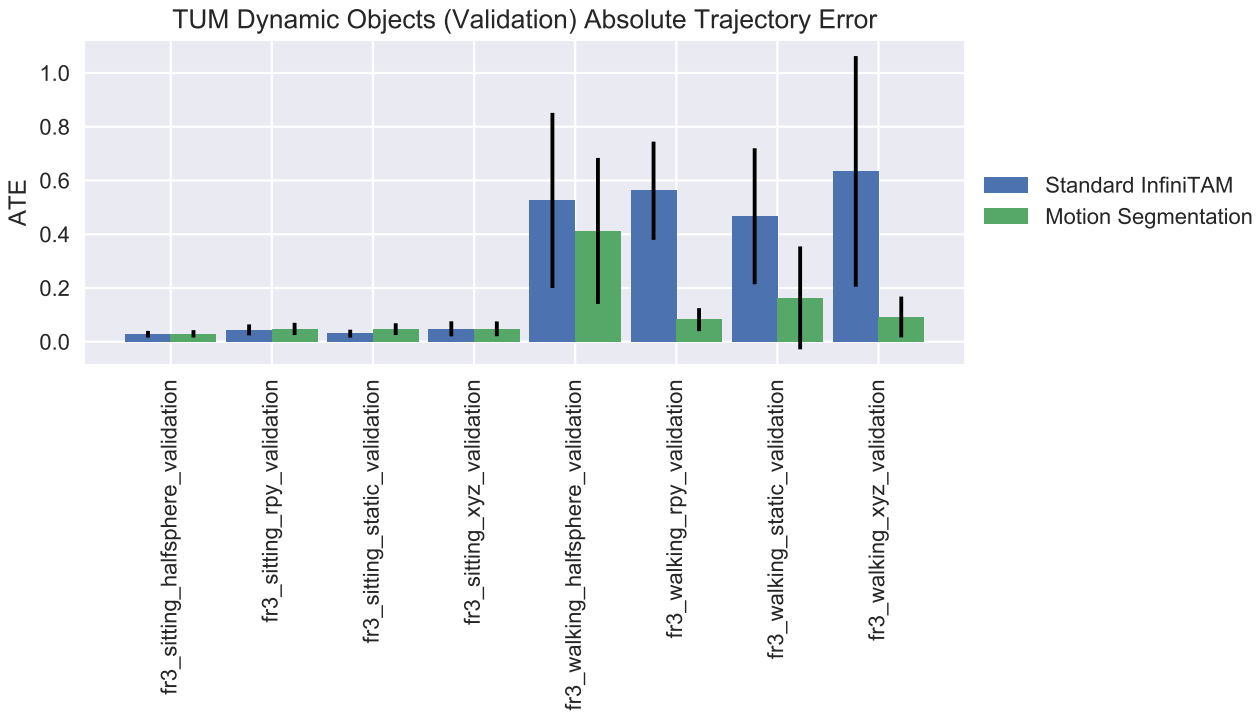


Figure 4: Absolute Trajectory Error for the TUM Dynamic Scenes *Validation* dataset.

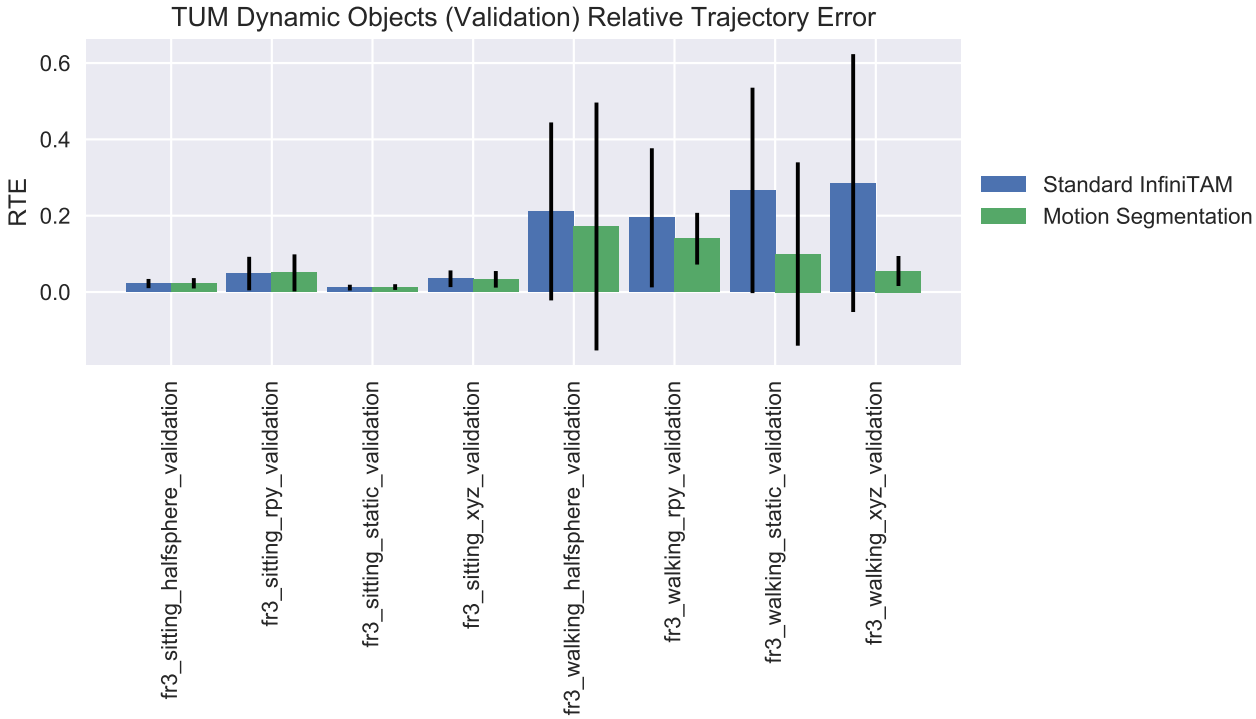


Figure 5: Relative Trajectory Error for the TUM Dynamic Scenes *Validation* dataset.

Bibliography

[1] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," vol. 14, no. 2, pp. 239–256.

[2] 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*, SIGGRAPH '96, pp. 303–312, ACM.

[3] K. Zhou, Q. Hou, R. Wang, and B. Guo, "Real-time kd-tree construction on graphics hardware," in *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, (New York, NY, USA), pp. 126:1–126:11, ACM.

[4] A. Censi, "An icp variant using a point-to-line metric," in *2008 IEEE International Conference on Robotics and Automation*, pp. 19–25.

[5] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136.

[6] M. Niessner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," vol. 32, no. 6, pp. 169:1–169:11.

[7] D. Thomas and A. Sugimoto, "A flexible scene representation for 3d reconstruction using an rgb-d camera," in *2013 IEEE International Conference on Computer Vision*, pp. 2800–2807.

[8] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, (Washington, DC, USA), pp. 1352–1359, IEEE Computer Society, 2013.

[9] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," vol. 25, no. 1, pp. 137–147.

[10] H. Pfister, M. Zwicker, J. Baar, and M. Gross, "Surfels: Surface elements as rendering primitives,"

[11] R. F. Salas-Moreno, B. Glocken, P. H. J. Kelly, and A. J. Davison, "Dense planar slam," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 157–164.

[12] V. A. Prisacariu, O. Kähler, M. Cheng, C. Y. Ren, J. P. C. Valentin, P. H. S. Torr, I. D. Reid, and D. W. Murray, "A framework for the volumetric integration of depth images," vol. abs/1410.0925.

[13] O. Kahler, V. Adrian Prisacariu, C. Yuheng Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," vol. 21, no. 11, pp. 1241–1250.

[14] 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," vol. 34, no. 4-5, pp. 598–626.

[15] Q.-Y. Zhou and V. Koltun, "Depth camera tracking with contour cues," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 632–638.

[16] O. Kähler, V. A. Prisacariu, and D. W. Murray, *Real-Time Large-Scale Dense 3D Reconstruction with Loop Closure*, pp. 500–516. Springer International Publishing.

[17] J. Civera, D. Galvez-Lopez, L. Riazuelo, J. D. Tardos, and J. M. M. Montiel, "Towards semantic slam using a monocular camera," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1277–1284, Sept 2011.

[18] J. Stuckler, N. Biresev, and S. Behnke, "Semantic mapping using object-class segmentation of rgb-d images," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3005–3010, Oct 2012.

[19] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Niessner, A. Criminisi, S. Izadi, and P. Torr, "Semanticpaint: Interactive 3d labeling and learning at your fingertips," vol. 34, no. 5, pp. 154:1–154:17.

[20] S. Golodetz, M. Sapienza, J. P. C. Valentin, V. Vineet, M. Cheng, A. Arnab, V. A. Prisacariu, O. Kähler, C. Y. Ren, D. W. Murray, S. Izadi, and P. H. S. Torr, "Semanticpaint: A framework for the interactive segmentation of 3d scenes," vol. abs/1510.03727.

[21] H. Abdulsalam, D. B. Skillicorn, and P. Martin, "Streaming random forests," in *11th International Database Engineering and Applications Symposium (IDEAS 2007)*, pp. 225–232.

[22] E. P. Xing, M. I. Jordan, and S. Russell, "A generalized mean field algorithm for variational inference in exponential families," in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, (San Francisco, CA, USA), pp. 583–591, Morgan Kaufmann Publishers Inc.

[23] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, eds.), pp. 109–117, Curran Associates, Inc.

[24] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," vol. 35, pp. 1798–1828.

[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pp. 580–587, IEEE Computer Society.

[26] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Synthcam3d: Semantic understanding with synthetic indoor scenes," vol. abs/1505.00171.

[27] T. Cavallari and L. Di Stefano, *On-Line Large Scale Semantic Fusion*, pp. 83–99. Cham: Springer International Publishing, 2016.

[28] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," vol. 39, no. 4, pp. 640–651.

[29] L. V. Tsap, D. B. Goldof, and S. Sarkar, "Nonrigid motion analysis based on dynamic refinement of finite element models," vol. 22, no. 5, pp. 526–543.

[30] J. Chen, X. Wu, M. Y. Wang, and F. Deng, *Human Body Shape and Motion Tracking by Hierarchical Weighted ICP*, pp. 408–417. Springer Berlin Heidelberg.

[31] D. Sun, E. B. Sudderth, and M. J. Black, "Layered segmentation and optical flow estimation over time," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1768–1775.

[32] M. Unger, M. Werlberger, T. Pock, and H. Bischof, "Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1878–1885.

[33] E. Herbst, X. Ren, and D. Fox, "Rgb-d flow: Dense 3-d motion estimation using color and depth," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2276–2282.

[34] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, pp. 25–36. Springer Berlin Heidelberg.

[35] J. Stueckler and S. Behnke, "Efficient dense 3d rigid-body motion segmentation in rgb-d video," in *Proc. of the British Machine Vision Conference (BMVC)*.

[36] 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 *Proceedings of the 2013 International Conference on 3D Vision*, 3DV '13, pp. 1–8, IEEE Computer Society.

[37] S. Perera, N. Barnes, X. He, S. Izadi, P. Kohli, and B. Glocker, "Motion segmentation of truncated signed distance function based volumetric surfaces," IEEE.

[38] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 343–352.

[39] L. Kavan, S. Collins, and J. Zara, "Dual quaternions for rigid transformation blending," tech. rep.

[40] K. Kolev, T. Brox, and D. Cremers, "Robust variational segmentation of 3d objects from multiple views," in *Proceedings of the 28th Conference on Pattern Recognition*, DAGM'06, (Berlin, Heidelberg), pp. 688–697, Springer-Verlag.

[41] 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*, pp. 1630–1637.

[42] C. Ren, V. Prisacariu, D. Murray, and I. Reid, "Star3d: Simultaneous tracking and reconstruction of 3d objects using rgb-d data," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1561–1568.

[43] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *Proceedings of European Conference on Computer Vision*.

[44] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi, "3d scanning deformable objects with a single rgbd sensor," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 493–501.

[45] T. Gupta, D. Shin, N. Sivagnanadasan, and D. Hoiem, "3dfs: Deformable dense depth fusion and segmentation for object reconstruction from a handheld camera," vol. abs/1606.05002.

[46] V. A. Prisacariu and I. Reid, "Shared shape spaces," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2587–2594.

[47] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," vol. 6, pp. 1783–1816.

[48] A. Dame, V. A. Prisacariu, C. Y. Ren, and I. D. Reid, "Dense reconstruction using 3d object shape priors," in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pp. 1288–1295.

[49] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[50] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[51] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[52] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem, "Completing 3d object shape from one depth image," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[53] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," vol. 32, no. 11, pp. 1231–1237.

[54] J. Gwak, C. B. Choy, A. Garg, M. Chandraker, and S. Savarese, "Weakly supervised generative adversarial networks for 3d reconstruction," *CoRR*, vol. abs/1705.10904, 2017.

[55] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1992.

[56] M. D. Shuster, "Survey of attitude representations," vol. 41, pp. 439–517.

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

[58] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, pp. 629–642, Apr 1987.

[59] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *in In Proceedings of the International Conference on Robotics and Automation (ICRA*, 2009.