Dynamic SLAM, Object Reconstruction, Shape and Pose Prediction for 3D Scene Understanding

Department of Engineering Science

University of Oxford

Jack Miles Hunt

Kellogg College

Michaelmas Term 2019

This Thesis is submitted to the Department of Engineering Science, University of Oxford, for the degree of Doctor of Philosophy.

DECLARATION

I declare that the work contained in this Thesis is entirely my own, and except where otherwise indicated, describes my own research.

ACKNOWLEDGEMENTS

I would first like to express my gratitude to the Engineering and Physical Sciences Research Council for their support of the Centre for Doctoral Training in Autonomous, Intelligent Machines and Systems, without which I could not have embarked on my DPhil journey.

Additionally, I cannot understate my gratitude to the personnel of the AIMS CDT who from the very beginning have been supportive and encouraging. In particular, I would like to thank Wendy Adams for her personal support when I have faced difficulties and for introducing me to the Fish Finger Sandwich at the Royal Oak!

Of course, none of the work contained in this Thesis would have been possible without the support and guidance of my DPhil supervisors, Prof. Philip Torr and Prof. Victor Prisacariu. The work that they have overseen in their respective research groups has been a source of academic inspiration. Additionally, I would like to extend my thanks to Stuart Golodetz and Michael Sapienza, both of whom helped me immeasureably at the start.

Of course, my partner Zoe, my family and my friends must be noted here for their continued patience with me for the duration of my DPhil. I am sure that I have been difficult at times!

ABSTRACT

Advances in 3-dimensional (3D) computer vision have tremendously impacted the way that humans and computers interact. Applications of 3D vision, such as virtual reality video games and robotics are underpinned by a range of computer vision competencies, including pose estimation, mapping and semantic understanding. However, many challenging research problems remain, three of which are the focus of this work.

Firstly, the difficult task of dense mapping in dynamic environments is tackled, utilising a novel scene representation allowing dynamic and static components to be handled separately. This approach demonstrates improved pose estimation accuracy in dynamic scenes, compared to established reconstruction approaches. The second topic of this Thesis is 3D object reconstruction, with a novel representation and formulation that provides online error correction. This approach demonstrates improved reconstruction quality and geometric accuracy compared to both a start of the art method and a vanilla approach. The final focus of this work is the simultaneous inference of object shape and pose in large scale, outdoor environments. The approach taken regresses shape and pose in a combined supervised/weakly-supervised manner, utilising a combination of Convolutional Neural Networks and Gaussian Processes.

This Thesis provides a foundation for further research in this area. However, immediate applications are evident. The motion segmentation and dense mapping approach allows for operation in previously prohibitive scenarios, such as robotics. The 3D object reconstruction work is applicable to the collection of geometrically consistent 3D object data. Finally, the simultaneous inference of shape and pose is applicable to modelling scenarios of specific semantic interest, where an entire scene need not

be reconstructed, preliminarily demonstrating potential for large scale, semi-dense, geometric mapping.

NOTATION AND ABBREVIATIONS USED

MATHEMATICAL NOTATION

This preliminary section introduces the mathematical notation used in this work. The following table outlines essential notational details, separated into sections on Sets, Fields and Groups, Linear Algebra, Sums and Products, Calculus and Probability Theory.

Symbolic Form	Meaning
$\mathcal{A}$	Calligraphic variables indicate mathematical sets (unless otherwise indicated).
$\{a, b, c\}$	A mathematical set consisting of the elements a, b and c.
$\{a \mid \rho(a)\}$	A mathematical set whose elements are defined by predicate $\rho(a)$.
$a \in \mathcal{A}$	The element a, in the set $\mathcal{A}$.
$f(a) \forall a \in \mathcal{A}$	$f(a)$ for all a in $\mathcal{A}$.
$\lvert\mathcal{S}\rvert$	Cardinality of $\mathcal{S}$; the number of elements in $\mathcal{S}$.
$\inf \mathcal{S}$	Infimum of $\mathcal{S}$.
$\sup \mathcal{S}$	Supremum of $\mathcal{S}$.
$\mathbb{R}^N$	The field of real numbers, of dimension N.
$\mathbb{SE}(3)$	The Special Eucledian Group of 4×4 transform matrices.
$\mathbb{SO}(3)$	The Special Orthogonal Group of 3×3 rotation matrices.
$\mathbf{A}$, $\mathbf{\Theta}$	Bold, uppercase symbols indicate matrix or tensor quantities.

$\mathbf{a}$, $\boldsymbol{\theta}$	Bold, lowercase symbols indicate vector quantities.
$\|\mathbf{a}\|_n$	n-norm of a vector.
$\mathbf{a}^\top$, $\mathbf{A}^\top$	Transpose of a vector or matrix.
$\mathbf{A}^{-1}$	The matrix inverse of $\mathbf{A}$.
$\mathrm{tr}(\mathbf{A})$	Trace of the matrix A.
$\sum_{a=0}^{N} a$	The sum from 0 to N of a.
$\sum_{a \in \mathcal{A}}$	The sum of the elements in $\mathcal{A}$.
$\prod_{a=0}^{N} a$	The product from 0 to N of a.
$\frac{\partial f(\mathbf{a})}{\partial \mathbf{a}_i}$	Partial derivative of f with respect to the i^{th} element of $\mathbf{a}$.
$\nabla f(\mathbf{a})$	Gradient vector of $f(\mathbf{a})$. The vector of partial derivatives.
$\int f(a) da$	Infedinite integral over $f(a)$ with respect to a.
$\int_m^n f(a) da$	Definite integral over $f(a)$ with respect to a, between m and n.
$P(a, b, c)$	Joint distribution over the random variables a, b and c.
$P(a \mid b, c)$	Distribution over the random variable a, conditioned on b and c.
$P(a)$	Marginal distribution over the random variable a.
$\mathcal{N}(a \mid \mu, \sigma)$	Gaussian distribution parameterised by mean μ and standard deviation σ.
$\mathcal{N}(\mathbf{a} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian distribution parameterised by mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.
$\mathbb{E}(a)$	Expectation over the random variable a.
$b \sim P(a)$	A sample b, drawn from the distribution $P(a)$.
$f(a) \sim \mathcal{GP}(a)$	A function $f(a)$, drawn from the Gaussian Process $\mathcal{GP}(\boldsymbol{\mu}), \boldsymbol{\Sigma}$.

ABBREVIATIONS

In this preliminary section, a reference of abbreviations used in this work is given. The abbreviations and their associated meanings are given in the following table. Abbreviations are listed in order of first appearance in the text.

Abbreviation	Meaning
RF	Random Forest.
MRF	Markov Random Field.
DNN	Deep Neural Network.
2D	Two Dimensional.
3D	Three Dimensional.
VR	Virtual Reality.
AR	Augmented Reality.
RGB	Red-Green-Blue.
RGBD	Red-Green-Blue-Depth.
SLAM	Simultaneous Localisation and Mapping.
AI	Artificial Intelligence.
ML	Machine Learning.
TAM	Tracking and Mapping.
GPU	Graphical Processing Unit.
CPU	Central Processing Unit.
GRAM	Graphical Random Access Memory.
DoF	Degrees of Freedom.
SDF	Signed Distance Function.
TSDF	Truncated Signed Distance Function.

ICP	Iterative Closest Points.
ATE	Absolute Trajectory Error.
RTE	Relative Trajectory Error.
RMSE	Root Mean Squared Error.
SVM	Suport Vector Machine.
FPFH	Fast Point Feature Histogram.
CRF	Conditional Random Field.
MLE	Maximum Likelihood Estimate.
PDF	Probability Density Function.
PMF	Probability Mass Function.
CDF	Cumulative Density Function.
PwP	Pixel-wise-Posteriors.
PGM	Probabilistic Graphical Model.
MAP	Maximum a Posteriori.
CNN	Convolutional Neural Network.
GP	Gaussian Process.
GPLVM	Gaussian Process Latent Variable Model.
DCT	Discrete Cosine Transform.
IDCT	Inverse Discrete Cosine Transform.
PCA	Principal Component Analysis.
RoI	Region of Interest.
SGD	Stochastic Gradient Descent.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

1

INTRODUCTION

This introductory chapter outlines the motivation and background of this thesis, as well as it's objectives and structure.

In recent years there has been much research activity in the field of three dimensional (3D) computer vision, a field concerned with the processing of 3D geometric data for machine vision. This work addresses a number of open technical challenges within the field of active, 3D vision. Specifically, the 3D reconstruction of dynamic environments, robust 3D reconstruction of arbitrary objects and, the prediction of shape and pose of objects. These areas of research are of interest to the computer vision community due to the broad application potential of such systems. Scene reconstruction for example, has applications ranging from mobile robotics to recreating the physical world for Virtual Reality (VR) viewing. Object reconstruction has applications including the reproduction of tangible objects via 3D printing and building 3D object models for the training of machine learning systems. Finally, shape and pose prediction allows representations of a machine's environment to be inferred when traditional methods of reconstruction may not be feasible.

1.1 3D SCENE RECONSTRUCTION AND UNDERSTANDING

Driven by the availability of consumer grade depth sensing equipment such as the Microsoft Kinect Red-Green-Blue-Depth (RGBD) sensor (introduced by Microsoft in 2009 for the XBox 360), there has been a renewed interest in dense scene reconstruction. Advances in recent years have allowed for the creation of digital reconstructions of the tangible world with consumer grade computer equipment [1, 2, 3]. Such systems iteratively integrate observed world points into a *global* model, such that over time, a smooth representation of the observed world surfaces is built. In addition to the integration of such information into a model, there is the task of inferring how the sensor has moved in world space, such that the observed points may be transformed and integrated into the appropriate model location. The amalgamation of these two tasks is known as Simultaneous Localisation and Mapping (SLAM). The basic SLAM pipeline is given in Figure 1.1. A typical reconstruction using such a system is given in Figure 1.2.

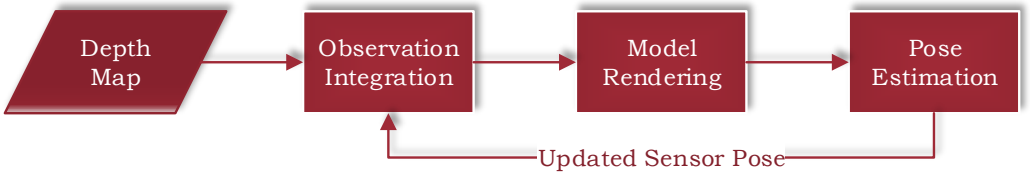


Figure 1.1: A high level overview of the basic SLAM pipeline.

There has been much advancement in the two dimensional (2D) semantic scene understanding literature [4, 5, 6], which can be utilised within the context of 3D vision to introduce a semantic component to dense SLAM systems [7, 8, 9]. Such a

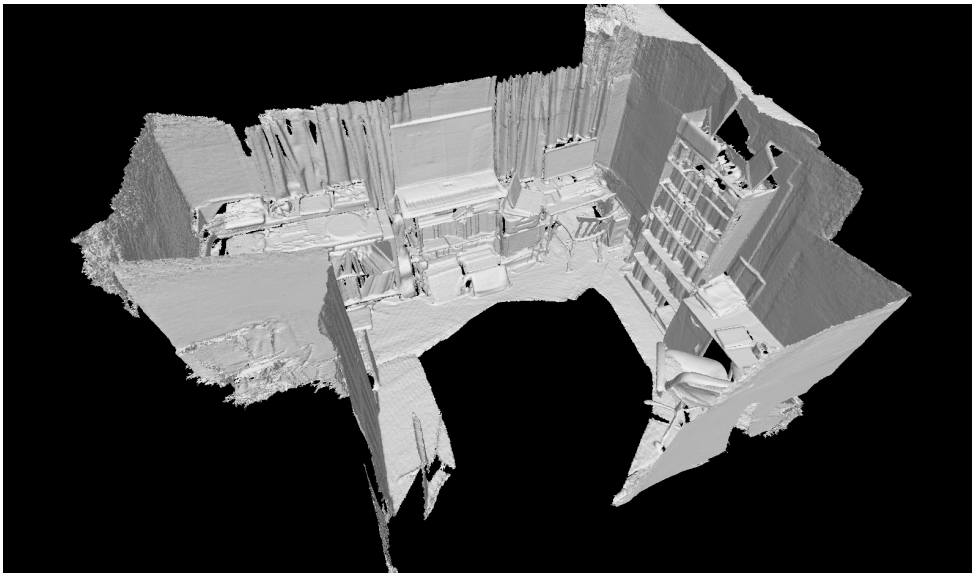


Figure 1.2: An example of room-scale dense reconstruction using a dense SLAM system.

combination of techniques provides an adaptable component to Augmented Reality (AR) and robotics applications. Early work on amalgamating the two areas of research has allowed one to view a reconstruction of their environment in VR and interactively label some of the objects within it, with the system inferring the remaining labels. An example of the output of such a system is given in Figure 1.3.

Though the results of the systems shown in Figures 1.2 and 1.3 represent impressive advances in computer vision, there are however, open technical challenges. One such challenge is the successful modelling of real environments in which there are *dynamic* components (such as people walking in the camera's view). The traditional dense SLAM pipeline is unable to accurately build a globally consistent model in such environments. In addition, when using a combined reconstruction and semantics system such as that shown in Figure 1.3, many of the descriptive cues that enable the

[1] Copyright *Golodetz et al*, 2015.
http://www.robots.ox.ac.uk/~tvg/projects/SemanticPaint/index.php

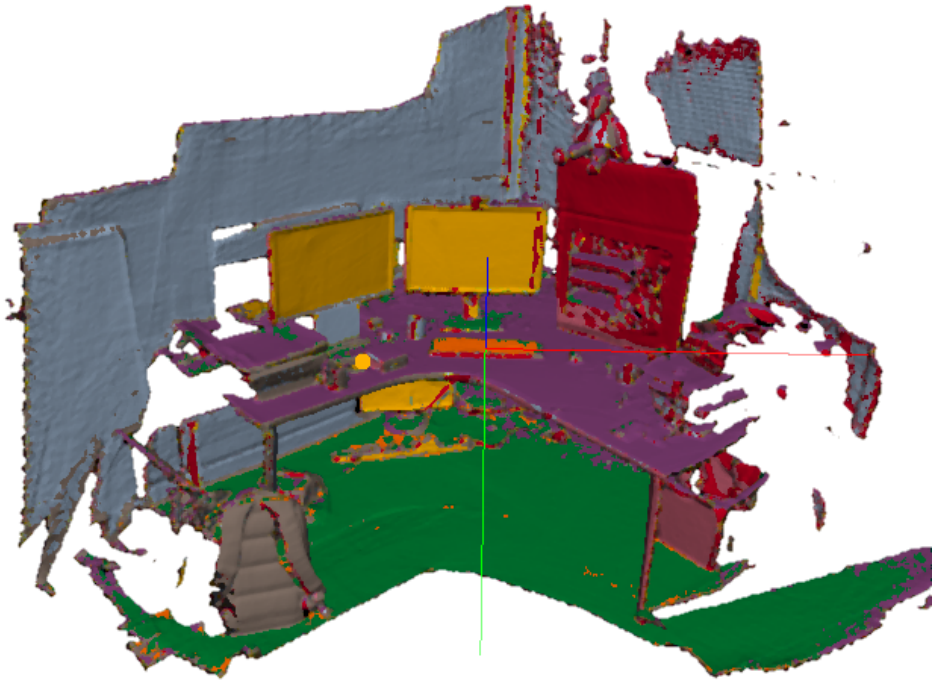


Figure 1.3: An example of semantic SLAM.[1]

segmentation to be performed rely on features utilising 2D image information. As such, there is no true 3D object learning and recognition.

As outlined, traditional dense SLAM systems have difficulty performing dense reconstruction in an environment where there is motion. The aforementioned sensor pose estimation phase in these systems is prone to error or failure in such a scenario. The reason for this is due to the reliance on point based correspondences between frames. If a static environment is being modelled, then a high number of valid point correspondences will be found. However, when motion (independent of the sensors motion) is introduced into the scene, invalid correspondences may be found. For example, points that belong to a non moving object such as a chair may erroneously be matched to those on a moving scene component, such as a walking person. Such erroneous correspondences can incur failure cases ranging from moderate model inconsistencies to total loss of sensor tracking [10].

Though there are many use cases for static scene reconstruction, the lack of robustness to dynamics is prohibitive in scenarios where a high level of machine perception is required. For example, if reconstructing a busy working environment in which there is a high level of dynamics (people walking, doors opening etc), an ideal reconstruction would not include artefacts of such motion. As such, the reconstruction system would be required to identify such components and account for them in the reconstruction process. Additionally, a system that is capable of detecting and segmenting such motion would additionally be capable of extracting pertinent cues for object recognition.

1.2 OBJECT RECONSTRUCTION

Modern machine learning provides much of the semantic and contextual information required to make meaningful inferences over the state of the world, as observed by a sensor (such as a camera). Many advances have been made in recent years on the tasks of object detection and semantic segmentation, in standard 2D images [4, 5, 6]. The application of such techniques in 3D vision allows for semantic reasoning and/or discrimination about 2D representations of 3D objects, as shown in the semantic SLAM system of Figure 1.3, in which semantic and/or class labellings of 3D objects are applied to a dense 3D reconstruction of a scene.

However, there are many technical challenges that must be overcome before such efficacy on these tasks is reached for the *true* 3D case (where both learning and inference are performed from geometry). Many modern Artificial Intelligence (AI) and Machine Learning (ML) algorithms require vast quantities of data to learn to perform a given task successfully. This is not prohibitive for systems that operate on standard 2D images, due to the abundance of available data. However, for the 3D case there

is not a comparable volume of 3D data with real world geometric information from which a system can learn to perform complex tasks in the real world. One method of obtaining such geometric data is the reconstruction of objects, providing geometrically accurate models of real world objects. An example of an object centric reconstruction is given in Figure 1.4.

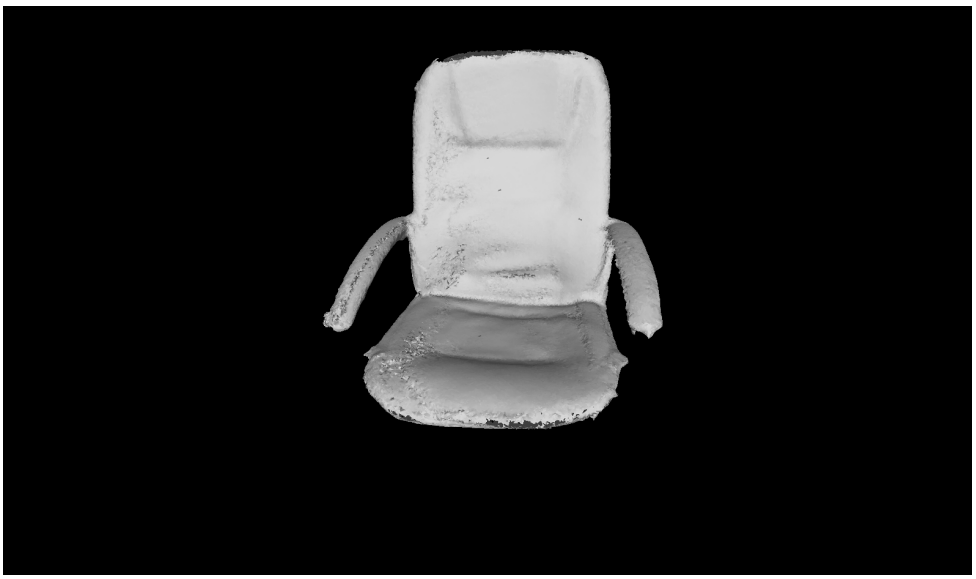


Figure 1.4: An example of a reconstructed Chair.

A related problem to that of sensor pose estimation as outlined in Section 1.1, is pose estimation when performing reconstruction of individual objects. As with the larger scale case, point correspondences are problematic. One prominent reason is that for a smaller object versus a full scale scene (such as a room), there is less geometric data available in the former case than in the latter. As with the larger scale reconstruction, inconsistencies in the pose estimation phase can have varying effects on the resultant reconstruction. Such inconsistencies in the reconstruction can have a detrimental effect on learning based systems for 3D tasks if used to train such models. This is particularly troublesome as inconsistencies in object scale models in some cases have a more pronounced effect than in the case where an entire room is being reconstructed.

Though the model consistency problem for objects may be circumvented somewhat when using specialist equipment such as modelling turntables and laser range scanners, there are financial and practical issues that can be prohibitive. Thus, the ability to build high quality, consistent reconstructions of arbitrary objects with commodity depth sensing and computing hardware is desirable. Additionally, such a system would be required to perform robustly in non-specialised scenarios, such as scanning objects in a busy setting where there is no *ideal* object reconstruction setup and motion not related to the object of interest may occur in the sensors view. Thus, it is desirable for a dynamic SLAM system to be able to separate a given object from other components in view.

1.3 SHAPE AND POSE PREDICTION

Though for many reconstruction scenarios the approaches outlined in Sections 1.1 and 1.2 are applicable, there are some situations in which the aforementioned approaches are not practical. For a complete, closed model, the reconstruction based approaches require that the object be fully observable, such that full coverage with the depth sensor is possible. As such, a clear failure point is the case in which the object is not fully observable, for example, when reconstructing a large object that is on a wall.

Additionally, the scene and object reconstruction approaches previously outlined depend on the iterative integration of observed range data. This approach may be troublesome in a scenario where an object may not be visible to the sensor for a sufficient period of time to build a smoothly reconstructed model. Circumventing this issue would require a very high framerate sensor. Additionally, the highly dynamic nature of such a scenario is likely to be problematic in a similar manner to the case of dynamic dense SLAM.

As such, a desirable approach to the 3D modelling of objects in problematic environments would not rely on direct reconstruction (in the sense of the integration of range data into a 3D model). Rather, an inference based approach is appropriate, due to the inherent stochasticity of building a full model of a partially observable object. Additionally, due to the lack of a separate frame-to-frame pose estimation phase, such a system would be required to infer accurate shape and pose from input data that is not necessarily temporally consistent.

1.4 TECHNICAL AIMS AND THESIS STRUCTURE

The aforementioned technical challenges pertain to the dense reconstruction of dynamic scenes, the reconstruction of objects and the reconstruction of objects for which no full view is available. As such, the following main research challenges are addressed in this work.

- The dense reconstruction of dynamic environments.
  - With real time performance.
  - With comparable reconstruction quality to static counterpart.
  - With an improvement in pose estimation over static counterpart.
- Identifying the dynamic components of a scene.
  - Utilising for object recognition.
- The reconstruction of arbitrary objects in a consistent manner.
  - With comparable reconstruction quality to scene based alternative.
  - With commodity hardware for wider applicability.
  - Without known pose.

- The inference of object centric scene properties where traditional reconstruction may not be possible.
  - Pose inference.
  - Experimentation with shape inference.
  - Without requiring temporally consistent frames, averting tracking errors.

With the central technical challenges of this work outlined, the remainder of this work is structured as follows. Firstly, Chapter 2 provides a comprehensive survey of the literature pertinent to this work. Initially, a survey of the dense SLAM (as introduced earlier in this chapter) literature is provided. The research outlined in this section is fundamental to much of the content of this work. Additionally, relevant works on semantics (such as semantic SLAM) are reviewed. Next is an assessment of relevant research on the topic of dynamics in 3D vision; topics include motion segmentation, optical and scene flow. Much of the material reviewed in this section is pertinent to the subject matter of Chapter 3. The next major area of research to be reviewed is on the topic of object reconstruction; relevant background to the topic of Chapter 4. Finally, Chapter 2 concludes with an assessment of the literature on the topics of pose prediction and shape prediction.

Chapter 3 introduces the approach taken in this work to the problem of dense reconstruction in dynamic environments (environments with moving components). The chapter begins by outlining fundamental concepts in the static dense SLAM pipeline that shall be fundamental to much of the content in this work. Following this fundamental material, an approach to performing dense reconstruction and motion segmentation in dynamic scenes is presented. The method outlined in this chapter is evaluated against a state of the art dense SLAM system for static scenes, to which the presented approach demonstrates an overall improvement versus the static dense SLAM system. Additionally, the qualitative results of the presented approach

demonstrate high quality resultant reconstructions on the test scenes. Furthermore, a demonstration of utilising motion segmentation to perform rudimentary object recognition using 3D geometric features is given.

Chapter 4 introduces a novel approach to the segmentation and reconstruction of individual objects. The chapter outlines a novel probabilistic approach to object reconstruction that reduces inconsistencies in pose estimation, which positively impacts the overall reconstruction quality. The approach presented in this chapter works with commodity computer and depth sensing equipment (though in principle it is trivially extensible, by design) and yields high quality reconstructions. Reconstruction quality is evaluated quantitatively and qualitatively against multiple alternative approaches. A quantitative evaluation of pose estimation quality is also provided, demonstrating an improvement over alternative approaches. The work in this chapter has been peer reviewed and published in the *International Conference on 3D Vision*[2].

Chapter 5 approaches the problem of performing inference of shape and pose simultaneously. The work in this chapter is notably different in nature relative to the approaches taken in Chapters 3 and 4. The work in outlined in this chapter utilises a data driven, non-SLAM based approach to learn predictive models for shape and pose in a semi-supervised manner. A full view of the object of interest is not required, nor is temporal consistency between frames; ad-hoc prediction can be performed for arbitrarily sequenced frames.

Chapter 6 is a discussion chapter, that begins by providing a brief outline of the approaches, contributions and results of Chapters 3, 4 and 5. Followed by a more granular assessment of each, within the context of the research objectives outlined in Section 1.4. Additionally, consideration is given to the limitations of the approaches

[2] Probabilistic Object Reconstruction with Online Global Model Correction.
Jack Hunt, *Victor Prisacariu*, *Stuart Golodetz*, *Tommaso Cavallari*, *Nicholas Lord*, *Philip Torr*.

presented in this work and the potential directions of relevant, future work. Finally, this work is concluded with some brief closing remarks.

Appendices .1 and .2 provide extra mathematical details and results.

2

LITERATURE REVIEW

This chapter provides a comprehensive survey of pertinent literature in the fields of Tracking and Mapping, Semantic SLAM, Dynamics in 3D Vision, Object Reconstruction and the prediction of Pose and Shape.

2.1 TRACKING AND MAPPING

There has been much research in the field of Tracking and Mapping (TAM) in recent years, with many large scale works being driven by the availability of once costly depth sensing equipment. The availability of such equipment combined with the ever increasingly parallel nature of modern Graphical Processing Units (GPU) has seen the field advance greatly beyond the seminal but compute resource limited works of it's infancy. This advancement is most predominant within the dense SLAM literature. This section shall first explore the earlier, fundamental works of this area of research, followed by an assessment of the current state of the art in dense SLAM.

Besl & McKay [11] introduced their seminal work on 3D shape registration in 1992, providing a method to estimate the full Six Degrees of Freedom (DoF) pose between 3D point sets. The authors present an Iterative Closest Point Algorithm that consists of

three operations per iteration; computation of closest point, computation of a 6DoF transformation and application of the transformation. The authors present a proof of convergence based on that of least squares minimisation, however there must be sufficiently complex geometry present in the structure of the data to converge to a meaningful transformation. The algorithm introduced is commonly known as Iterative Closes Points (ICP).

Complementary to the aforementioned works of *Besl & McKay* [11] in the foundational aspects of Dense SLAM is that of *Curless & Levoy* [12], 1996. The authors present an early volumetric integration framework for the reconstruction of shapes from range data obtained from a sensor such as a laser scanner. The authors introduce the Signed Distance Function (SDF), a volumetric, implicit shape representation in which entries are cumulatively updated in a weighted manner. Once observations have been integrated in to the SDF volume, an isosurface representation of the shape is extracted by a Marching Cubes [13] procedure. Though the approach may lead to gaps in the resultant model, the authors mitigate this by introducing a surface tessellation step.

Klein & Murray [14], in 2007 introduced the seminal, sparse TAM work *PTAM Parallel Tracking and Mapping*. The authors present an approach to the parallel estimation of camera pose and sparse scene mapping, making use of early parallel CPU hardware. The proposed scene representation is a map, consisting of point based features and multiple resolution key-frames. Pose estimation is achieved by the minimisation of the projection error between map points and the current live frame, following a coarse-to-fine strategy. The authors report per-frame running times of $\approx 20ms$, irrespective of increases in map size. However, it is reported in the work that failure cases exist; blurring of frames can impede feature detection, as can a lack of rigid geometry in the scene (due to the systems dependence on corner based features).

Later work by *Zhou et al* [15] in 2008 introduces an alternative shape representation to that of *Curless & Levoy* [12], based on the spatial KD-Tree data structure and a highly data parallel Breadth First Search (BFS) construction algorithm. The level of parallelism introduced allows for application to problems that require real time performance. The authors provide examples of use in ray tracing [16] and photon mapping [17].

In the same year, *Censi* [18] introduced Point-to-Line ICP, a variant of the ICP algorithm introduced in the work of *Besl & McKay* [11]. The presented approach utilises a point-to-line metric rather than a point-to-point metric and has a closed form solution in the planar case. For the non planar case the presented approach achieves quadratic convergence in a finite number of steps, utilising a normal weighting and a Lagrangian optimisation scheme. However, it is highlighted that prior to the optimisation procedure it is necessary to trim outliers from the point data sets as an additional preprocessing procedure.

In 2011, *Newcombe et al* [19] introduced an approach to dense TAM using monocular RGB images. Unlike the work of *Klein & Murray* [14], the authors do not make use of features extracted from the scene, but rather dense, textured depth maps. However, similar to the approach of *Klein & Murray* [14], the authors generate map components on a keyframe basis, which are used for the pose estimation phase within a photometric loss against a dense model. The authors report real time performance with commodity GPU acceleration. However, due to the reliance on a monocular RGB source, the authors report failure cases in the presence of illumination changes in the scene.

Culminating much of the aforementioned work, in 2011 *Newcombe et al* [1] also introduced the seminal *KinectFusion* pipeline, allowing for real time mapping of indoor scenes with the Kinect RGBD sensor from Microsoft. The authors utilise a sparser form of the SDF structure introduced by *Curless & Levoy* [12], the Truncated Signed Distance Function (TSDF), allowing for reconstruction at scene scale. For pose estimation, a multi-level variant of the ICP algorithm utilising a point-to-plane metric similar to that

of *PL-ICP* [18] is used. The pipeline consists of four phases; *measurement*, *integration*, *isosurface extraction* and *pose update*. Applications of the presented system however are limited only to those that require the reconstruction of static scenes; dynamic scenes are not supported by *KinectFusion*.

Further optimisations were made in 2013 by *Neißner et al* [20] to the *KinectFusion* pipeline proposed by *Newcombe et al* [1]. The authors introduce a spatially hashed TSDF data structure, in which the TSDF is split in to hashed blocks of voxels allowing for very fast voxel lookups. The presented approach yields low space and time complexity for such operations, vastly increasing the potential for real time, large scale use. Additionally, a streaming system is introduced to dynamically handle data transmission between the Central Processing Unit (CPU) and GPU, allowing for the reconstruction of scenes that may exceed the Graphical Random Access Memory (GRAM) bounds of commodity GPU's. The proposed system is capable of running at $\approx$ 46Hz on an NVIDIA Titan GPU.

In the same year, *Thomas et al* [21] introduced an alternative scene representation, based on the notion that a scene may be represented as a set of planar components with attributes such as surface normal vectors, confidences and Red-Green-Blue (RGB) colour. The motivation of the authors approach is that many common scenes that one might reconstruct are indoors and consist of components that are planar in nature, such as walls, floors and ceilings. Additionally, many planar objects are common, such as tables and cabinets. The authors present an alternative rendering approach based on quadrangulation [22] and utilise a *KinectFusion* [1] like ICP based algorithm for pose estimation.

Salas-Moreno et al [23] in 2013 also, introduced an alternative approach to that of the *KinectFusion* [1] like pipelines that, similarly to *Thomas et al* [21], utilises the prior information that many scenes consist of predictable, repeated structures. As such, the authors introduce a so called "Object Oriented" dense SLAM paradigm, in which the

reconstruction of the scene is split in to a graph of observed objects. Pose estimation is achieved by running an ICP based algorithm against renderings of the individual objects in the reconstructed scene model. Following pose estimation, the proposed system detects newly observed objects and inserts the appropriate object model in to the scene model. Consistency between scene components is enforced with pose graph optimisation, with re-localisation achieved in a similar manner. The proposed approach does however require a database of known objects a-priori.

Stückler et al [24] in 2014 introduced a non implicit, non volumetric representation based on multiple resolution Surfel [25] maps. The core data structure used for scene representation is a Voxel Octree [26], containing both Surfel's and probability distributions over appearance and shape. Pose estimation is achieved by optimising for a unit quaternion [27] and translation vector within a maximum likelihood framework, in which the energy function to be maximised is the likelihood of the RGBD observations given the accumulated probability distributions stored in the Octree. The presented pipeline also incorporates a randomised, graph and keyframe based loop closure component.

Following the approach of *Thomas et al* [21], *Salas-Moreno et al* [28] in 2014 introduced another reconstruction system that utilises the planarity property of many common scenes. The proposed approach focuses on the detection and modelling of planes in the scene, proceeding with their refinement over time. The proposed approach generates Surfel [25] Maps from observed RGBD frames, from which the planar regions are detected and integrated, filling holes in the reconstruction over time. The authors utilise an ICP algorithm to register the vertex maps of the RGBD observations and the reconstructed model. Additionally, re-localisation is achieved by the use of fern encoding [29] on key-frames.

Prisacariu et al [30, 31] in 2014 followed up the optimisations to the *KinectFusion* pipeline proposed by *Neißner et al* [20]. The authors presented, in addition to the

original publication, a technical report and an open source implementation. The proposed work provides further improvements to those of *Neißner et al* [20] including a number of low level optimisations to the core hashed TSDF data structure, it's allocation and update (integration of observation points) and the rendering phase of the pipeline. In addition, the authors demonstrate that pose estimation quality may be greatly improved by the use of commodity Inertial Motion Unit devices, commonly found on mobile phones and tablet computers. *Prisacariu et al* report running times of $\approx$ 47Hz on an NVIDIA Shield tablet and $\approx$ 910Hz with a commodity NVIDIA Titan X GPU.

Whelan et al [3] in 2015 proposed another *KinectFusion* [1] like pipeline intended to enable reconstruction of large scale scenes, achieving reconstruction over hundreds of metres. The approach taken by the authors to enable such large scale reconstructions is centred around the use of a cyclic buffer on the GPU. For pose estimation, the authors impose both geometric and photometric constraints on the camera pose. Additionally, the author's approach performs map updates in an as-rigid-as-possible [32] manner, combining frame recognition such that on a recognition event, a map update is performed. The proposed pipeline provides loop closure capabilities by utilising pose graph optimisation [33].

Zhou et al [34] also, in 2015, proposed another variant of the *KinectFusion* [1] pipeline proposed by *Newcombe et al*. The authors present improvements to the pose estimation phase of the pipeline, utilising contour cues to aid association and enforcing correspondence constraints on the estimated pose, with respect to scene geometry. Central to the presented approach is the depth image pre-processing steps of in-painting [35] regions of the depth image for which there are no depth measurements, followed by the aforementioned contour extraction stage.

The optimised pipeline proposed in 2014 by *Prisacariu et al* [30] was in 2016 improved with the addition of loop closure handling by *Kahler et al* [2]; *Kahler* being

one of the authors of the original 2014 contribution. Drift correction is achieved by the use of a multiple scene representation, with online alignment being performed periodically between the scenes. Corrections between the scenes are made via the use of Pose Graph Optimisation [33]. Loop closures are detected by the use of fern conservatories [29] as with the contributions of *Salas-Moreno et al* [23].

Many of the approaches and techniques evaluated in this section are foundational to the algorithms presented in later chapters. The fundamental geometric representations used in Chapters 3, 4 and 5 are variants of the volumetric representation introduced by *Curless & Levoy* [12]; the SDF. Additionally, a central theme in the work that follows is pose estimation, specifically utilising variants of the ICP algorithm, as introduced by *Besl & McKay* [11]. Later work on the *KinectFusion* pipeline and it's variants [30, 20], first introduced by *Newcombe et al* [1] builds on the aforementioned volumetric representation and pose estimation approaches to provide a full, modern pipeline for dense reconstruction. This pipeline, in turn, is foundational to the approaches taken in Chapters 3 and 4.

2.2 SEMANTIC SLAM

Over the years there has been much interest within the computer vision research community on the semantic understanding of our environment. The ability of machines to recognise and extract information about their environments and the components of them (such as people and objects) has wide application potential, ranging from autonomous robotics to AR video games. The application potential of this semantic scene understanding ability is amplified when it is combined with the vast progress that has been made in dense SLAM. This section shall provide a survey of research that amalgamates the two fields of semantic scene understanding and SLAM.

Civera et al [6] in 2011 introduced an approach to semantic SLAM that utilises image based features to attach semantic meaning to 3D observations. The SLAM system itself is based on Monocular Extended Kalman Filter SLAM [36], with semantics added to points via correspondences between Speeded Up Robust Features [37], extracted from the observed RGB frames and precomputed object descriptors. Consistency is then enforced by a geometric compatibility measure.

Stückler et al [38] in 2012 presented a semantic dense SLAM pipeline for the object centric integration of RGBD images. Given an RGBD frame, objects are detected using a Random Forest (RF) [39] classifier trained on hand crafted features extracted from RGBD images. The proposed approach does not reconstruct an entire scene, rather it reconstructs scene components (such as objects) that have been semantically segmented from the current RGBD frame.

Valentin et al [7] in 2015 proposed a fully integrated dense SLAM and semantic scene understanding pipeline with interaction being a primary focus. The proposed pipeline at it's core is based on that of *KinectFusion* [1], so requires the use of RGBD images and is restricted to the reconstruction of static scenes. Once a scene has been reconstructed, the author's pipeline allows users to interact with objects in the scene to provide training data for streaming RFs [40], which are used to detect and label parts of the rendered isosurface belonging to a given object class. Segmentations are refined using Variational Bayesian Mean Field Inference [41, 42]. The features extracted for this training process are Voxel Oriented Patch features, consisting of surface normal vectors and appearance information using the CIELab colour space.

Golodetz et al [8], in the same year, released an open source implementation of the pipeline proposed by *Valentin et al* [7], utilising the implementation of the *KinectFusion* [1] pipeline provided by *Prisacariu et al* [30]. The framework proposed by the authors extends that of *Valentin et al* [7] greatly, for example by supporting the use of

motion capture systems and VR headsets. In addition, the implementation provided is optimised to allow for real time use.

Handa et al [43], again in 2015 introduced an alternative, real time dense semantic SLAM pipeline. Much like the approaches of *Valentin et al* [7] and *Golodetz et al* [8], the proposed system is based on the *KinectFusion* [1] pipeline, with semantic scene understanding performed on the rendered isosurface. Contrary to previous approaches however, the authors make use of stacked Deep Autoencoders [44], trained on synthetic depth images a priori. As such, the proposed system makes use only of depth cues and may not be adapted to new object classes on an ad-hoc basis.

Cavallari et al [9], in the following year, presented another semantic dense SLAM pipeline built on top of the dense SLAM system presented by *Neißner et al* [20]. Much like the work of *Handa et al* [43], the proposed approach depends on a model pre-trained on a set of object classes. Unlike *Handa et al* [43], the authors make use of an Fully Connected Network [45], taking the Probability Mass Function (PMF) output to determine the class to be assigned to an isosurface region.

Later work by *McCormack et al* [46] in 2017, further integrates CNN based semantic information with dense SLAM. The authors make use of CNN generated semantic probability maps over a set of object classes when densely reconstructing a scene, using the approach of *Whelan et al* [47]. The instantaneous, pixel-wise distributions over class labels are fused into the scene model via the use of a Bayesian update procedure. The authors report a high degree of semantic accuracy with their approach, and highlight that due to the multi-view nature of the approach, an improvement on the 2D case is also achieved on the NYUv2 dataset.

Following thir aforementioned *2017* contribution, *McCormack et al* [48] introduce an object centric Dense SLAM system, leveraging recent advances in 2D semantic segmentation understanding [49] to augment the traditional Dense SLAM pipeline. The authors utilise 2D segmentation masks to spawn object centric TSDF volumes,

into which RGBD depth measurements are fused. Contrary to the traditional point-to-plane, ICP based tracking (and it's variants), the authors estimate pose over a graph of individual objects, including the handling of loop closure events. Spurious object instance detections are suppressed by maintaining an existence probability for each detection and it's corresponding TSDF volume.

The theme of dense 3D reconstruction with semantics is directly related to the research objectives outlined in Section 1.4. As outlined in this section, little research has investigated semantic learning *directly* on 3D geometry, though approaches such as that of *Handa et al* [43] work towards this. Prominent, prohibitive factors include the lack of readily available 3D data and the increased complexity of dealing with environments in which such capabilities are desirable, as outlined in Chapter 1. The contributions that follow in this work are intended to facilitate such research.

2.3 DYNAMIC & NON-RIGID SLAM, MOTION SEGMENTATION AND OPTICAL FLOW

Sections 2.1 and 2.2 provided an assessment of pertinent literature in the fields of SLAM and Semantic SLAM. However, all of the approaches outlined in these sections are limited to use in static scenes only without the capability to accurately operate in an environment that contains moving or deforming objects. This section shall explore pertinent literature on the topics of *Dynamic SLAM*, *Motion Segmentation* and *Optical Flow*. As such, the general focus of the work surveyed in this section is the detection, estimation and segmentation of motion in dynamic scenes.

Tsap et al [50], in 2000, presented an algorithm for non-rigid motion tracking of objects. The presented approach solves for dense motion vector fields between 3D objects by modelling motion with finite elements. The proposed system analyses differences between actual and predicted behaviour, using gradient descent to find a

set of optimal parameters for the non-linear Finite Element Model. Additionally, pose estimation is improved by using point correspondences.

Chen et al [51], in 2011, introduced a system to perform non-rigid motion tracking of the human body. The proposed system extracts and skins a surface mesh from multi-view video, after being fitted with a skeleton prior. To solve for non-rigid, articulated motion, the authors utilise a weighted, hierarchical ICP algorithm, where weightings are obtained by the Approximate Nearest Neighbour [52] algorithm.

In the following year, *Sun et al* [53] proposed an approach to motion estimation for objects in images. The proposed approach estimates optical flow in a layered manner, where each layer pertains to an object undergoing rigid body motion, with the number of layers being determined automatically. The authors utilise Maximum Flow [54] to solve a discretised flow field cost function for each layer, where object layers are a set of depth ordered Markov Random Fields (MRF) [55, 56].

Also in 2012, *Vicente & Agapito* [57] introduced an approach to the problem of non rigid reconstruction from monocular RGB video. In the proposed approach, the authors utilise an inextensibility constraint, such that the approach performs template-less reconstruction. The authors also cite the lack of post processing requirement as an advantage over related methods.

Unger et al [58], again in 2012, proposed an alternative system for the estimation of motion of objects undergoing rigid body motion in images. The authors present a variational formulation for motion estimation and segmentation with occlusion handling. As with the contributions of *Sun et al* [53], the authors utilise a parametric labelling of the flow field for each object undergoing motion, with labels encoded with an MRF Potts Model [59]. However, contrary to *Sun et al* [53] who utilise a Maximum Flow algorithm over the MRF models, *Unger et al* solved for flow and labels within a Primal-Dual [60] based optimisation framework.

In 2013, *Herbst et al* [61] proposed an extension to optical flow estimation to 3D scenes; the proposed system solves for Scene Flow based on RGBD data. The proposed approach is similar to that of *Brox et al* [62], with scene flow being formulated as a variational optimisation problem. The presented approach is a generalisation of the well established variational optical flow algorithm of *Brox et al* [62].

In the same year, *Stückler et al* [63] presented a framework for the segmentation of rigid body motion from RGBD data. The authors represent regions undergoing rigid body motion and their associated motion parameters as latent variables, with the resultant segmentations and parameters being solved for within an Expectation Maximisation [55, 56] framework. The presented approach is robust to both simultaneous foreground and background motion by giving each parity in the probabilistic model.

Though the motion estimation and segmentation approaches reviewed up to this point have not been within the SLAM framework, *Keller et al* [64] in 2013 introduced an RGBD based dense SLAM system capable of segmenting motion in a reconstructed scene. Unlike the *KinectFusion* [1] inspired dense SLAM pipelines, the presented approach does not utilise an implicit, volumetric representation. Rather, the authors opt for an explicit Surfel [25] based representation. Whilst performing live reconstruction, the proposed system detects and uses ICP outliers to determine dynamic scene components. With the information gained from detecting ICP outliers, the proposed system then propagates these detections by a flood fill operation. As the proposed approach utilises an explicit, flat data structure for scene representation, it does not have the advantages of it's highly optimised volumetric counterparts, such as that proposed by *Prisacariu et al* [65]. As such, scalability is limited.

In 2015, *Perera et al* [66] presented an approach to motion segmentation in TSDF volumes. Similar to it's planar counterparts presented by *Sun et al* [53] and *Unger et al* [58], the authors utilised a Markov network over the domain of interest. Motion segmentation is posed as a Maximum a Posteriori (MAP) [55, 56] inference problem

over a Conditional Random Field (CRF) [42] defined over TSDF voxels. The proposed system is able to segment objects undergoing both minor and major displacements, with motion labels and parameters found with respect to the live frame and the TSDF However, the proposed approach is limited only to very small scenes, with very long running times reported for TSDF volumes of dimensionality $256 \times 256 \times 256$.

Newcombe et al [10] again in 2015, introduced a dynamic dense SLAM system based on the earlier *KinectFusion* [10] pipeline, with the addition of the ability to handle non-rigidly deforming scenes. Non-rigid deformations are handled by the estimation of a 6DoF motion field that warps the model represented by the TSDF to the live frame. The solving of the warp field is achieved by the use of Dual Quaternion blending [67]. Though promising results are presented, there are limitations, such as lack of robustness to open/closed topology changes, such as hands. In addition, the authors highlight scalability issues.

The research outlined in this section is pertinent to the work presented in Chapter 3, which presents an approach to the handling of dynamics for dense reconstruction in real time, and with a volumetric representation. Though the aforementioned works of *Perera et al* [66] and *Newcombe et al* provide algorithms for handling dynamics in such representations, there remain complexity and scalability issues, the target of which is the focus of Chapter 3.

2.4 OBJECT RECONSTRUCTION

It is evident from Sections 2.1, 2.3 and 2.2 that much progress has been made in the fields of TAM/SLAM, dynamic SLAM and semantic SLAM. However, *Object Reconstruction* remains a very open and active field of research. As outlined in Section 2.4, cumulative errors in pose estimation are troublesome for the smaller scale (relative

to scene-scale), object centric SLAM. The combination of inherently less geometric information and potentially rapid, repetitive motion exacerbates the difficulties faced in scene-scale SLAM.

This section provides a review of pertinent literature on the task of reconstructing consistent models of objects, rather than full scale scenes. The problem of interest in this section, though related to SLAM, incurs additional complications with regards to pose estimation.

Curless & Levoy [12] as introduced in Section 2.1 presented a method of statically reconstructing shapes from range images taken from different viewpoints. However, the presented approach pre-dates many of the advances that have allowed for simultaneous tracking and mapping.

Kolev et al [68], in 2006, presented a probabilistic approach to 3D shape segmentation and recovery. Rather than the direct reconstruction approach taken by *Curless & Levoy* [12], the authors take the approach of inferring the most probable shape with respect to the observed image sequences. The shape to be inferred is encoded as a zero level set, extracted from a level set representation (such as an SDF). The level set of the shape is evolved over time within a variational framework, with respect to a volume of segmentation probabilities (foreground versus background). However, the proposed approach does not have an additional pose estimation phase and has only been evaluated on synthetic data of very polarised appearance.

Weise et al [69], in 2009, proposed an approach to the in hand scanning of 3D objects. The authors utilise an explicit point cloud representation of shape, rendered as Surfels [25]. Objects are rotated in front of a sensor with poses recovered by the use of an ICP like algorithm. During pose estimation, a topology graph is built which is used to offset drift in estimated poses in an as-rigid-as-possible [32] manner. However, the specification of object rotation in front of a sensor is suggestive of limited tracking

ability. In addition, the type of sensor is not specified, as such it is not clear what quality of sensing equipment is required to yield high quality results.

Llado et al [70] introduced in 2011 an approach to the 3D reconstruction of *Deformable* objects. The proposed approach makes use of an uncalibrated RGB stereo rig, requiring minimal *a-priori* setup. The approach to nonrigid reconstruction taken centers around the computation of a mean shape, to which live and reference frames are registered. From the registration to the mean shape, rigidly moving points may be identified. The final *deformed* model is recovered as a nonlinear optimisation problem.

Prisacariu et al [71] in *2012* proposed a probabilistic approach to the simultaneous tracking and segmentation of objects with *a priori known* 3D shape. Appearance based segmentation is performed in 2D, utilising Pixel Wise Posteriors (PWP) [72], with tracking performed in 3D. The authors demonstrate real time performance with the use of GPU hardware and propose a simple extension to the multiple object tracking case.

In 2013, *Garg et al* [73] introduced an approach to dense, ron-rigid surface reconstruction from monocular video. The authors present an approach to non-rigid SfM as a variational energy minimisation problem that does not require a shape prior. The author's report efficacy on the modelling of an instantaneous objects deformed state for a given frame.

Also in 2013, *Ren et al* [74] proposed an approach to the tracking and reconstruction of objects. Like *Kolev et al* [68], the authors utilise a probabilistic formulation based on the evolution of a level set representation. Initialised with a shape prior level set, the proposed approach evolves the shape prior with respect to observations. Crucially, unlike the approach of *Kolev et al* [68], the proposed approach simultaneously optimises for object pose. The proposed system works with RGBD data and segments the object of interest using PWP [72], as with the aforementioned work of *Prisacariu et al* [71]. It is

noteworthy however that there are performance limitations of the proposed approach and experiments show success for a limited set of target shapes.

Agudo et al [75] in 2014 proposed an approach to dense, non-rigid SfM. The authors outline an algorithm that is sequential in nature (rather than offline, batch processed) and takes as input a single monocular RGB stream. The approach taken is to model the mechanical dynamic behaviour of an objects surface over a rolling temporal window. The authors perform EM in a tractable manner by marginalising over the time dependent deformation parameters of the mechanical model. The proposed approach demonstrates efficacy for on-line, instantaneous recovery of a given objects mesh.

Later in 2015, *Dou et al* [76] present a system for the reconstruction of deformable objects using a Microsoft Kinect RGBD sensor. The proposed approach solves for a latent target shape and shape deformations by utilising bundle adjustment [77]. The authors report that loop closures are automatically detected, with errors incurred by drift being distributed backwards from the detection point. The resultant shape surface is extracted as a triangular mesh. The presented experiments demonstrate high quality reconstruction results, but with overnight run times indicating that it is not suitable for real time use.

Also in 2015, *Yu et al* [78] proposed a novel approach to non-rigid SfM using a monocular RGB video stream. The authors outline a shape template based algorithm, in which the template shape is constructed from a short rigid sequence (i.e an object of interest is not *non-rigidly deforming*). Once a template shape has been obtained, the algorithm proceeds to recover, for each frame, the deformed object model via an energy minimization over a photometric loss. Though the authors present impressive, high quality results, the requirement of a rigid sequence *a-priori* may limit the scope of applicability of the approach.

In the following year, *Gupta et al* [79] proposed a system for the reconstruction and segmentation of 3D objects from data obtained with an RGBD sensor. The

reconstructed object is represented implicitly within an SDF volume, but notably observations are integrated using the Softmax [56] function rather than weighted means as with *KinectFusion* [1]. Each voxel in the volume is assigned a label pertaining to it's membership of the object set, with objects refined utilising Graph Cuts [80] and Alpha Expansions [80]. The proposed approach utilises a photometric loss to optimise for object pose, with keyframe based loop closure detection. However, the authors report difficulties in building sufficiently granular reconstructions. In addition, the authors report drift in pose estimation to be problematic.

Also in 2016, *Agudo et al* [81] propose an approach to pose and 3D shape estimation of nonrigid and "potentially extensible" surfaces. As with previously outlined methods [50, 75], the authors model the deformation dynamics of the object to be reconstructed. In the presented approach, nonrigid dynamics are modelled by Navier's equations, which are then solved using FEM. The propsoed approach proves efficacious for the tasks of pose estimation and 3D shape recovery, however, performance is reported to be considerably below real time.

The contributions of Chapter 4 for the problem of object reconstruction outlined in Section 1.2 draw on the work of *Kolev et al* [68]. The probabilistic volumetric representation used for the authors level set evolution approach influences the formulation of object segmentation given in Chapter 4. Additionally, the work of *Ren et al* [74] provides a suitable base of comparison for the approach outlined later in this work. The approach of *Ren et al* performs simultaneous segmentation, pose estimation and reconstruction of objects from an RGBD image source, as is the case with the approach taken in Chapter 3.

2.5 SHAPE AND POSE PREDICTION

Section 2.4 provided a review of pertinent object reconstruction research, which demonstrates that although much progress has been made since the early work of *Curless & Levoy* [12], many open research problems remain. This section provides a survey of research into an alternative, inference driven approach to obtaining 3D models of observed objects, whereby rather than direct optimisation and integration being used for pose estimation and model building, the process is posed as a probabilistic inference procedure.

Prisacariu et al [65] in 2011 introduced an approach to shape prediction, segmentation and pose estimation. Shape is predicted from a hierarchy of generative Gaussian Process Latent Variable Models (GPLVM) [82], encoding a latent space embedding of common shape properties. Candidate shapes are generated as a one off regression in latent space, with a unified energy function optimised with respect to the shape latent space point and the object pose parameters.

In 2013, *Dame et al* [83] proposed an approach to dense object reconstruction from a monocular image source. Like the approach of *Prisacariu et al* [65], the authors utilise GPLVM's as shape priors to aid reconstruction and segmentation of the object of interest. Depth maps for the observed monocular sequence are optimised for within a Primal-Dual [60] framework, utilising Total Variation [84] regularisation. However, there is no pose estimation ability in the formulation, as poses are known a priori from PTAM [14].

In the following year, *Toshev et al* [85] proposed an approach to pose estimation utilising cascaded Deep Neural Network (DNN) [86] regressors. The authors utilise the DNN framework for the complex task of articulated human pose estimation.

Wohlhart et al [87], in 2015, presented an approach to the 3D detection and pose recovery of objects. The proposed approach utilises features extracted from a Convolutional Neural Network (CNN) [86] within a nearest neighbour cost function for object detection and recovery of rough pose. As such, the proposed approach poses the problem as a K-Nearest Neighbour [88] search in descriptor space. Object and pose are coupled in training (i.e. two similar cars with different poses will have spatially distant descriptors).

Chang et al [89], also in 2015, presented a large scale dataset of 3D shapes. The dataset can be used for a variety of 3D vision tasks due to the potential of modern machine learning techniques to learn rich latent space embeddings, as demonstrated by the approaches of *Prisacariu et al* [65] and *Dame et al* [83]. However, the shapes in the dataset are synthetic and as such have no depth sequences. Though, such sequences may be artificially rendered.

Also proposed in 2015 by *Rock et al* [90], is an approach to the recovery of complete 3D models from a single depth image of an object of interest. The input depth image is regressed into a database of a priori known objects by the use of an RF [39]. The matched shapes are coarsely matched to the input depth map, then later deformed at a higher granularity by a separate optimisation process.

Kendall et al [91], again in 2015, proposed a CNN approach to the regression of 6DoF camera pose from RGB input. The authors base their CNN architecture on that of *Szegedy et al* [92], with a depth of 23 layers. The authors report high levels of accuracy on indoor scenes, attributing this to the use of Transfer Learning [93] applied to classification models.

In 2017, *Zhou et al* [94] introduced an approach to object detection from 3D point clouds. Central to the proposed approach is an end-to-end trainable convolutional Region Proposal Network [95]. The authors evaluate the proposed approach on the

KITTI LIDAR [96] dataset, with input point clouds quantized into a voxel volume prior to training and prediction.

Gwak et al [97], also in 2017 introduced a Generalised Adversarial Network [98] like approach to shape prediction. The proposed network is trained in a weakly-supervised manner on silhouettes and 3D shapes with a log-barrier objective function. However, the applicability to "real world" scenarios is questionable, due to the synthetic nature of the data used to train and evaluate the network.

Grabner et al [99] in *2018* introduce a CNN based approach to the problem of simultaneous pose estimation and 3D shape retrieval. The authors use the estimated pose of a given object of interest as a shape prior for 3D model lookup. The authors render depth images of a given retrieved shape under the predicted pose for evaluation in a multi-view photometric loss for evaluation against learned image descriptors. The authors report impressive performance on *Pascal3D+* [100].

Pumarola et al [101], also in *2018* introduced an approach to the prediction of deformable shape surfaces from a single view. The outlined algorithm takes a two stage, CNN based approach consisting of detection and shape estimation, respectively. The approach is evaluated a synthetic dataset to which the authors artificially apply deformations and varying textures. The proposed approach demonstrates efficacy in both synthetic and non-synthetic data scenarios.

2.6 SUMMARY

From the evaluation of the literature given in Sections 2.1, 2.3, 2.2, 2.4 and 2.5, it is clear that much progress has been made in many areas of 3D computer vision, with a high level of commonality across the different domains. However, when assessed

within the context of the research objectives of this work, as outlined in Section 1.4, there is still much that can be contributed.

The literature reviewed in Sections 2.1 and 2.3 is directly related to the subject matter of Chapter 3, which approaches the problems faced when utilising dense SLAM techniques in an environment that is dynamic, rather than static, as with the approaches outlined in Section 2.1. Though it is evident from the literature assessment of Section 2.3 that much progress has been made, it is also evident from Section 2.3 that there exist limitations in current work. Section 3.1 introduces approaches to solving some of these limitations.

Section 2.4 reviewed literature pertinent to the object reconstruction subject matter of Chapter 4. Though it is clear that many advances have been made on representations and combined TAM for objects, Section 4.1 introduces an approach to the ongoing research problem around global model consistency. Additionally, the primarily 2D featured nature of the systems outlined in Section 2.2 provide a motivation for such approaches, as outlined in Sections 1.2 and 4.1.

The data driven works outlined in Section 2.5 provide a basis for the approach to shape and pose prediction presented in Chapter 5. Though much of the work reviewed in Section 2.5 addresses each of these problems in a *decoupled* way, may of the techniques are pertinent to the integrated approach taken in Chapter 5, as outlined in Section 5.1.

3

REAL TIME MOTION SEGMENTATION FOR DENSE VOLUMETRIC FUSION

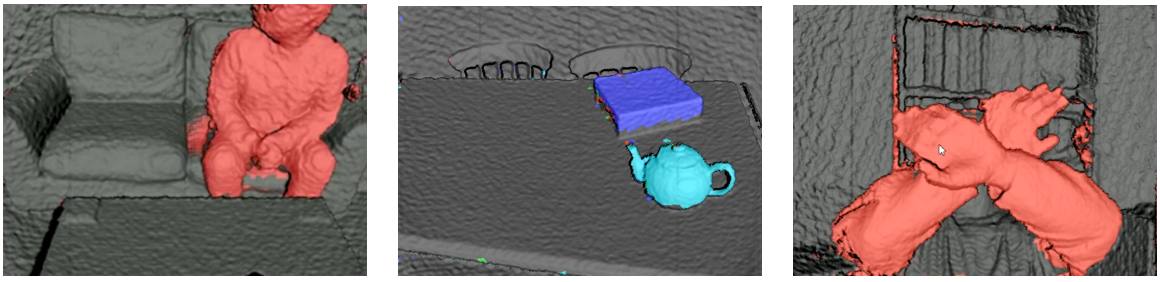
This chapter introduces an approach to motion segmentation and dense reconstruction in dynamic scenes with RGBD observations. The approach outlined in this chapter is capable of performing dense reconstruction in dynamic environments and segmenting objects undergoing motion. The approach presented in this chapter yields an improvement in pose estimation accuracy with respect to the standard KinectFusion-like pipeline. Furthermore, it is demonstrated that the segmentation of dynamic objects may be leveraged for 3D object recognition purposes.

3.1 INTRODUCTION

Progress in dense volumetric fusion has been accelerated in recent years with the availability of consumer 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 [1] allow one to build high quality, globally consistent scene models trivially, as outlined in Section 1.1 and Figure 1.2. Applications of such reconstruction pipelines however are limited due to the inability of such systems to

handle scenes in which there are dynamics; these 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 of the pipeline, resulting in model corruption ranging from noise in the reconstruction to spurious surface data due to erroneous pose estimation.

In this chapter, an approach to mitigating the problems present when performing dense volumetric reconstruction in dynamic scenes is presented. The basic SLAM pipeline on which this work is based is the *InfiniTAM* [30] variant of the *KinectFusion* [1] pipeline (InfiniTAM is also used as a base of comparison in Section 3.5). Central to the proposed, modified pipeline is the introduction of a dual scene representation based on the use of an implicit TSDF [12]. 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. Examples of motion segmentation are given in Figure 3.1.



(a) (b) (c)

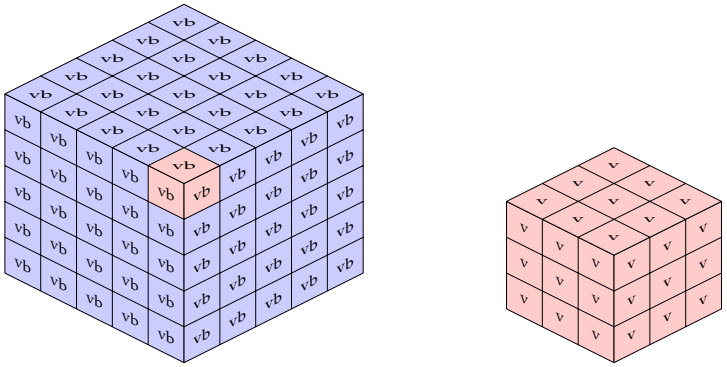
Figure 3.1: Examples of motion segmentation (note that red indicates motion):
(a) A person sits on a sofa.
(b) Dynamic objects that have become stable and have been identified by class.
(c) A person waving their arms in the scene.

Without the segmentation of dynamic scene components, when tracking against the current reconstruction the integrated dynamic components may cause artefacts

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 interference in the ICP pose estimation process can be mitigated. However, it should be noted that due to the rapid updating of the dynamic scene representation, the dynamic model does incur artefacts. This is due to the shorter TSDF truncation band of the dynamic model. However, empirically, these artefacts do not get fused into the static model as they do not gain a sufficient level of stability. Pose estimation is thus unaffected due to the static scene being used for ICP registration.

Once a part of the dynamic model has been stable for a sufficient period, its volumetric data is integrated in to the static model and 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 [20], 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. A graphical representation of the voxel block based structure of a given scene is presented in Figure 3.2.

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



Voxel Block Volume **Voxel Volume**

Figure 3.2: A graphical representation of an SDF/TSDF with it's voxels subdivided into voxel blocks. In the above, v represents voxel blocks and v voxels.

3.2 STATIC VOLUMETRIC FUSION

The volumetric fusion approach taken in this work draws on previous volumetric integration techniques [12, 1, 20, 30] and shall be introduced in this section as preliminary material, as it 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 ray-casting to prepare for tracking in the next frame. The static Fusion pipeline consists of the following three consecutive stages:

- 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.

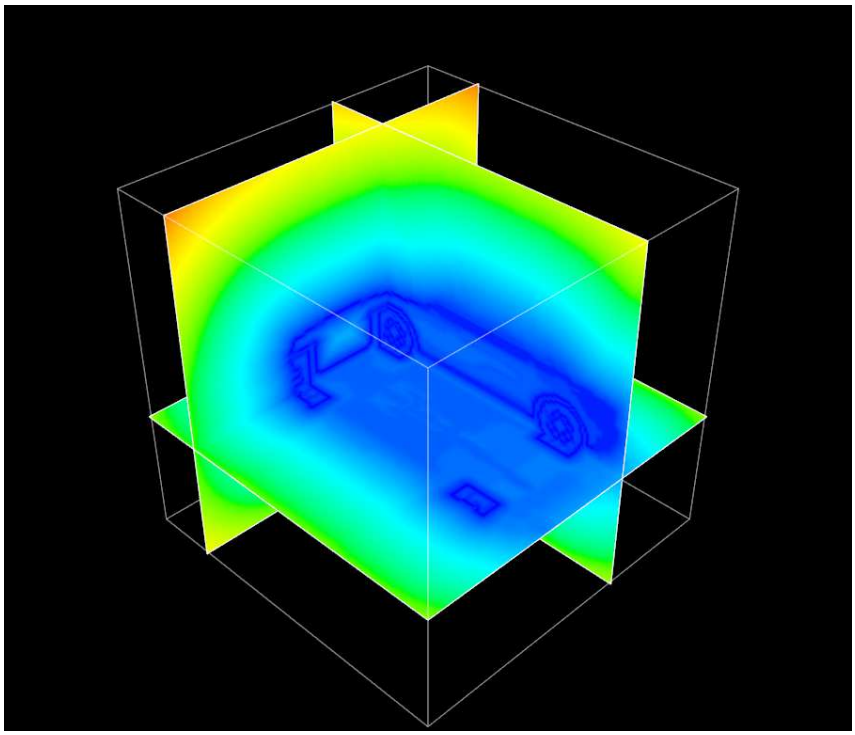


Figure 3.3: Slices of a car embedded within a three dimensional SDF.

For a given TSDF $\mathbf{\Phi}$, the Zero Level Set is defined as follows in Equation 3.1 where v denotes a TSDF voxel. An example of such an embedding is given in Figure 3.3.

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

To facilitate real time fusion, the InfiniTAM framework employs a voxel block hashing mechanism for fast access to scene voxels [20]. Within this context, voxel blocks are collections of $\mathbb{R}^{N \times N \times N}$ 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.

3.2.1 *Camera Pose Estimation*

As in previous works [1, 30], 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 non-linear least squares method [102]. 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 given in Equation 3.2, where $\mathbb{SO}(3)$ is the Special Orthogonal Group of Skew Symmetric Rotation Matrices.

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

The definition of Equation 3.2 has the alternative matrix form given in Equation 3.3.

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

Attitude Representation

The rotation matrix component of the transformation $\mathbf{T}$ is generated by a Rodriguez parameterization [103], 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, as shown in Figure 3.4.

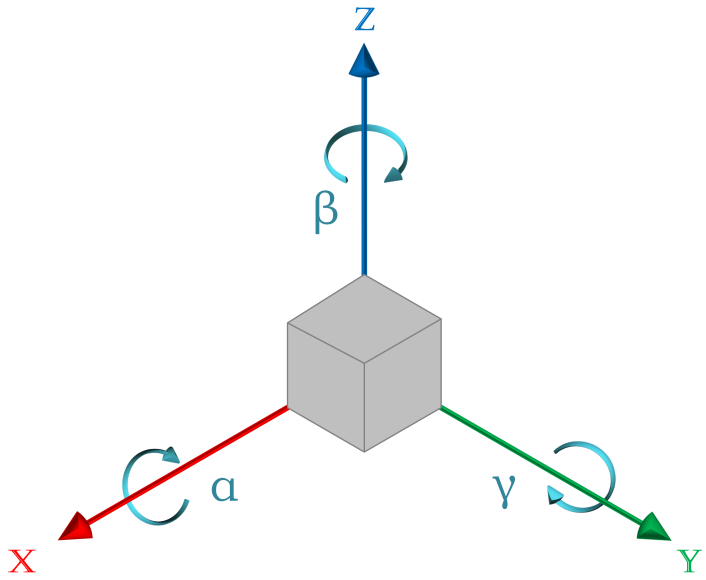


Figure 3.4: Right handed coordinate system and the three rotational axes; α, β and γ.

The Rodriguez parameters α, β and γ form a member of the Lie Algebra $\mathfrak{g}$ corresponding to the tangent space of the $\mathbb{SO}(3)$ Lie Group. Elements of the Lie Algebra map to the Lie Group by the matrix exponential. The formulation of the Rodriguez parameterization (performing the aforementioned matrix exponential) [103] is given by Equation 3.4, 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 in Equation 3.4, for an arbitrary vector $\boldsymbol{\nu}$, $\omega(\boldsymbol{\nu})$ is defined as the cross product matrix operator, as is given in Equation 3.5.

$$\omega(\boldsymbol{\nu}) = \begin{bmatrix} 0 & \nu_3 & -\nu_2 \\ -\nu_3 & 0 & \nu_1 \\ \nu_2 & -\nu_1 & 0 \end{bmatrix} \tag{3.5}$$

Evaluating Equation 3.4 leads to the form of the rotation matrix $\mathbf{R}$ of Equation 3.6.

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

The translational component $\mathbf{t}$ of the transformation $\mathbf{T}$ is given by the vector $\mathbf{t} \in \mathbb{R}^3$ of Equation 3.7, 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 point-to-plane energy minimisation problem of Equation 3.8.

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

In Equation 3.8, $\mathbf{R}$ 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 ray-casting from $\mathbf{\Omega}$ under the transformation $\mathbf{T}$. Finally, $\mathcal{N}$ is a normal map of $\mathbf{\Phi}$ and is defined as follows in Equation 3.9.

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

In Equation 3.9, $\nabla\mathbf{\Phi}$ is approximated by central finite differencing, as follows in Equation 3.10.

$$\nabla\mathbf{\Phi} = \begin{bmatrix} (2\delta v_x)^{-1} \\ (2\delta v_y)^{-1} \\ (2\delta v_z)^{-1} \end{bmatrix} \odot \begin{bmatrix} \mathbf{\Phi}(v_x + \delta v_x) - \mathbf{\Phi}(v_x - \delta v_x) \\ \mathbf{\Phi}(v_y + \delta v_y) - \mathbf{\Phi}(v_y - \delta v_y) \\ \mathbf{\Phi}(v_z + \delta v_z) - \mathbf{\Phi}(v_z - \delta v_z) \end{bmatrix} \tag{3.10}$$

For the gradient update phase of the ICP algorithm, the partial derivatives $\frac{\partial E}{\partial R_\lambda} \forall \lambda \in \{\alpha, \beta, \gamma\}$ may be derived as follows in Equation 3.12. As a first step, the following definition is made in Equation 3.11.

$$\phi(\mathbf{R}, \mathbf{t}, \mathbf{x}, \bar{\mathbf{x}}) = [\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{V}(\bar{\mathbf{x}})]^T \mathcal{N}(\bar{\mathbf{x}}) \tag{3.11}$$

With the substitution of Equation 3.11 in place, the derivation proceeds as follows in Equation 3.12.

$$\frac{\partial E}{\partial \lambda} = \frac{\partial}{\partial \lambda} \sum_{p \in \Omega} \|\phi(.)\|_2 \tag{3.12}$$

$$= \sum_{p \in \Omega} \frac{\partial}{\partial \lambda} \|\phi(.)\|_2 \tag{3.13}$$

$$= \sum_{p \in \Omega} \frac{\partial}{\partial \phi(.)} \|\phi(.)\|_2 \frac{\partial \phi(.)}{\partial \lambda} \tag{3.14}$$

$$= \sum_{p \in \Omega} \frac{1}{2} \frac{2\phi(.)}{\sqrt{\phi(.)^T \phi(.)}} \frac{\partial \phi(.)}{\partial \lambda} \tag{3.15}$$

$$= \sum_{p \in \Omega} \frac{\phi(.)}{\|\phi(.)\|_2} \frac{\partial \phi(.)}{\partial \lambda} \tag{3.16}$$

$$= \sum_{p \in \Omega} \frac{\phi(.)}{\|\phi(.)\|_2} \left[\frac{\partial}{\partial \lambda} \left[\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{V}(\bar{\mathbf{x}}) \right]^T \mathcal{N}(\bar{\mathbf{x}}) + \left[\mathbf{R}\mathbf{x} + \mathbf{t} - \mathcal{V}(\bar{\mathbf{x}}) \right]^T \frac{\partial}{\partial \lambda} \mathcal{N}(\bar{\mathbf{x}}) \right] \tag{3.17}$$

$$= \sum_{p \in \Omega} \frac{\phi(.)}{\|\phi(.)\|_2} \left[\frac{\partial \mathbf{R}}{\partial \lambda} \mathbf{x} \right]^T \mathcal{N}(\bar{\mathbf{x}}) \tag{3.18}$$

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 1, 2 and 3 respectively, of Appendix .1. The partial derivatives $\frac{\partial \mathbf{R}}{\partial \lambda} \forall \lambda \in \{\alpha, \beta, \gamma\}$ may be multiplied with $\mathbf{x}$ and combined in to the Rotation Jacobian of Equation 3.19.

$$\mathbf{J}_R = \left[\frac{\partial \mathbf{R}}{\partial \lambda}\mathbf{x}\right]^T\Bigg|_{\lambda=0} \tag{3.19}$$

$$= \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{3.20}$$

Note that the derivation for the partial derivatives $\frac{\partial E}{\partial \mathbf{t}_{\bar{\lambda}}} \forall \bar{\lambda} \in \{x, y, z\}$ is analogous with that of Equation 3.12, with the result given as follows in Equation 3.21.

$$\frac{\partial E}{\partial \mathbf{t}_{\bar{\lambda}}} = \sum_{p \in \mathbf{\Omega}} \frac{\phi(.)}{\|\phi(.)\|_2} \left[\frac{\partial \mathbf{t}}{\partial \bar{\lambda}}\right]^T \mathcal{N}(\bar{\mathbf{x}}) \tag{3.21}$$

The translational partial derivatives $\frac{\partial \mathbf{t}}{\partial \lambda} \forall \bar{\lambda} \in \{x, y, z\}$ may also be combined in to the following translation Jacobian as in Equation 3.19.

$$\mathbf{J}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

The overall combined Jacobian for the energy function defined in Equation 3.8, for use in pose optimisation is as follows in Equation 3.23.

$$\mathbf{J} = \frac{\phi(.)}{\|\phi(.)\|_2} \begin{bmatrix} \mathbf{J}_R & \mathbf{J}_t \end{bmatrix}^T \mathcal{N}(\bar{\mathbf{x}}) \tag{3.23}$$

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 [102] algorithm for solving non-linear least squares problems. The gradient update equation for the Levenberg-Marquardt algorithm is given in Equation 3.24.

$$\theta_{t+1} = \theta_t - (\mathbf{H} + \lambda \text{diag}(\mathbf{H}))^{-1}\mathbf{J} \tag{3.24}$$

In Equation 3.24 $\theta_t = [\alpha, \beta, \gamma, t_x, t_y, t_z]^T$ is the parameter vector of $\mathbf{T}$ at time t, $\mathbf{J}$ is the Jacobian introduced in Equation 3.23 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 $\epsilon \leqslant \epsilon_{old}$ **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 3.23
    $\mathbf{H} = \mathbf{J}^T\mathbf{J}$
    $\mathbf{C} = \text{chol}(\mathbf{H} + \lambda \text{diag}(\mathbf{H}))$
    $\delta = \text{backsub}(\mathbf{C}, \mathbf{J})$
    $\theta_{tmp} \leftarrow \theta_{tmp} - \delta$
  end while return θ_t
end procedure

Note that for numerical stability, the matrix inverse in Equation 3.24 may be avoided by utilising the chol and backsub routines to compute the Cholesky Decomposition [102] and solve a linear system with back-substitution [104], respectively. Additionally, it should be noted that the cost term of Equation 3.8 and jacobian term of Equation 3.23 may be evaluated and reduced on the GPU, as there exists no spatial data dependence between either the model points or depth map points.

3.2.2 *Volumetric Integration*

The second phase in the scene reconstruction pipeline is volumetric integration. That is, the integration of observed depth images into 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 camera tracking at the next time step.

As previously outlined, the TSDF is 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.

As with KinectFusion [1] 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 transformation given in Equation 3.25, noting that $\mathbf{x}_v$ is in homogeneous form.

$$\mathbf{x}_\Omega = \mathbf{K}\mathbf{T}_i^{-1}\mathbf{x}_v \tag{3.25}$$

In Equation 3.25, $\mathbf{K}$ 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 form of the camera intrinsic calibration is as follows in Equation 3.26.

$$\mathbf{K} = \begin{bmatrix} f_x & s & x_0 & 0 \\ 0 & f_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.26}$$

In the calibration matrix of Equation 3.26, f_x, f_y, s, x_0 and y_0 are the focal lengths, scale and camera principal points respectively.

The integration of new data points into the TSDF volume is achieved by computing running means; each voxel contains a running average of its SDF value over time. Projecting to the depth image $\mathbf{\Omega}$ coordinates as in Equation 3.25 to perform a depth

pixel 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 in Equation 3.27.

$$\eta = \mathbf{x}_\Omega - \mathbf{x}_v^z \tag{3.27}$$

If $\eta \geqslant -\mu$, then the depth of the point is not beyond the truncation band of the TSDF (behind the isosurface), where μ is half the width of the truncation band, then the TSDF depth measurement update proceeds as follows in Equation 3.28 for a voxel $\mathbf{x}_v \in \mathbf{\Phi}$. A graphical depiction of the 2D case of a TSDF truncation region is given in Figure 3.5.

$$\mathbf{\Phi}(\mathbf{x}_v)_t = \frac{1}{\phi(\mathbf{x}_v)_{t-1} + 1}\left[\phi(\mathbf{x}_v)_{t-1}\mathbf{\Phi}(\mathbf{x}_v)_{t-1} + \min\left(1, \frac{\eta}{\mu}\right)\right] \tag{3.28}$$

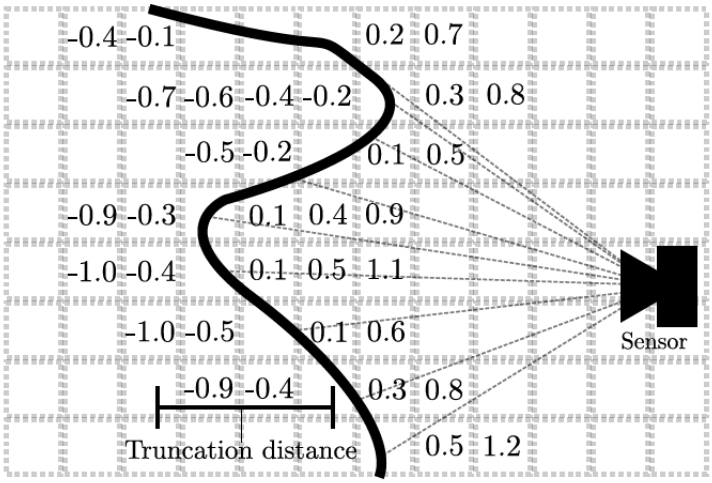


Figure 3.5: A 2D TSDF with it's truncation region specified.[1]

In addition, the voxel weight $\phi(\mathbf{x}_v)$ update is as follows in Equation 3.29.

$$\phi(\mathbf{x}_v)_t = \phi(\mathbf{x}_v)_{t-1} + 1 \tag{3.29}$$

[1] Image copyright: *Whelan et al* [3].

3.2.3 *Rendering*

Following the integration process outlined in Section 3.2.2 is the rendering stage in which an image of the scene under the current pose is generated, to provide an updated rendering for the tracking stage outlined in Section 3.2.1, at the next time step.

Rendering in the pipeline is achieved by Ray-casting [105], the process of "casting" a ray from the camera frame into the volume representation of the scene, to find intersections with the scene isosurface. The basic Ray-casting process is Given in the following Algorithm.

Algorithm 2 Volume Raycasting.

procedure RAYCAST($\Phi, \Omega_{in}, \Omega_{out}, \mathbf{C}, \mathbf{P}, z_{min}, z_{max}, D$)
  for $y \leftarrow 0$ to H **do**
    for $x \leftarrow 0$ to W **do**
      $\mathbf{p}_{min} \leftarrow \begin{bmatrix} \frac{x-\mathbf{C}_{0,2}}{\mathbf{C}_{0,0}} & \frac{y-\mathbf{C}_{1,2}}{\mathbf{C}_{1,1}} & z_{min} & 1 \end{bmatrix}^T$ ▷ Unproject
      $\mathbf{p}_{max} \leftarrow \begin{bmatrix} \frac{x-\mathbf{C}_{0,2}}{\mathbf{C}_{0,0}} & \frac{y-\mathbf{C}_{1,2}}{\mathbf{C}_{1,1}} & z_{max} & 1 \end{bmatrix}^T$
      $\mathbf{p}^\star_{min} \leftarrow \mathbf{P}^{-1}\mathbf{p}_{min}$ ▷ Object Point
      $\mathbf{p}^\star_{max} \leftarrow \mathbf{P}^{-1}\mathbf{p}_{max}$
      $\mathbf{p}^\star_{min} \leftarrow D\frac{\mathbf{p}^\star_{min}}{\mathbf{p}^\star_{min,3}} + \eta$
      $\mathbf{p}^\star_{max} \leftarrow D\frac{\mathbf{p}^\star_{max}}{\mathbf{p}^\star_{max,3}} + \eta$
      $\mathbf{s} \leftarrow \mathbf{p}^\star_{max} - \mathbf{p}^\star_{min}$ ▷ Ray Step
      $\mathbf{s}^\star \leftarrow \frac{\mathbf{s}}{\|\mathbf{s}\|}$
      $\mathbf{p}_m \leftarrow \mathbf{p}^\star_{min} - \mathbf{s}^\star$ ▷ Starting Point
      for $\delta \leftarrow 0$ to $\|\mathbf{s}\|$ **do** ▷ Traverse the Ray
        if $\Phi(\mathbf{p}_m)$ is valid **then** ▷ Write Shaded Pixel
          $\phi \leftarrow \Phi(\mathbf{p}_m)$
          $\mathbf{x} \leftarrow \mathbf{P}(\frac{\mathbf{p}_m}{D} - \eta)$
          $\mathbf{x}^\star \leftarrow \frac{\mathbf{x}}{\mathbf{x}_3}$
          $\Omega_{x,y} \leftarrow \mathbf{x}^\star_2$
          break
        end if
        $\mathbf{p}_m \leftarrow \mathbf{p}_m + \mathbf{s}^\star$ ▷ Increment Model Point
      end for
    end for
  end for
end procedure

It should be noted that the loop over image pixels $\Omega_{x,y}$ can be trivially optimised in a data parallel fashion on a GPU. This is possible as there exists no data dependence between image pixels and TSDF accesses are read only.

3.3 VOLUMETRIC FUSION WITH DYNAMIC SCENES

The conventional approach described in Section 3.2 is adapted 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 given 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 proposed pipeline is given in Figure 3.6.

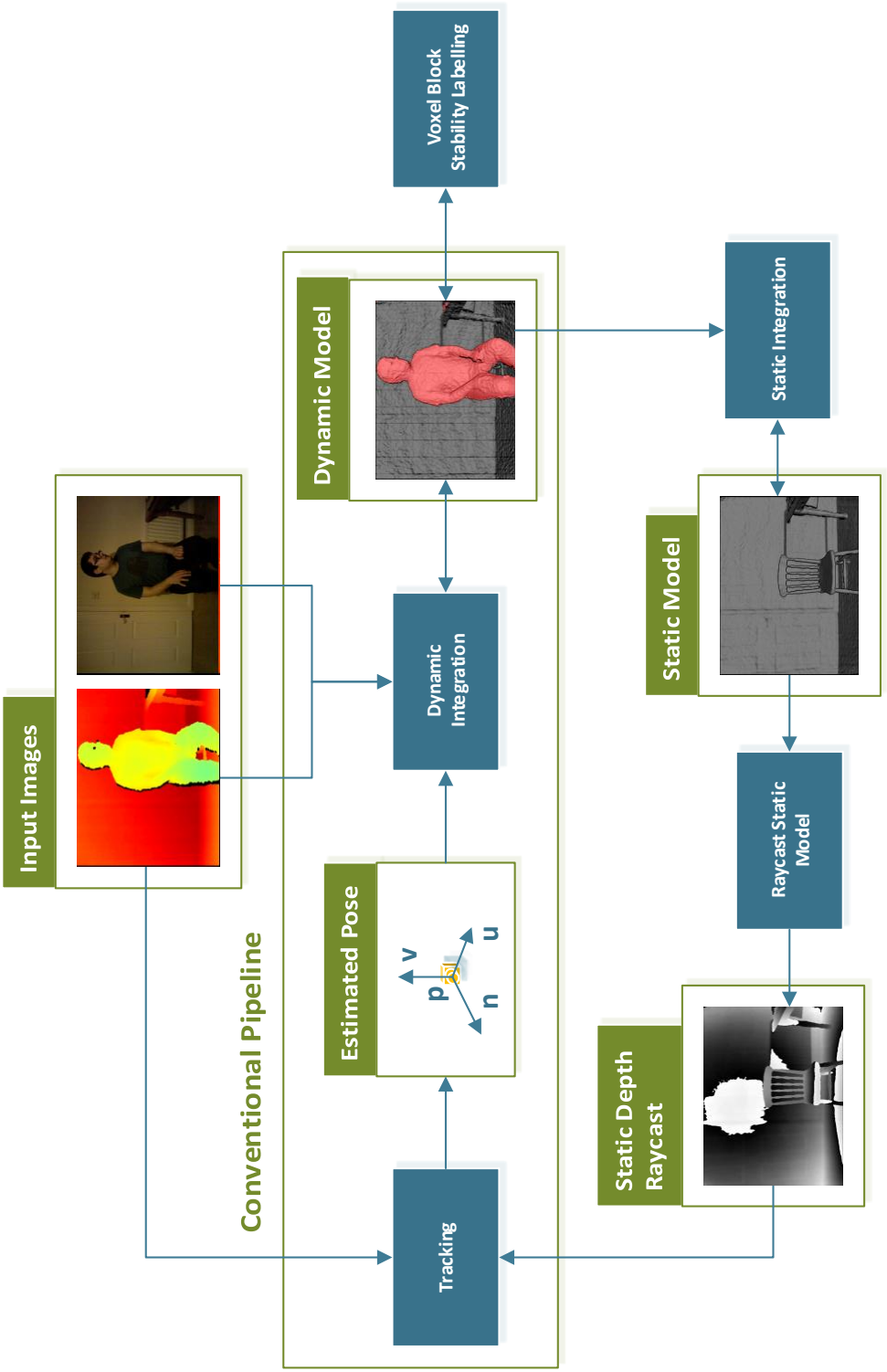


Figure 3.6: The proposed motion segmentation pipeline. Note the additional voxel block labelling stage with a feedback to the model integration stage.

3.3.1 *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 and fused textures for the voxels in a given voxel block (visible in the current view frustum under the current pose) have remained sufficiently similar over time.

The stability value for each voxel block in the scene is initialised to nought. At each frame, the instantaneous TSDF values and textures 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 in Equation 3.30.

$$\tau_{\mathcal{V}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \left[\left| \mathbf{\Phi}^{d}(v) - \min\left(1, \frac{\eta}{\mu}\right) \right| + \left| \mathbf{\Phi}^{d}_{rgb}(v) - \boldsymbol{\omega} \right| \right] \tag{3.30}$$

Note that in Equation 3.30, $\boldsymbol{\omega}$ is the RGB pixel corresponding to the depth pixel from which the depth raycast was performed. Additionally, $\mathbf{\Phi}^{d}_{rgb}(v)$ is the RGB value of the voxel v in the dynamic scene model.

The stability label $l_{\mathcal{V}} \in \{\text{stable}, \text{unstable}\}$ for a given voxel block $\mathcal{V}$ is determined by thresholding on ξ, as follows in Equation 3.31. Note that the voxel block score computed in Equation 3.30 are min-max normalised block wise to the interval $[0, 1]$ prior to the evaluation of Equation 3.31.

$$l_{\mathcal{V}} = \begin{cases} \text{stable} & \textbf{if } \tau \leqslant \xi \\ \text{unstable} & \textbf{if } \tau > \xi \end{cases} \tag{3.31}$$

If the label l_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_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 outlined in Equation 3.32.

$$\tau_v = \begin{cases} \tau_v + 1 & \textbf{if } l_v = \text{stable} \\ 0 & \textbf{if } l_v = \text{unstable} \end{cases} \tag{3.32}$$

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 3.3.2.

3.3.2 *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 of it's voxel TSDF values integrated into the static model. In a similar formulation to that given in Equation 3.28 of Section 3.2.2, 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 in Equation 3.33.

$$\phi_t^s(v) = \phi(v)_{t-1}^s(v) + \phi(\bar{v})_{t-1}^d(\bar{v}) \tag{3.33}$$

Similarly, the update for the TSDF values is as follows in Equation 3.34.

$$\mathbf{\Phi}_t^s(v) = \frac{1}{\phi_t^s(v)} \left[\phi_{t-1}^s(v) \mathbf{\Phi}_{t-1}^s(v) + \phi_{t-1}^d(\bar{v}) \mathbf{\Phi}_{t-1}^d(\bar{v}) \right] \tag{3.34}$$

Removal of Static Blocks Undergoing Motion

In the case that a voxel block that has been integrated into the static model is detected to be undergoing motion in the dynamic model, it is simply removed from the static model. Due to the 1 : 1 correspondence between the static and dynamic scenes, this removal is trivial and computationally inexpensive.

Following the removal of a previously static voxel block from the static scene model, any depth observations that have been integrated into the corresponding space in the dynamic model must follow the integration procedure of Equations 3.33 and 3.34 once labelled as static, as these quanitites are reset to their initial state. The removal of a voxel block from the static model excludes it from the following ICP pose estimation procedure.

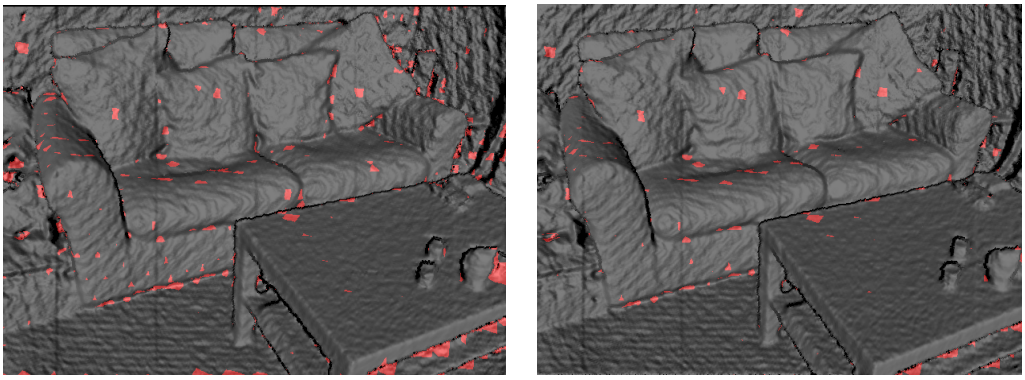
CRF Refinement of Voxel Block Scores

The outlined motion segmentation and dynamic SLAM pipeline may also be augmented with an additional 2D segmentation step. Given the normalised voxel block wise stability scores (refer to Section 3.3.1), the corresponding stability labels may be refined in image space and unprojected back into 3D scene space.

During the raycast procedure outlined previously, a raster image of voxel block indices is generated, such that each pixel contains the lookup index for the voxel block for which it's ray intersects. Thus, a 2D map of unary potentials is generated based upon the stability scores for the voxel blocks in the view frustum.

With known extrinsic calibration between the depth and RGB sensors, the energy function of Equation 3.35, over voxel block stability scores may be defined.

$$E(\mathbf{\Phi}^{d}(\mathbf{v})) = P(\mathbf{v}) + \sum_{\mathbf{u} \in \mathcal{V}} P(\mathbf{\Omega}^{d}(\mathbf{v}) \,|\, \mathbf{\Omega}^{d}(\mathbf{u})) \tag{3.35}$$



(a) (b)

Figure 3.7: Denoising of stability scores.
(a) Without CRF refinement.
(b) With CRF refinement.

In Equation 3.35, $\boldsymbol{\nu}$ is the raster index of the voxel block for which the energy calculation is to be computed, with respect to neighbouring raster locations $\mathbf{u} \in \mathcal{V}$, for some set of neighbouring locations $\mathcal{V}$. The unary term $\mathrm{P}(\boldsymbol{\Phi}^{\mathrm{d}}(\boldsymbol{\nu}))$ is proportional to the normalised voxel block score $\boldsymbol{\tau}$ given by Equation 3.30 and the pairwise term $\mathrm{P}(\boldsymbol{\Omega}^{\mathrm{d}}(\boldsymbol{\nu}) \,|\, \boldsymbol{\Omega}^{\mathrm{d}}(\mathbf{u}))$ is given by a standard Gaussian density $\mathcal{N}(\boldsymbol{\Omega}^{\mathrm{d}}(\boldsymbol{\nu}) \,|\, \boldsymbol{\Omega}^{\mathrm{d}}(\mathbf{u}), \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is given by the texture variance of the voxel block corresponding to $\boldsymbol{\nu}$.

Following CRF refinement, the voxel block scores may be directly updated via the aformentioned index image. This process is carried out prior to the labelling of Equation 3.31. Empirically, although this CRF refinement procedure does have a denoising effect on the stable scene rendering, it appears to have minimal impact on pose estimation quality. Inference over the CRF outlined in this section is performed via the efficient, filter based message passing algorithm of *Kr'ahenb'uhl* [42].

3.3.3 *Pipeline Summary*

With the central components of the proposed algorithm now outlined, an algorithmic summary may be given. As has been highlighted, the central components of the

approach are the dual volume representation, stability labelling and inter-volume surface integration. The process is given in the following algorithm.

Algorithm 3 Motion Segmentation and Dynamic SLAM

procedure MoSeg Iteration($\mathbf{\Phi}_s, \mathbf{\Phi}_d, \mathbf{\Omega}, \mathbf{T}, t$)
  if $t \neq 0$ **then**
    $\mathcal{R} \leftarrow$ raycast($\mathbf{\Phi}_s, \mathbf{\Omega}, \mathbf{T}$) ▷ Section 3.2.3
    $\mathbf{T}_{t+1} \leftarrow$ estimatePose($\mathcal{R}, \mathcal{D}$) ▷ Section 3.2.1
    updatePose($\mathbf{T}_{t+1}$)
  end if
  $\mathcal{D} \leftarrow$ unproject($\mathbf{\Omega}_d$)
  $\mathbf{\Phi}_d \leftarrow$ integrate($\mathcal{D}, \mathbf{\Phi}_d, \mathbf{T}$) ▷ Equation 3.28
  for $p \in \mathcal{D}$ **do**
    $b \leftarrow$ getVoxelBlock(p)

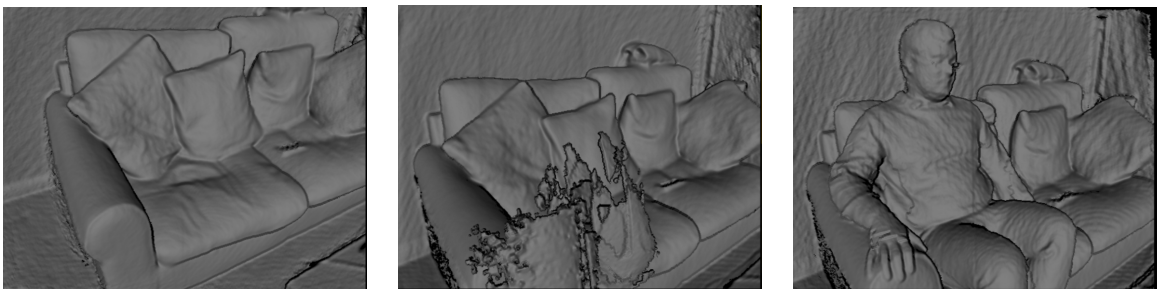
    $\tau_b \leftarrow$ getStability(b) ▷ Equation 3.30
    $l_b \leftarrow$ getLabel(τ_b) ▷ Equation 3.31
    updateStability(b, l_b) ▷ Equation 3.32

    if $t \leqslant$ initial_frame_count $\vee\, l_b ==$ stable **then**
      for $v \in$ getVoxels(b) **do**
        $\phi_s(v) \leftarrow$ getWeight(v)
        $\phi_d(v) \leftarrow$ getWeight(v)
        $\mathbf{\Phi}_s(v) \leftarrow$ integrate($\mathbf{\Phi}_s(v), \mathbf{\Phi}_d(v)), \phi_s(v), \phi_d(v)$ ▷ Equation 3.34
        updateStableWeight($\phi_s(v), \phi_d(v)$) ▷ Equation 3.33
      end for
    end if
  end for
  raycast($(\mathbf{\Phi})_d, \mathbf{T}$) ▷ Visualise Dynamics
end procedure

The presented algorithm is trivially parallelisable with commodity GPU hardware. The integration of points in the depth map $\mathcal{D}$ is parallelisable with minimal risk of race conditions occurring. However care must be taken for edge cases where multiple simultaneous writes may occur at the same TSDF location. However, the integration of TSDF data from the dynamic model to the static model has no such risk and as such is trivially parallelisable.

3.4 QUALITATIVE RESULTS

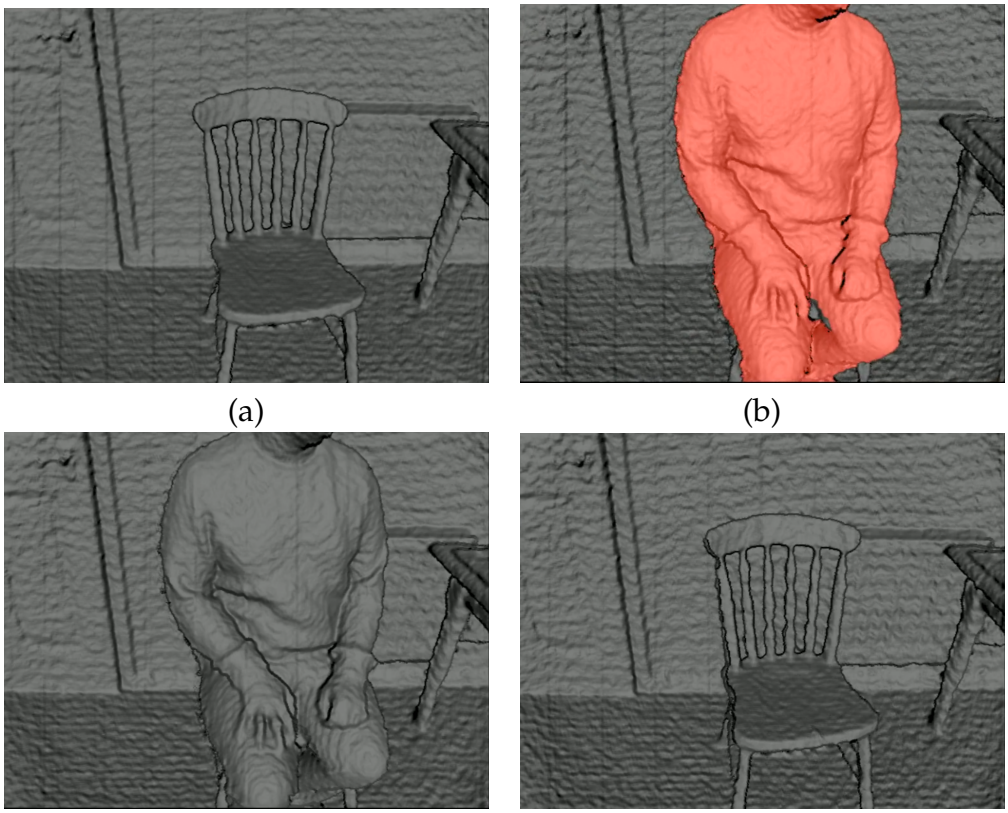
Empirically, the proposed motion segmentation system is capable of retaining globally consistent tracking within the dense SLAM framework for a range of scenarios that would prove to be problematic for other, static dense SLAM systems. The experiments performed demonstrate a robustness to dynamics in 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 frustum and sitting on a sofa is given in Figure 3.8.



(a) (b) (c)

Figure 3.8: A qualitative comparison between the proposed system and InfiniTAM [30].
(a) A static scene containing a sofa 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.

In addition to the ability to reconstruct a moving object that becomes static in the scene as shown in Figure 3.8, the proposed system is also capable of segmenting dynamic objects in a scene that are undergoing non-rigid motion. Whilst these objects are segmented and labelled as "dynamic" they are not used for camera pose estimation. An example of this behaviour may be observed in Figure 3.9.



(a) (b)

(c) (d)

Figure 3.9: 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.

3.5 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 TUM²RGBD dataset [106] 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. It should be noted that both the approach outlined in this chapter *and* the baseline implementation (InfiniTAM) utilise a simple ICP outlier rejection routine to improve tracking performance.

TUM Standard Sequence Name	*MoSeg* ATE (m)	*ITM* ATE (m)	*EF* ATE (m)
fr2-desk-with-person	**0.131**	0.297	NA
fr3-sitting-static	0.013	0.012	**0.009**
fr3-sitting-xyz	0.068	0.053	**0.026**
fr3-sitting-halfsphere	0.141	**0.115**	0.138
fr3-sitting-rpy	**0.052**	0.081	NA
fr3-walking-static	0.272	0.999	**0.062**
fr3-walking-xyz	0.373	0.544	**0.216**
fr3-walking-halfsphere	0.544	0.762	**0.209**
fr3-walking-rpy	**0.547**	0.843	NA

Table 3.1: The Absolute Trajectory Error (ATE) results (in metres) achieved by the proposed approach in comparison to the baseline InfiniTAM [30] framework on a variety of the standard sequences from the TUM RGBD dataset [106]. The lower ATE result on each sequence is highlighted in bold. Additionally, ATE scores for ElasticFusion [47] are also provided.

[2] Technical University of Munich, RGB-D SLAM Dataset and Benchmark. https://vision.in.tum.de/data/datasets/rgbd-dataset

Given in Table 3.1 is the Absolute Trajectory Error (ATE) measures for each of the TUM Dynamic Scenes. The ATE utilises the method of Horn [107] 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 3.1 are visualised in Figure 3.10. For completeness, the performance of the current state of the art *explicit* dense SLAM system *ElasticFusion* [47] is also provided. Note that *ElasticFusion* is not volumetric in nature (rather, using an explicit, surfel representation), so is not directly comparable with the approach outlined in this work.

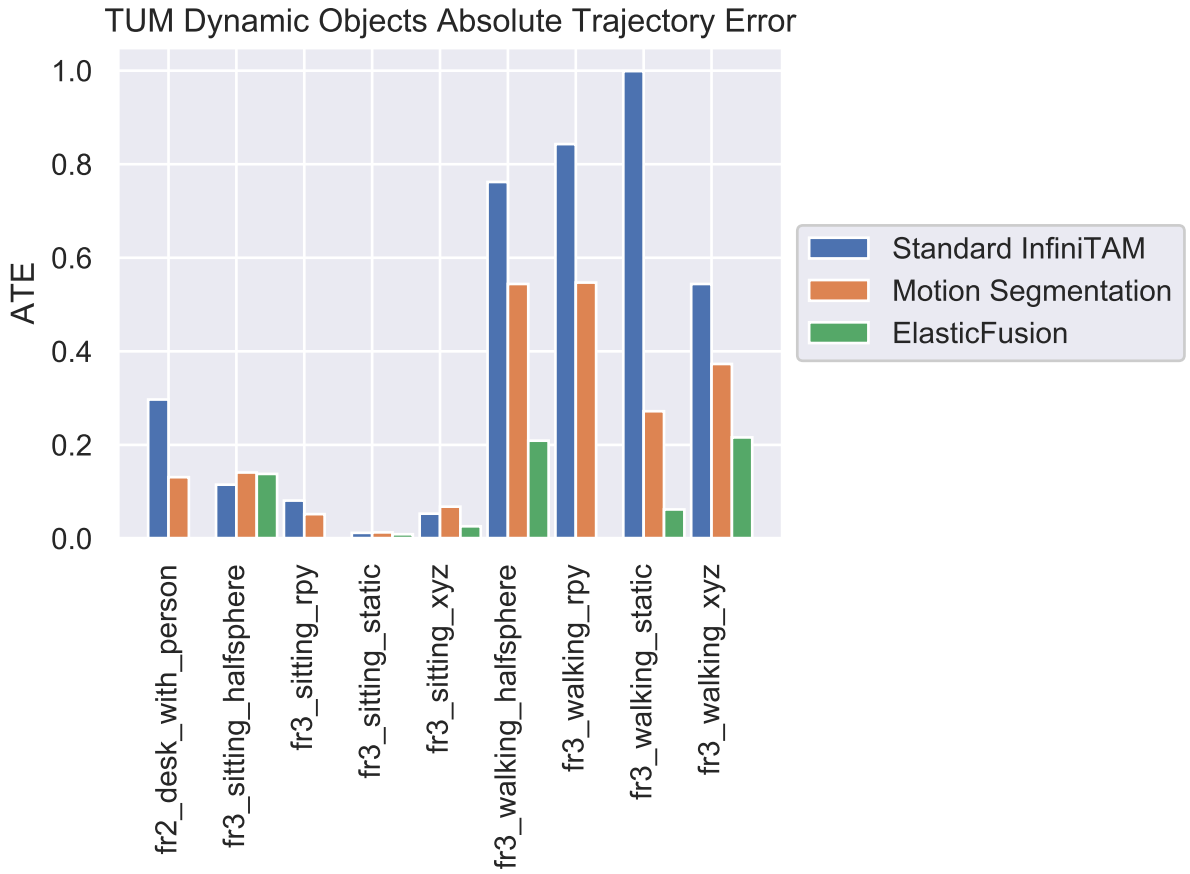


Figure 3.10: Absolute Trajectory Error (ATE) for the TUM Dynamic Scenes dataset.

In addition to evaluating quantitatively in terms of Absolute Trajectory Error, additional results are provided in terms of Relative Trajectory Error (RTE) [106]. RTE measures the relative pose error over a fixed time interval, with the final score being the Root Mean Squared Error (RMSE) over all time windows. RTE results are provided in Table 3.2 and visualised in Figure 3.11.

TUM Standard Sequence Name	*MoSeg* RTE (m)	*ITM* RTE (m)	EF (m)
fr2-desk-with-person	**0.024**	0.026	NA
fr3-sitting-static	0.011	**0.010**	**0.010**
fr3-sitting-xyz	0.031	**0.028**	**0.028**
fr3-sitting-halfsphere	**0.032**	**0.032**	0.102
fr3-sitting-rpy	0.071	**0.067**	NA
fr3-walking-static	0.077	0.163	**0.058**
fr3-walking-xyz	0.406	0.300	**0.214**
fr3-walking-halfsphere	0.252	0.305	**0.163**
fr3-walking-rpy	0.469	**0.406**	NA

Table 3.2: The Relative Trajectory Error (RTE) results (in metres) achieved by the proposed approach in comparison to the baseline InfiniTAM [30] framework on a variety of the standard sequences from the TUM RGBD dataset [106]. The lower ATE result on each sequence is highlighted in bold. As with Figure 3.10, ElasticFusion results are also provided for completeness.

3.6 PERFORMANCE EVALUATION

As outlined in the research objectives of Section 1.4, the proposed dynamic SLAM pipeline is required to run in real-time if it is to be suitable for use at scale. The basic dense SLAM pipeline of *Prisacariu et al* [30] is a highly optimised implementation of the voxel hashing approach of *Neißner et al* [20], achieving very high frame-rates on consumer computing equipment. For the purpose of evaluating the performance of the modified pipeline outlined in this work, a direct comparison is drawn to *InfiniTAM*, the implementation of *Prisacariu et al* [30].

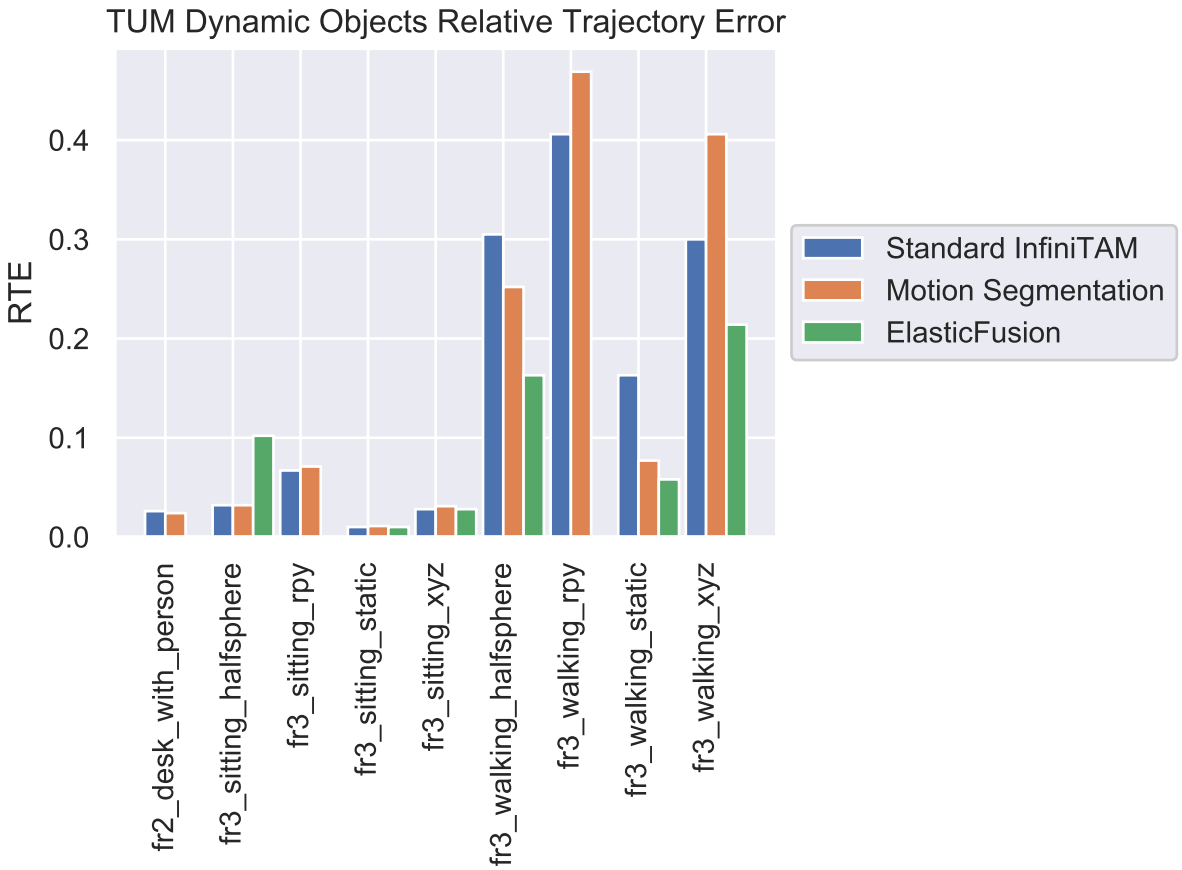


Figure 3.11: Relative Trajectory Error (RTE) for the TUM Dynamic Scenes dataset.

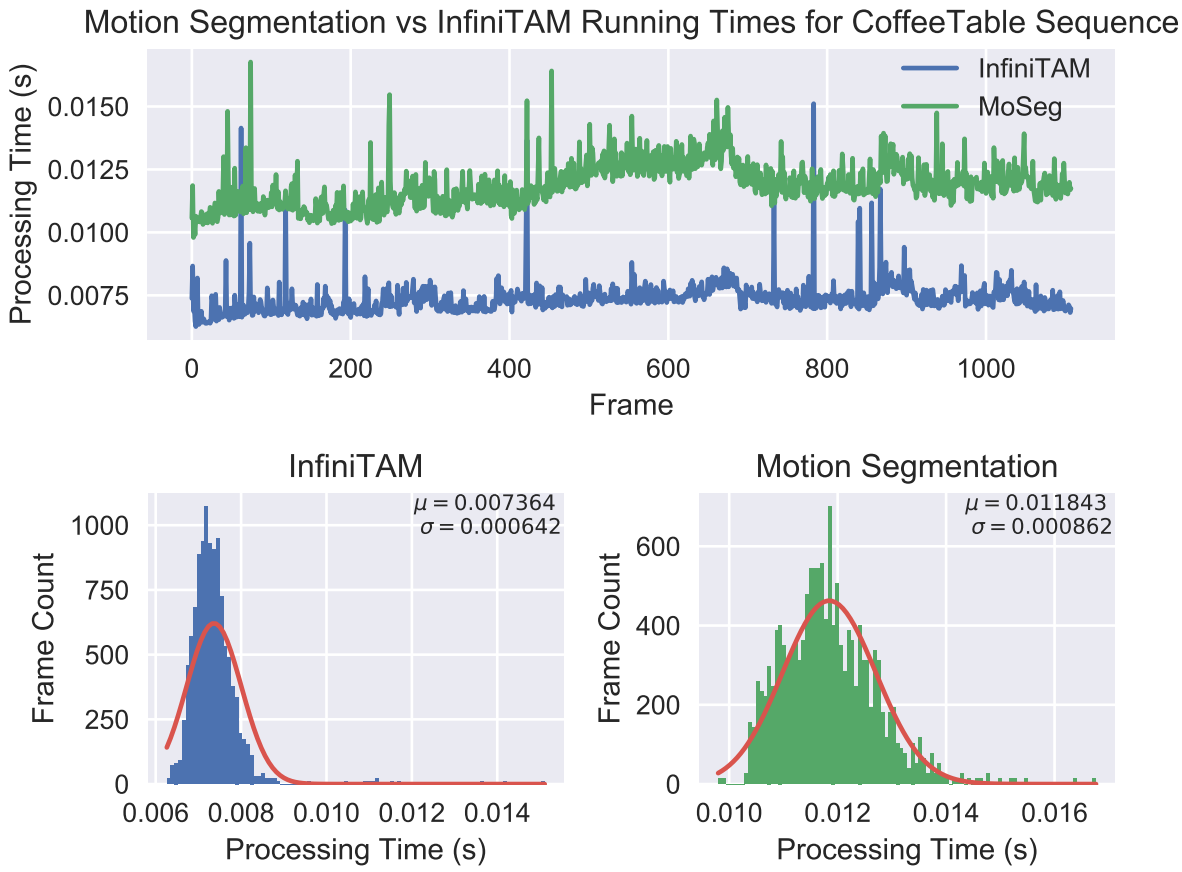


Figure 3.12: Performance of the proposed approach versus the standard dense SLAM pipeline of *Prisacariu et al* [30].

It can be seen from Figure 3.12 that the motion segmentation and dynamic SLAM system proposed in this work is capable of operating at above real time performance. The sequence from which the performance statistics of Figure 3.12 are derived consists of 1108 frames in which dynamics are prevalent throughout. It can be seen that although the cost of the proposed system is higher than that of the standard SLAM pipeline, performance is still better than real-time. It should be noted however that the mean processing time for a given frame is 0.011843 seconds, the standard deviation over the sequence is 0.000862 seconds. Such varying per frame times are due to the varying degree of observed dynamic scene components. However, the proposed approach performs at $\approx$ 84Hz versus the standard pipeline's performance of $\approx$ 136Hz, indicating a non detrimental performance deficit.

It should be noted that the quoted performance is that of the pipeline outlined in this chapter and does not include the time taken to perform on-screen rendering (OpenGL texture writing and GLUT window updating).

3.7 APPLICATION TO 3D OBJECT RECOGNITION

The dynamic scene handling approach described in Section 3.3 can be used to prevent moving objects from being integrated into the static scene model. However, 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, as outlined in Section 1.1. In this section, it is therefore shown 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 object, a one-class Support Vector Machine (SVM) [108] with a polynomial kernel is trained and used to recognise new instances of the object class.

Upon seeing a new object, the system first tries to classify the object into a category that has already been seen, by predicting the objects class with all of the existing single class SVMs. If this fails, the system generates a label for the object and trains a new SVM for it's class.

To make the training examples for the detected object, points are uniformly sampled from the object's isosurface and Fast Point Feature Histogram (FPFH) descriptors [109] are computed at these points. FPFH descriptors are geometric features that provide a per-point statistic of the curvature within some neighbourhood of the point (empirically, a 2.5cm radius around the point is used). Figure 3.13 shows an example of this training and prediction process for dynamic objects.

3.8 SUMMARY

As outlined in Sections 3.4 and 3.5, the proposed approach provides an improvement on the quality of camera pose estimation when performing dense reconstruction in scenes that have dynamic components, versus the standard *KinectFusion* [1] like pipeline as implemented by *Prisacariu et al* [30]; a research objective that was outlined in Section 1.4.

Additionally, it is evident that the proposed approach is capable of segmenting dynamic components (such as people moving in an articulated fashion) that are visible in the camera's view frustum, from those that are static (such as furniture in the scene). This segmentation also demonstrates a novel way of obtaining 3D geometry information to perform rudimentary scene understanding. Additionally, the

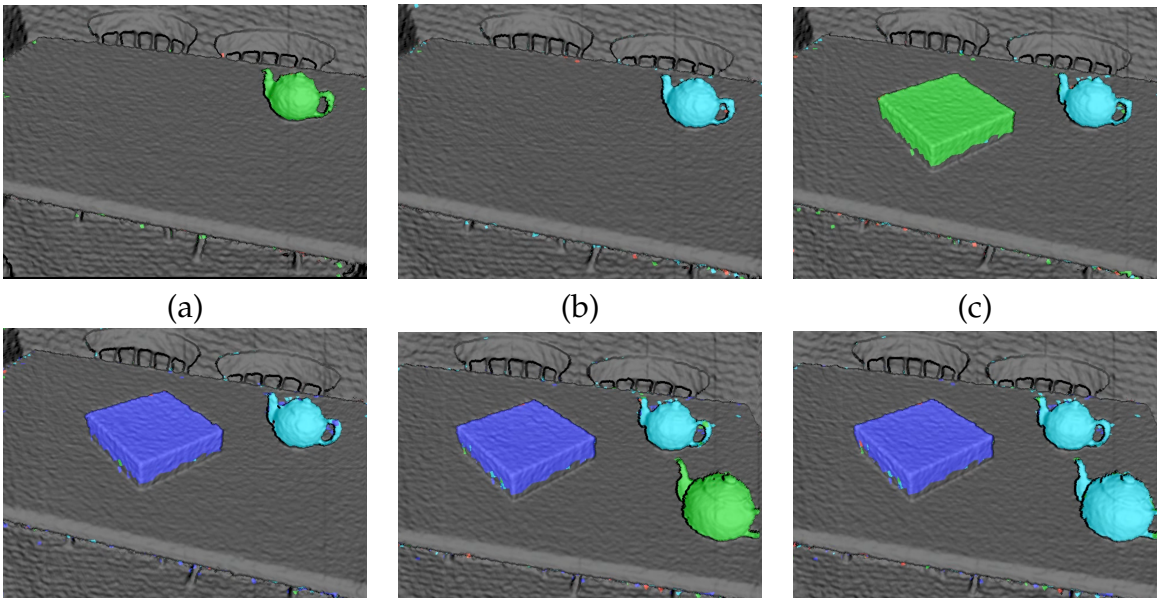


Figure 3.13: An example of the training and prediction process for dynamic objects:
(a) A teapot is placed in the scene.
(b) The teapot is recognised as a new object and an SVM is trained for it.
(c) A box is placed in the scene.
(d) The box is recognised as being distinct from the teapot, so a separate SVM is trained.
(e) Another teapot is placed in the scene.
(f) The new teapot is recognised, and is labelled accordingly.

resultant reconstructions of the proposed approach are qualitatively comparable to the comparison static dense SLAM system (InfiniTAM). Both of these results relate again to the research objectives of Section 1.4.

The novel dual volumetric representation approach taken in this work of maintaining both a *static* and a *dynamic* scene, utilising only the former for camera pose estimation provides a simple and robust system for dense 3D reconstruction in environments that would otherwise be troublesome. Contrary to existing approaches outlined in Section 2.3, the presented system has scope for use in large-scale reconstruction scenarios, due to it's leverage of efficient, hashed volumetric data structures. In addition, by utilising such data structures, the proposed system performs to the level of highly optimised [30] *KinectFusion* [1] implementations, demonstrating feasibility for application to problems that require real time performance.

Since the development of the work outlined in this chapter, there has been further research into combined dynamic *and* object-centric SLAM. *R´unz et al* [110] in *2017* introduced an approach to dynamic SLAM in which dynamic objects are segmented (in image space), reconstructed and tracked independently. In a future iteration of this work, such a schema could be adopted and combined with the approach of this chapter to yield a real-time, object aware dynamic SLAM system. The authors cite the performance of their approach as being $\approx$ 12Hz. Further advancements by *R´unz et al* [111] utilise cues from a state-of-the-art [49] semantic segmentation algorithm to identify probable dynamic objects in a given scene, for instance humans, and exclude them from pose estimation accordingly. This contribution provides an additional, potential future expansion upon the approach of this chapter.

With the contributions outlined in this work, the proposed approach provides a platform for further research into problems such as live semantic scene understanding and dense scene flow. The proposed approach has the potential to be further developed in to a fully dynamic, semantic scene understanding and reconstruction system for use

in robotics, VR and AR applications as outlined in Section 1.1. Though as previously outlined, the contributions of this chapter allow for the extraction of 3D geometry data for machine learning purposes, it is reliant on the detection of motion. As such, the topic of Chapter 4 is 3D object reconstruction.

4

PROBABILISTIC OBJECT RECONSTRUCTION WITH ONLINE DRIFT CORRECTION

This chapter introduces an approach to the reconstruction of arbitrary objects in a globally consistent manner with RGBD observations. In this chapter, a probabilistic formulation of object reconstruction is given, in which correction of model inconsistencies is performed on-line. The approach outlined in this chapter demonstrates improvements in pose estimation and reconstruction quality against multiple baseline approaches.

4.1 INTRODUCTION

Dense SLAM has proven to be an effective paradigm for the reconstruction of scenes of moderate scale, with much research on the topic being 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. This is due to the small but cumulative

errors incurred in pose estimation that prevent a reconstructed model from being closed when the camera returns to it's starting position.

Such model inconsistencies and corruptions are prohibitive when accurate, non-synthetic geometric models of real world objects are required. As outlined in Section 1.2, such accurate geometric data is needed for learning based approaches to semantic scene understanding.

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 problems, 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 reconstruction 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 sub-segments, such that deformations may be inferred to keep the sub-segments aligned, resulting in a consistent overall model. The system utilises a volumetric representation for each of these object sub-segments, as with many larger scale reconstruction systems. Each voxel in a given sub-segment 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 [55] framework. Textured renderings of various objects reconstructed with the proposed system are given in Figure 4.1.

(a) (b) (c)

Figure 4.1: (a) Textured reconstruction of a Teddy Bear.
(b) Textured reconstruction of a Chair.
(c) Textured reconstruction of a Filing Cabinet.

The proposed system is inspired by that of *Kolev et al* [68] in that the representation used for the shape of the object to be modelled is a volume of probabilities, pertaining to posteriors over a voxel's 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 [1] and InfiniTAM [30].

The probabilities that are "fused" into the volume are generated from an appearance model, initialised prior to reconstruction by a Maximum Likelihood Estimation (MLE) [55] 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 input by the user on the first RGB frame. An appearance distribution is fitted over the appearance features of each region, foreground and background.

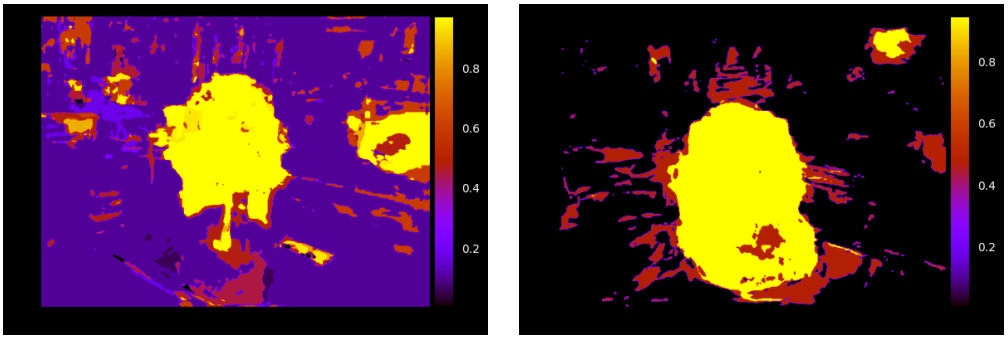
During the fusion process, the Probability Density Functions (PDF) or PMF's of these distributions are evaluated on the latest colour observation for a given frame. The aforementioned voxel posteriors are computed & updated accordingly in the probability volume, with respect to the instantaneous image space probability maps shown in Figures 4.2. Only those voxels with a posterior higher for the foreground than the background are rendered.

The remainder of this chapter proceeds as follows; Section 4.2 provides a high level overview of the proposed algorithm in terms of it's separate components. Section 4.3 introduces and formalises the probabilistic framework on which the proposed object reconstruction system is based. Following the introduction of the probabilistic framework, Section 4.4 introduces it's use for the online correction process. Section 4.5 describes the volumetric segmentation approach taken for refinement of the resultant object reconstructions. To conclude the introduction of the technical aspects of the system introduced in this chapter, Section 4.6 provides an algorithmic summary of the proposed approach. Sections 4.7 and 4.8 present comparative qualitative and quantitative results, respectively, against alternative object reconstruction approaches. Finally, Sections 4.9 and 4.10 provide performance analyses and a discussion of findings, respectively.

4.2 ALGORITHM OVERVIEW

In the proposed system, the object model is divided into *sub-volumes*, each consisting of a TSDF, colour volume and object probability volume. Additionally, each has associated with it, a rigid body transform $\mathbf{T} \in \mathbb{SE}(3)$ that specifies it's 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 (as shown in Figure 4.2) 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 sub-volume. Examples of such instantaneous probability maps are given in Figure 4.2.



(a) (b)

Figure 4.2: (a) An instantaneous probability map in which much noise is present in the background.
(b) A map that is very probabilistically polarized with respect to foreground and background beliefs.

As with the dense SLAM system outlined earlier in this work, the proposed system also has *Integration*, *Tracking* and *Rendering* stages in its 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 sub-volumes, mitigating tracking drift. Once the reconstruction process is finished, a CRF based optimisation is performed to refine the resulting object segmentation over all sub-volumes.

The proposed approach is not tied to the use of any one probabilistic model, though in the presented experiments PwP of *Bibby et al* [72] is used. An overview of the object reconstruction pipeline is shown in Figure 4.3.

4.3 PROBABILISTIC FORMULATION OF OBJECT RECONSTRUCTION

The surface map and camera pose are estimated using the standard KinectFusion like pipeline of *Newcombe et al, Prisacariu et al* [1, 30]. The surface is represented as the Zero Level Set of a TSDF discretised over voxels, with the iso-surface embedding

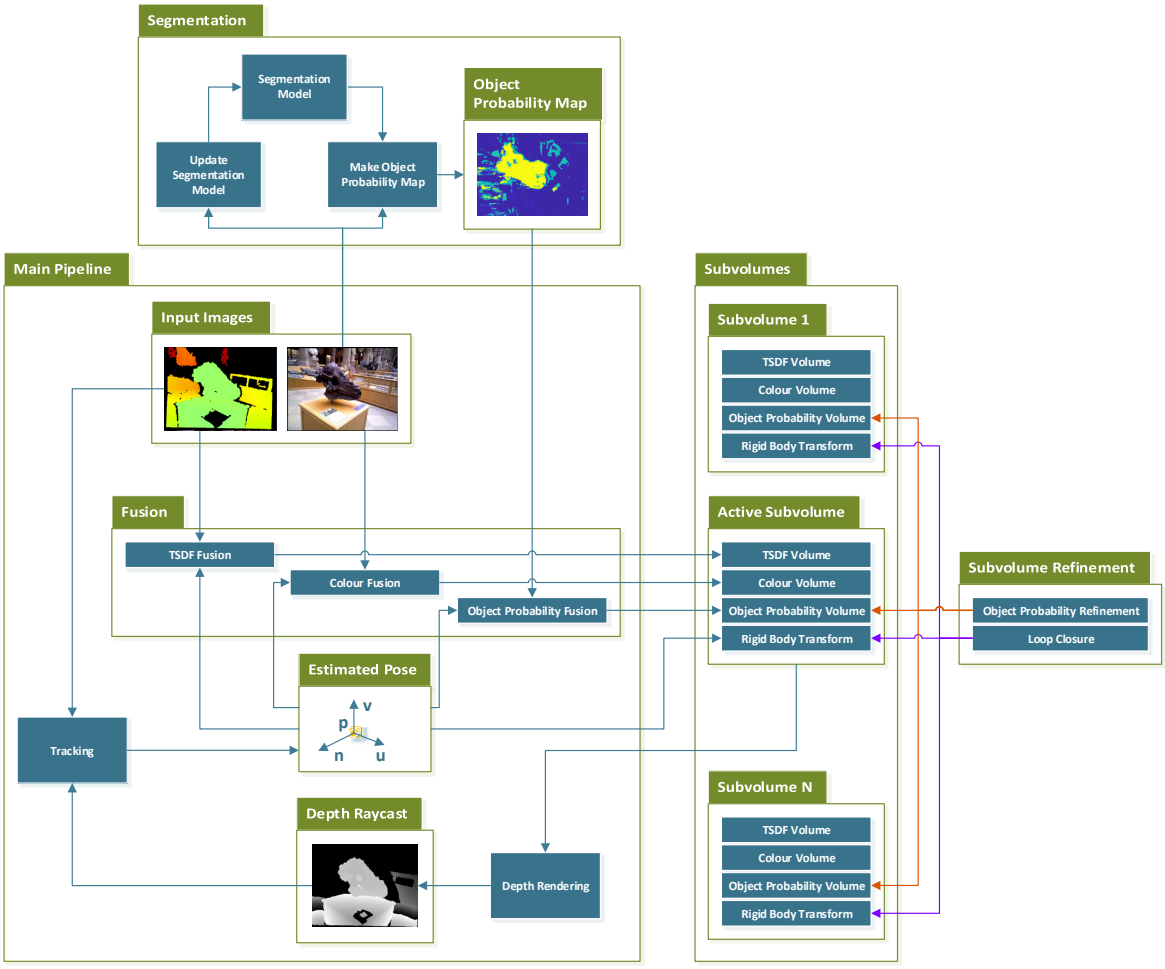


Figure 4.3: The pipeline of the proposed Object Reconstruction approach.

built by a weighted mean of new observations, as outlined in Equations 3.28 and 3.29. Camera pose estimation is performed with ICP, as outlined in Section 3.2.1 and is run simultaneously against the evolving map. Here, inspired by [68], this procedure is 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 *sub-volumes*, each pertaining to some patch on the object surface. Additional sub-volumes are created when sufficiently many new voxels have been allocated and have had SDF data integrated. By ensuring overlap between the sub-volumes, transformations between them can be found and pose inconsistencies addressed, online. Empirically, the threshold for starting a new sub-volume is defined as the event when 50% of the voxels fused in to the current volume are newly observed points, such that there is sufficient overlap between two sub-volumes that shall later be registered.

4.3.1 *Volumetric Appearance Model*

At each observed RGBD frame, the object posterior probabilities for the visible voxels in the active sub-volume are updated via an appearance-derived probability map for that frame, as described in Section 1.3. Under the assumption of conditional independence between frames (for the sake of tractability), the posterior probability of a given voxel $\psi \in \Psi$ belonging to the object has the following form (noting that $\Phi \subset \Psi$):

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

In Equation 4.1 Ψ is the volume of voxels for which measurements are accumulated, Φ is the volume of voxels pertaining to the object Φ of interest, Ω_t is the current RGBD image observation at time t and p_t is the currently tracked pose at time t. Equation 4.1 gives 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.

4.3.2 *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 for formulation of the full joint distribution over the object as the Probabilistic Graphical Model (PGM) of Figure 4.4.

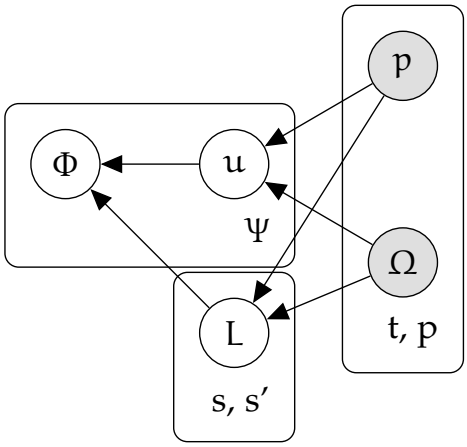


Figure 4.4: Probabilistic Graphical Model representing the full joint distribution over the shape Φ of the object of interest.

In Figure 4.4, 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 of Equation 4.1), L is the set of consistency constraints for each adjacent sub volume pair in the form of

rigid body transformations, Ω is the set of RGBD image pixels and p the set of poses over time.

The PGM given in Figure 4.4 leads to the factorisation over the full joint distribution given in Equation 4.2.

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

In Equation 4.2, Ψ is the set of voxels across all sub-volumes, $\mathcal{P}$ is the set of RGBD pixels for a given frame Ω_t, and $\mathcal{S}$ is the set of sub-volumes. Note that the notation $s, s' \in \mathcal{S}$ refers to pairs of adjacent, overlapping sub-volumes.

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 4.5.

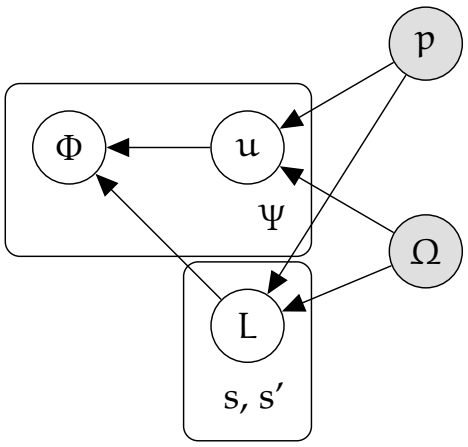


Figure 4.5: Probabilistic Graphical Model representing the simplified joint distribution over the shape Φ of the object of interest.

The simplifications transforming the PGM of Figure 4.4 in to that of Figure 4.5 lead to the factorisation of the joint distribution over Φ given in Equation 4.3.

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

The formalisms defined in Figures 4.4 and 4.5, and Equations 4.2 and 4.3 describe a probabilistic framework in which online corrections can be made to the reconstructed model (piecewise over sub-volumes), to counter errors caused by pose tracking inconsistencies. As with scene scale dense SLAM systems [1, 30, 20], the presented system follows a pipeline that consists of a tracking stage and an integration stage, as outlined in Section 3.2.

However, the presented formulation of this pipeline consists of an additional, novel estimation module that relies on the use of a sub-volume representation (an example of which is given in Figure 4.6) to correct tracking errors by applying rigid body transformations to the sub-segments of the reconstructed shape (the sub-volumes) to correct their alignment when there are intra sub-segment tracking inconsistencies. As inference on the full joint distribution of the presented probabilistic model is intractable, conditional independence assumptions are made that empirically do not appear to cause any functional issues.

Note that only voxels whose appearance posterior is greater for the foreground class are used in the correction procedure; only voxels that satisfy the probabilistic condition $P(\psi \in \Phi | \Omega, p) > 1 - P(\psi \in \Phi | \Omega, p)$ are considered.

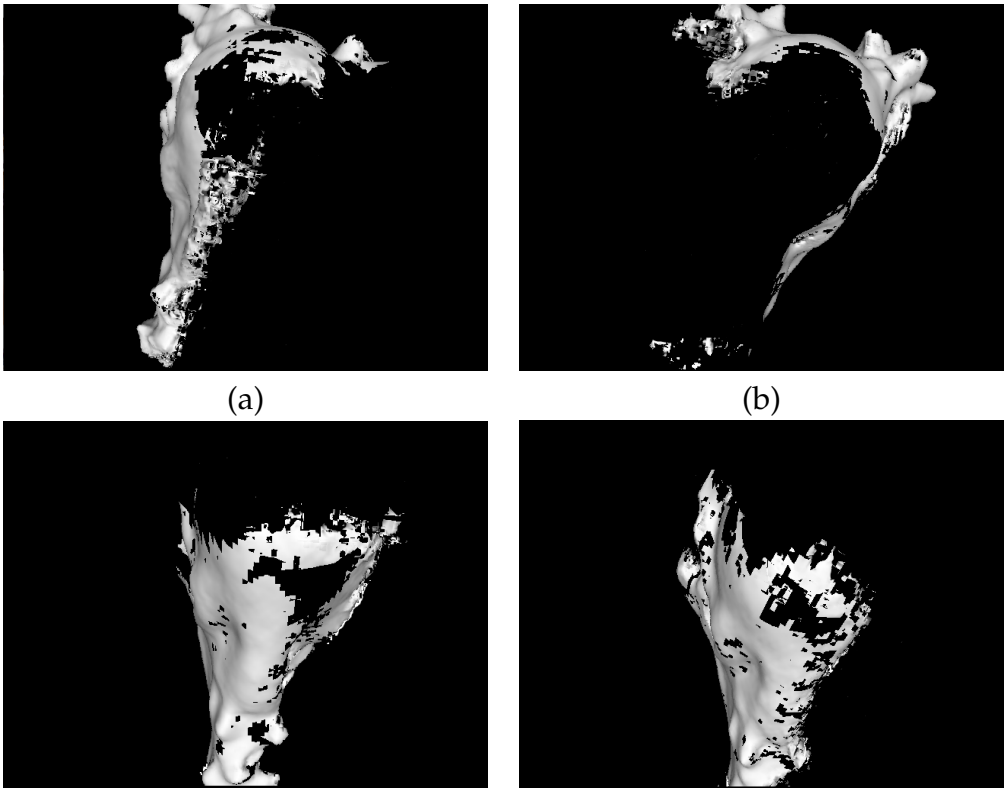


Figure 4.6: Example subvolume renderings of a given object of interest. As viewed from the front:
(a) Left hand side and back.
(b) Back and right hand side.
(c) Top and sides
(d) Top and left hand side.

4.3.3 *Appearance Marginal*

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

$$P(\Phi, \Omega, p, L) = \int \prod_{\psi \in \Psi} P(\Phi \mid u_\psi) \prod_{s,s' \in \mathcal{S}} P(u_\psi \mid \Omega, p, L_{s,s'}) P(L_{s,s'} \mid \Omega, p) P(L_{s,s'}) P(p) P(\Omega) du \tag{4.4}$$

$$= \prod_{\psi \in \Psi} \int P(\Phi \mid u_\psi) \prod_{s,s' \in \mathcal{S}} P(u_\psi \mid \Omega, p, L_{s,s'}) P(L_{s,s'} \mid \Omega, p) P(L_{s,s'}) P(p) P(\Omega) du \tag{4.5}$$

$$= \prod_{s,s' \in \mathcal{S}} P(L_{s,s'} \mid \Omega, p) P(L_{s,s'}) P(p) P(\Omega) P(\Phi) \tag{4.6}$$

Note that the appearance posterior volume outlined in Section 4.3.1 is reintroduced later in this work in Section 4.4 for the purposes of sub-volume alignment and determining the subset of voxels $\Phi \subset \Psi$ that yield the target object shape. Further details pertaining to the inference procedure for the per-sub-volume deformations are provided in Section 4.4.

4.4 ONLINE MODEL CORRECTION

The tracking consistency constraints denoted by the variables $L_{s,s'}$ such that $s, s' \in \mathcal{S}$, where $\mathcal{S}$ is the set of overlapping sub-volume pairs s, s' in the PGM's given by Figures 4.4 and 4.5. These constraints can be enforced in terms of minimising the disparity between each pair of adjacent sub-volumes. The effect of this minimisation 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 sub-volume pair.

4.4.1 *Alignment MAP Estimate*

Referring back to the joint distribution of Equation 4.3, to achieve the aforementioned minimisation of disparity between overlapping sub-volumes, a MAP estimate is desirable. The MAP estimate over $L_{s,s'}$ in Equation 4.3, for a given sub-volume pair s, s', may be derived as follows.

$$P(\Omega, p|L_{s,s'}) \propto \prod_{s,s' \in \mathcal{S}} \frac{P(L_{s,s'}|\Omega, p)P(\Omega|p)P(p)P(L_{s,s'})}{\int P(L_{s,s'} \mid \Omega, p) dL_{s,s'}} P(\Phi) \tag{4.7}$$

$$\propto \prod_{s,s' \in \mathcal{S}} P(L_{s,s'}|\Omega, p)P(\Omega|p)P(p)P(L_{s,s'})P(\Phi) \tag{4.8}$$

$$\propto \prod_{s,s' \in \mathcal{S}} P(L_{s,s'}|\Omega, p)P(L_{s,s'})P(\Phi) \tag{4.9}$$

Note that in the third step of Equation 4.7 the distributions $P(\Omega|p)$ and $P(p)$ are taken to be uniform and as such may be omitted whilst retaining proportionality. The prior distribution $P(L_{s,s'})$ is conjugate to $P(L_{s,s'}|\Omega, p)$ and is of the form of a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0} \mid \mathbf{I})$ over the $\mathbb{SE}(3)$ deformation parameters described in Section 3.2.1.

The choice of such a prior distribution is motivated by the assumption that motion between consecutive frames is minor (for a real-time sensor running at at least 30Hz), thus the given prior will have the effect of constraining the $\mathbb{SE}(3)$ transformation accordingly. 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 4.4.2.

The rationale of Equation 4.7 is that the deformation $L_{s,s'}$ applied to the sub-volume 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.

4.4.2 *Analytic Form of Alignment MAP Estimate*

With the probabilistic framework now outlined, the analytic form of the posterior given in Equation 4.7 may be explored. The likelihood term in Equation 4.7 quantifies the ability of the constraint $L_{s,s'}$ to maximise consistency between sub-volumes 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 frustum at time t, the posterior $P(\Omega, p|L_{s,s'})$ is given. As outlined in Section 4.4.1, the prior on the constraints $L_{s,s'}$ has the form of a multivariate Gaussian distribution of the form $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

As such, the analytic form of the likelihood term $P(L_{s,s'}|\Omega, p)$ is given as follows.

$$P(L_{s,s'}|\Omega, p) = \prod_{(s,s')\in\mathcal{S}} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}\left[\Phi_s(x) - \Phi_{s'}\left(\Lambda(x,p,t)\right)\right]^2} \tag{4.10}$$

In Equation 4.10, the function $\Lambda(.)$ applies the resultant $\mathbb{SE}(3)$ transform to a given voxel position $\mathbf{x}$ and is defined as follows.

$$\Lambda(\mathbf{x}, \mathbf{p}, \mathbf{t}) = \mathbf{R}_\mathbf{p}\mathbf{x} + \mathbf{t} \tag{4.11}$$

$$= \begin{bmatrix} \mathbf{R}_\mathbf{p} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \mathbf{x} \tag{4.12}$$

In Equation 4.11 $\mathbf{R}_\mathbf{p}$ is the $\mathbb{SO}(3)$ rotation matrix generated from the Rodrigues parameters given by the Vector $\mathbf{p}$ (recall the definition of the Rodrigues parameterisation given in Equation 3.4 of Section 3.2.1).

The form of the aforementioned surface prior $P(\Phi)$ is taken to be that of the logistic distribution, due to the *symmetric* $1 - f(x) = f(-x)$ and *squashing* $\text{Range}[f(x)] = [0, 1]$ properties of it's Cumulative Density Function (CDF). The PDF of the logistic distribution is given as follows.

$$\text{Logistic}(x \mid \mu, \sigma) = \frac{e^{-\frac{x-\mu}{\sigma}}}{\sigma\left(1 + e^{-\frac{x-\mu}{\sigma}}\right)^2} \tag{4.13}$$

To encode the probability of a given voxel $\psi \in \Psi$ containing an isosurface point, represented as the Zero Level Set defined in Equation 3.1 of a TSDF, it is desirable to

quantify the probability with a function that has the aforementioned properties. As such, the CDF of $P(\Phi)'$ is derived as follows.

$$\bar{P}(\Phi) \propto P(\Phi < b) \tag{4.14}$$

$$\propto \int_{-\infty}^{b} \frac{e^{-\frac{\Phi-\mu}{\sigma}}}{\sigma\left(1+e^{-\frac{\Phi-\mu}{\sigma}}\right)^2} d\Phi \tag{4.15}$$

$$\propto \lim_{y\to-\infty} \int_{y}^{b} \frac{e^{\frac{-\Phi+\mu}{\sigma}}}{\sigma\left(1+e^{\frac{-\Phi+\mu}{\sigma}}\right)^2} d\Phi \tag{4.16}$$

$$\propto \lim_{y\to-\infty} \frac{1}{\sigma} \int_{y}^{b} \frac{e^{\frac{u}{s}}}{\left(1+e^{\frac{u}{\sigma}}\right)^2} du \tag{4.17}$$

Where $u = -\Phi + \mu$ and $du = -d\Phi$

$$\propto \lim_{y\to-\infty} -\int_{y}^{b} \frac{e^{p}}{\left(1+e^{p}\right)^2} dp \tag{4.18}$$

Where $p = \frac{u}{\sigma}$ and $dp = \frac{1}{\sigma}du$

$$\propto \lim_{y\to-\infty} -\int_{y}^{b} \frac{1}{w^2} dw \tag{4.19}$$

Where $w = 1 + e^p$ and $dw = e^p dp$

$$\propto \lim_{y \to -\infty} \left[\frac{1}{w} + C \right]_y^b \tag{4.20}$$

$$\propto \lim_{y \to -\infty} \left[\frac{1}{1 + e^p} + C \right]_y^b \tag{4.21}$$

$$= \lim_{y \to -\infty} \left[\frac{1}{1 + e^{\frac{u}{\sigma}}} + C \right]_y^b \tag{4.22}$$

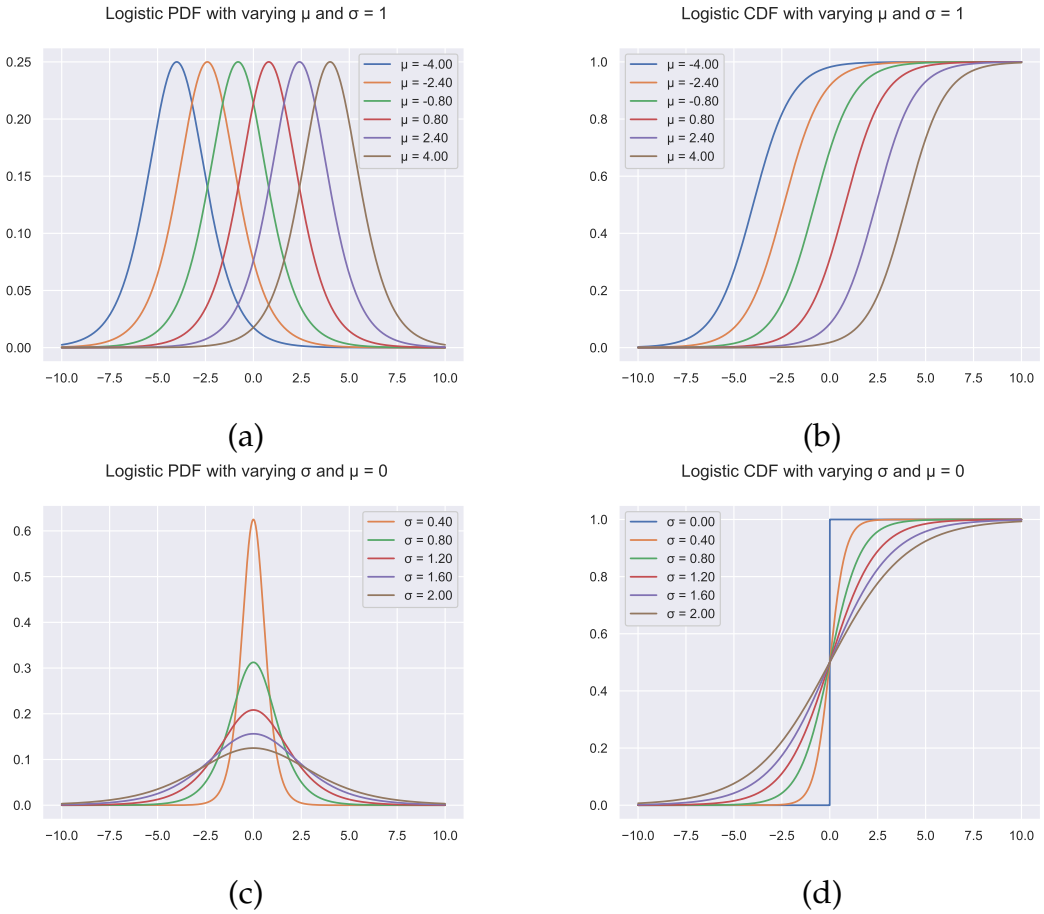
$$\propto \lim_{y \to -\infty} \left[\frac{1}{1 + e^{\frac{-\Phi+\mu}{\sigma}}} + C \right]_y^b \tag{4.23}$$

$$\propto \left[\frac{1}{1 + e^{\frac{-\Phi+\mu}{\sigma}}} + C \right]_{\Phi=b} - \left[\lim_{y \to -\infty} \frac{1}{1 + e^{\frac{-\Phi+\mu}{\sigma}}} \Bigg|_{\Phi=y} + C \right] \tag{4.24}$$

$$\propto \frac{1}{1 + e^{\frac{-\Phi+\mu}{\sigma}}} \tag{4.25}$$

With the CDF of Equation 4.13 derived in Equation 4.14, an appropriate choice of the parameters μ and σ must be made for the prior over the SDF. The logistic PDF and CDF are plotted in Figure 4.7 for varying values of μ and σ.

The requirement of the shape prior is to penalise SDF values that are both far from the isosurface and behind the isosurface. As such, a suitable prior may be found by parameterising the CDF is an appropriate manner. With the parameterisation of $\mu = 4$ and $\sigma = 0.7$, the CDF is shifted appropriately and penalises on a strict range of SDF values. However, in this form the CDF penalises SDF points close to the isosurface. However, this is trivially solved as in Figure 4.8, note the symmetric property of the logistic CDF.



(a) (b)

(c) (d)

Figure 4.7: (a) PDF of the Logistic Distribution with varying μ and $\sigma = 1$.
(b) CDF of the Logistic Distribution with varying μ and $\sigma = 1$.
(c) PDF of the Logistic Distribution with varying σ and $\mu = 0$.
(d) CDF of the Logistic Distribution with varying σ and $\mu = 0$.

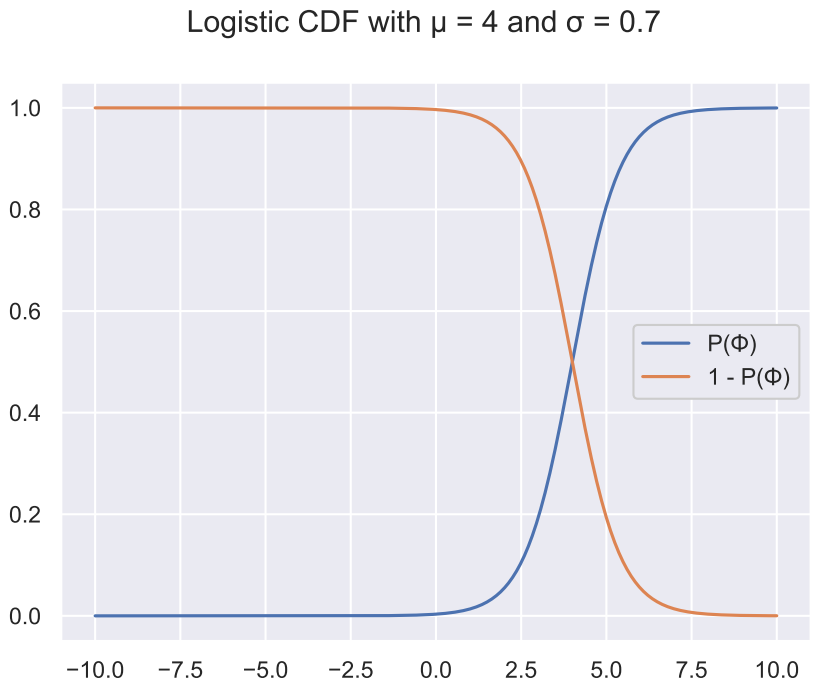


Figure 4.8: $P(\Phi)$ and $1 - P(\Phi)$

Given the CDF derived in Equation 4.14, the prior $P(\Phi)$ is redefined as follows.

$$P(\Phi) \propto \begin{cases} 1 - \int_{-\infty}^{b} \text{Logistic}(\Phi \mid \mu, \sigma) d\Phi \Big|_{\mu=4,\sigma=0.7} & \textbf{if} \Phi \geqslant 0 \\ 0 & \textbf{if} \Phi < 0 \end{cases} \tag{4.26}$$

$$\propto \begin{cases} 1 - \bar{P}(\Phi) & \textbf{if} \Phi \geqslant 0 \\ 0 & \textbf{if} \Phi < 0 \end{cases} \tag{4.27}$$

It is evident from Equation 4.26 that under an appropriate paramaterisation, the CDF of the logistic distribution is the logistic sigmoid function prevelant in the Artificial Neural Network literature. As given in Equation 4.26 the mean μ and standard deviation σ are 0 and 1, respectively.

Finally, the analytic form of the log-posterior is given as follows, optimising for the Rodriguez parameters p and translation vector t of the constraint $L_{s,s'}$ between sub-volumes s and s'.

$$\ln P(\Omega, p|L_{s,s'}) \propto \ln \prod_{s,s' \in \mathcal{S}} P(L_{s,s'}|\Omega, p) P(L_{s,s'}) \bar{P}(\Phi) \tag{4.28}$$

$$\propto \sum_{s,s' \in \mathcal{S}} \ln P(L_{s,s'}|\Omega, p) + \sum_{s,s' \in \mathcal{S}} \ln P(L_{s,s'}) + \sum_{s,s' \in \mathcal{S}} \ln \bar{P}(\Phi) \tag{4.29}$$

$$\propto \sum_{s,s' \in \mathcal{S}} \left[\ln \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}\left[\Phi_s(x) - \Phi_{s'}\left(\Lambda(x,p,t)\right)\right]^2} \right. \tag{4.30}$$

$$\left. + \ln \frac{-\frac{1}{2} e^{p^T \Sigma^{-1} p}}{\sqrt{(2\pi)^k |\Sigma|}} + \ln \frac{1}{1 + e^{-\Phi}} \right]$$

$$\propto \sum_{s,s' \in \mathcal{S}} \left[-\ln(2\pi\sigma) - \frac{1}{2\sigma^2}\left[\Phi_s(\mathbf{x}) - \Phi_{s'}\left(\Lambda(\mathbf{x}, \mathbf{p}, \mathbf{t})\right)\right]^2 - \frac{1}{2} \ln |\Sigma| \right. \tag{4.31}$$

$$\left. - \frac{1}{2} \mathbf{p}^T \Sigma^{-1} \mathbf{p} - \ln(1 + e^{-\Phi}) \right]$$

$$\propto \sum_{s,s' \in \mathcal{S}} \left[-\ln(2\pi\sigma) - \frac{1}{2\sigma^2}\left[\Phi_s(\mathbf{x}) - \Phi_{s'}\left(\Lambda(\mathbf{x}, \mathbf{p}, \mathbf{t})\right)\right]^2 - \frac{1}{2} \mathbf{p}^T \mathbf{p} \right. \tag{4.32}$$

$$\left. - \ln(1 + e^{-\Phi}) \right]$$

As with the gradients derived in Section 3.2.1, the gradient update when optimising Equation 4.28 takes the familiar Levenberg-Marquardt update form, akin to that of Equation 3.24.

4.4.3 *Optimisation for MAP Inference*

To infer the optimal consistency constraints between adjacent sub-volumes, the log-posterior given in Equation 4.28 may be optimised using second order gradient based methods such as Levenberg-Marquardt in a similar fashion to the procedure outlined in Section 3.2.1. As with the ICP procedure outlined in Section 3.2.1, the optimisation is performed with respect to the three Rodriguez rotation parameters α, β, γ and translation parameters t_x, t_y and t_z of the target $\mathbb{SE}(3)$ transformation.

In a similar manner to the process outlined in Section 3.2.1, the target energy function must be differentiated with respect to each of the $\mathbb{SE}(3)$ parameters. In this case, the log-posterior of Equation 4.28 is the target energy function and shall be denoted $E(.)$ in the following derivation. The derivation of the partial derivatives of $E(.)$ with respect to the Rodriguez rotational parameters is as follows for some parameter $\tau \in \{\alpha, \beta, \gamma\}$. It should be noted that the partial derivative with respect to the translation parameters can be derived in a similar manner.

$$\frac{\partial E}{\partial \tau} = \sum_{s,s' \in \mathcal{S}} \left[-\frac{\partial}{\partial \tau} \ln(2\pi\sigma) - \frac{\partial}{\partial \tau} \frac{1}{2\sigma^2} \left[\Phi_s(\mathbf{x}) - \Phi_{s'}(\Lambda(\mathbf{x})) \right]^2 - \frac{\partial}{\partial \tau} \frac{1}{2} \mathbf{p}^\top \mathbf{p} - \frac{\partial}{\partial \tau} \ln(1 + e^{-\Phi}) \right] \tag{4.33}$$

$$= \sum_{s,s' \in \mathcal{S}} \left[\frac{1}{2\sigma^2} \frac{\partial}{\partial \tau} \left[\Phi_s(\mathbf{x}) - \Phi_{s'}(\Lambda(\mathbf{x})) \right]^2 - \frac{1}{2} \frac{\partial}{\partial \tau} \mathbf{p}^\top \mathbf{p} - \frac{\partial}{\partial \tau} \ln(1 + e^{-\Phi}) \right] \tag{4.34}$$

$$= \sum_{s,s' \in \mathcal{S}} \left[\frac{1}{2\sigma^2} \frac{\partial \phi}{\partial \Phi} \frac{\partial \Phi}{\partial \Lambda} \frac{\partial \Lambda}{\partial \mathbf{R}_\tau} - \frac{1}{2} \frac{\partial}{\partial \tau} \mathbf{p}^\top \mathbf{p} - \frac{\partial}{\partial \tau} \ln(1 + e^{-\Phi}) \right] \tag{4.35}$$

Where $\phi(.) = \left(\Phi_s(\mathbf{x}) - \Phi_{s'}(\Lambda(\mathbf{x}))\right)^2$

$$= \sum_{s,s' \in \mathcal{S}} \left[\sigma^2 \phi(.) \frac{\partial \Phi}{\partial \Lambda} \frac{\partial \Lambda}{\partial \mathbf{R}_\tau} - \frac{1}{2} \frac{\partial}{\partial \tau} \mathbf{p}^\top \mathbf{p} - \frac{\partial}{\partial \tau} \ln(1 + e^{-\Phi}) \right] \tag{4.36}$$

$$= \sum_{s,s' \in \mathcal{S}} \left[\sigma^2 \phi(.) \frac{\partial \Phi}{\partial \Lambda} \frac{\partial \Lambda}{\partial \mathbf{R}_\tau} - \mathbf{p}^\top - \frac{\partial}{\partial \tau} \ln(1 + e^{-\Phi}) \right] \tag{4.37}$$

$$= \sum_{s,s' \in \mathcal{S}} \left[\sigma^2 \phi(.) \frac{\partial \Phi}{\partial \Lambda} \frac{\partial \Lambda}{\partial \mathbf{R}_\tau} - \mathbf{p}^\top + \frac{1}{1 + e^{\Phi}} \frac{\partial \Phi}{\partial \Lambda} \frac{\partial \Lambda}{\partial \mathbf{R}_\tau} \right] \tag{4.38}$$

4.4.4 *Implicit Surface Deformation*

The resultant object surface Φ is implicitly deformed according to the inferred consistency constraints L by a blending function $\zeta(\mathcal{X})$ over each set $\mathcal{X}$ of overlapping sub-volumes for which surface data has been integrated. The set $\mathcal{X}$ is defined as follows in Equation 4.39.

$$\mathcal{X} = \left\{ L_{s,s'}, \Psi_s, \Psi_{s'} \right\} \forall s, s' \in \mathcal{S} \tag{4.39}$$

The blending of surface data between two volumes Ψ_s and $\Psi_{s'}$ approximates the *true* object surface Φ, assumed to exist between that of Ψ_s and $\Psi_{s'}$. This notion is depicted in Figure 4.9. In Figure 4.9, $\mathbf{T}_1$ and $\mathbf{T}_2$ are $\mathbb{SE}(3)$ transforms applied to the surfaces of the object sub-volumes.

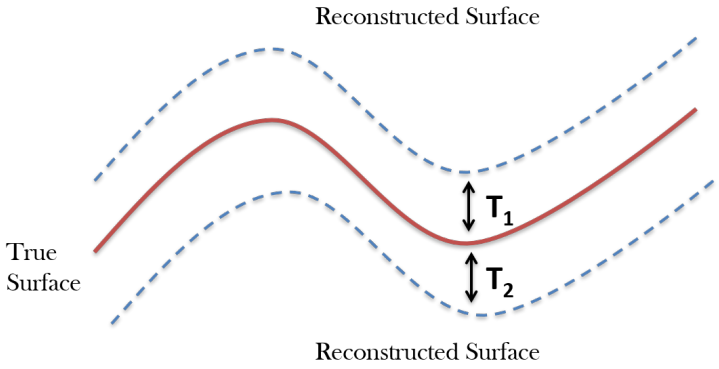


Figure 4.9: An example of deformations being applied to observed surfaces to approximate a true surface.

For the experiments performed in this work, the blending function ζ is defined as follows in Equation 4.40. Note that the output of the blending function ζ is the approximated object surface Φ.

$$\Phi = \zeta(\mathcal{X}) \tag{4.40}$$

$$= \frac{1}{|\mathcal{X}|} \sum_{\chi \in \mathcal{X}} \Big[\Psi_s(v) - \Psi_{s'}(v') \forall v, v' \in \Psi_s \cup \Psi_{s'} \leftrightarrow \tag{4.41}$$

$$1 - P(\Psi_s(v) \in \mathbf{\Phi} | \Omega, p) < 0.5 \wedge$$

$$1 - P(\Psi_{s'}(v') \in \mathbf{\Phi} | \Omega, p) < 0.5 \Big]$$

4.5 VOLUMETRIC SEGMENTATION AND EXPLICIT LOOP CLOSURE DETECTION

The final component of the proposed pipeline performs a segmentation in 3D observed model space to perform refinements over the appearance posteriors that separate the voxels pertaining to the object of interest from those that have irrelevant measurements

integrated, i.e. background measurements that have not been used to perform pose estimation and as such have not been rendered.

This segmentation is formulated within a CRF framework, with each node in the CRF graph representing a set of neighbouring voxels in volumetric space, where connections are made between adjacent voxel neighbourhoods. This segmentation process is posed as an energy minimisation problem to optimise for a cut in down-sampled voxel space between observations pertaining to the object of interest and those of irrelevant observations, such that a segmentation in 3D is obtained. The central purpose of this segmentation is to refine the resultant output model of the proposed object reconstruction system. The described CRF model is depicted in Figure 4.10

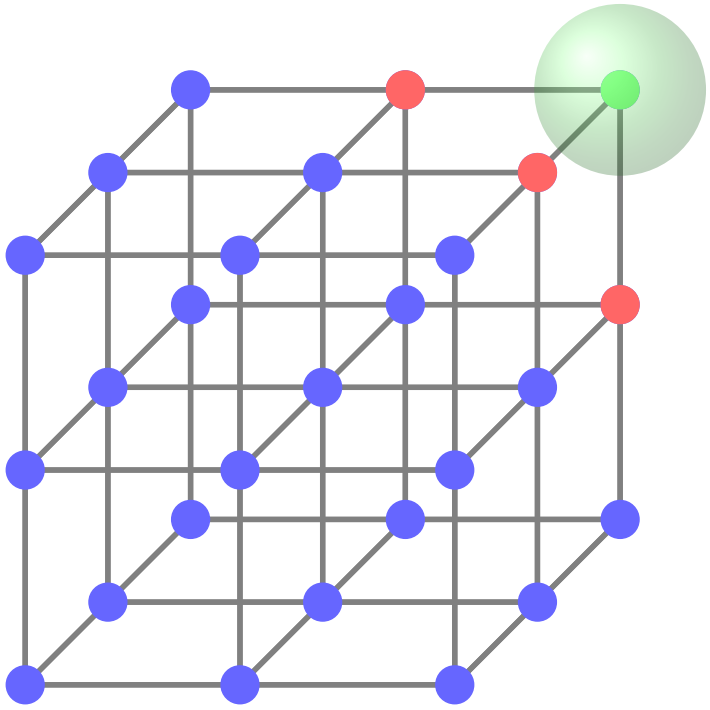


Figure 4.10: The 3D CRF model over Voxel space.

The following energy function consists of the appearance posterior probabilities accumulated during the on-line fusion process for a region in space as the CRF unary potentials. The pairwise smoothing term represents the physical appearance

similarity of the observation regions represented by the texture fused in to the voxels of neighbourhoods γ and γ':

$$E_n = \prod_{t=0}^{\infty} \prod_{\psi \in \Psi_n} P(\psi \in \Phi \mid \Omega_t, p_t) + P(\mathbb{E}[c]_{\gamma} \mid \mathbb{E}[c]_{\gamma'}) \tag{4.42}$$

In Equation 4.42 the terms $\mathbb{E}[c]_{\gamma}$ and $\mathbb{E}[c]_{\gamma'}$ of the pairwise component of the energy function are the expected values over appearance for the 3D regions γ and γ' respectively. Recall the unary term $P(\psi \in \Phi \mid \Omega_t, p_t)$ from Equation 4.1 of Section 4.3.1. In the implementation of this work, the voxel neighbourhood of a given voxel is defined to be it's first degree connected components for which valid SDF data has been integrated.

In the implementation used in this work, the aforementioned cut in voxel space is obtained by optimising the energy function of Equation 4.42 within a Max-Flow framework [112] due to GPU parallelisation potential.

4.6 PIPELINE SUMMARY

As outlined, the proposed object reconstruction system performs online correction of reconstruction errors incurred by erroneous pose estimation results. The process of reconstruction and correction is given in the algorithm that follows.

Algorithm 4 Object Reconstruction with Drift Correction

procedure OBJECT RECONSTRUCTION ITERATION($\boldsymbol{\Omega}, \mathcal{S}, u, t$)
  $\mathcal{P} \leftarrow \text{getProbMap}(\boldsymbol{\Omega}_{rgb})$
  for $s \in \mathcal{S}$ **do** ▷ For Each Submap s
    if isVisible(s) **then**
      $\Phi_s \leftarrow \text{getVolume}(s)$
      $\mathbf{T}_t^s \leftarrow \text{getPose}(s)$
      $\mathcal{R}_s \leftarrow \text{raycast}(\Phi_s, \boldsymbol{\Omega}_{depth}, \mathbf{T}_t^s)$
      $\mathbf{T}_{t+1}^s \leftarrow \text{estimatePose}(\mathcal{R}, \mathcal{D})$
      updatePose($\mathbf{T}_{t+1}^s$)
      $\mathcal{D} \leftarrow \text{unproject}(\boldsymbol{\Omega}_{depth})$
      $\Phi_s \leftarrow \text{integrate}(\mathcal{D}, \Phi_s, \mathbf{T}_{t+1}^s)$
      updatePosteriorVolume($\mathcal{D}, u, \mathcal{P}, \mathbf{T}_{t+1}^s$) ▷ Update Posterior with $\mathcal{P}$
      if adjacentVolumesOverlap(s) $\geqslant 50\%$ **then**
        newAdjacentSubmap($s, \mathbf{T}_{t+1}^s$)
      end if
    end if
  end for
  for $s, s' \in \mathcal{S}$ **do** ▷ For Overlapping Submaps s, s'
    $\Phi_s \leftarrow \text{getVolume}(s)$
    $\Phi_{s'} \leftarrow \text{getVolume}(s')$
    $\mathbf{L}_{s,s'} \leftarrow \text{getConstraints}(s, s')$
    $\mathbf{L}^{\star}_{s,s'} \leftarrow \text{inferOptimalConstraints}(\Phi_s, \Phi_{s'}, \mathbf{L}_{s,s'})$ ▷ Optimise Equation 4.33
    updateConstraints($s, s', \mathbf{L}^{\star}_{s,s'}$)
  end for
  if $t\%n == 0$ **then**
    for $s \in \mathcal{S}$ **do**
      maxFlow(s) ▷ Optimise Equation 4.42
    end for
  end if
end procedure

The process outlined in the given algorithm is highly amenable to GPU parallelisation. The first loop integrates the observed depth map into each active sub-map, estimates it's updated pose and adds instantaneous probability information to it's appearance posterior volume. As outlined in Section 3.3.3, the integration and pose estimation phases are largely parallelisable with minimal risk of race condition occurring if care is taken.

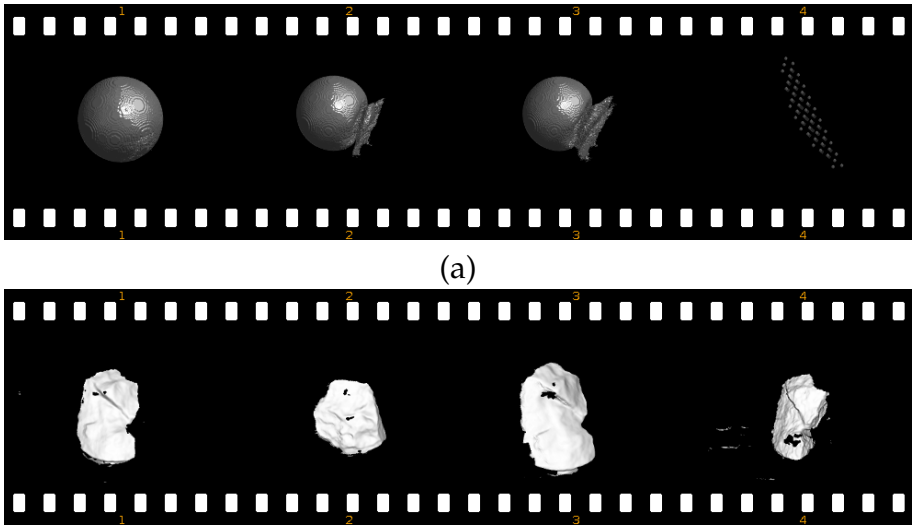
Though the second loop itself is not directly parallelisable due to the dependencies between sub-maps given their consistency constraints, the inference of optimal constraints is parallelisable. Though multiple constraints may not be optimised for at a time, the process of registering two adjacent sub-maps is easily amenable to GPU parallelisation, when each sub-map pair is taken sequentially.

Finally, with careful implementations and use of atomic primitives, the max flow procedure performed on each sub-map may be parallelised for GPU hardware. In addition, each sub-map may be processed in parallel as there is no data dependency at this stage that would be problematic for the task of performing segmentation on each individual sub-map.

4.7 QUALITATIVE RESULTS

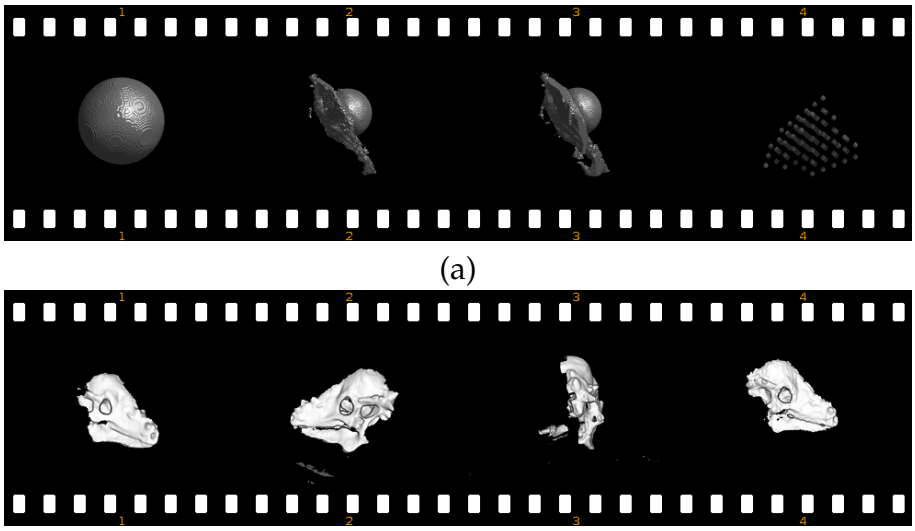
Empirically the proposed system is capable of reconstructing a range of objects characterised by a range of different sizes and geometries. The experiments in this section demonstrate efficacy over the approach of *Ren et al* [74] for the task of obtaining closed, small object reconstructions from RGBD data. Each evaluation sequence is run on the system proposed in this chapter versus that of *Ren et al*, with output snapshots taken at quarterly intervals for each sequence. In Section 4.8, a quantitative evaluation of reconstruction quality is given.

As can be observed in Figure 4.11, the proposed system of *Ren et al* begins with a discretised SDF shape prior of a sphere that is evolved over time whilst simultaneously optimising for pose. However, it is evident that through the course of the sequence it is unable to evolve sufficiently to model the object of interest, contrary to the approach outlined in this chapter.



(a) (b)

Figure 4.11: (a) The system of *Ren et al* [74] evolving a shape prior SDF from RGBD observations on the Museum Rock sequence.
(b) The proposed system (of this work) reconstructing the same object from RGBD observations of the Museum Rock sequence.



(a) (b)

Figure 4.12: (a) The system of *Ren et al* [74] evolving a shape prior SDF from RGBD observations on the Museum Dinosaur Head sequence.
(b) The proposed system (of this work) reconstructing same the object from RGBD observations of the Museum Dinosaur Head sequence.

As with the example given in Figure 4.11, it can be seen in Figure 4.12 that the system of *Ren et al* is again unable to evolve the SDF shape prior sufficiently to model the object of interest. As can be observed, again the proposed system of this work is capable of yielding a suitable reconstruction.

An additional evaluation is performed on the proposed systems ability to reconstruct a range of objects versus that of an implementation of the standard KinectFusion pipeline as outlined in Section 3.1. The central difference in the use of the two approaches for the purposes of this comparison is that in the proposed system, only the observed points of the live frame and of the reconstruction that belong to the object of interest are used for pose estimation.

In the KinectFusion pipeline, used as a base of comparison however, the entire scene is modelled as the camera moves, with the reconstruction of the object of interest being segmented as a post processing step. As such, the results extracted from the KinectFusion pipeline are taken to be the ground truth reconstructions of the object of interest. This is due to the more robust pose estimation that would be expected from utilising all available geometric data for tracking, versus utilising only that of a single, potentially small object. The reconstructions of each approach are given in Figures 4.13 and 4.14.

As can be seen in Figures 4.13 and 4.14, the proposed system is capable of providing reconstructions of a wide variety of objects in an aesthetically similar manner to a well established baseline. The proposed system provides globally consistent, closed reconstructions as is evident from the top-down views presented in Figures 4.13 and 4.14.

The efficacy of the approach described in this work is further demonstrated when comparing the pipeline as outlined in Figure 4.3 to a version that utilises only a single representation of the object of interest. Figure 4.15 demonstrates the difference in

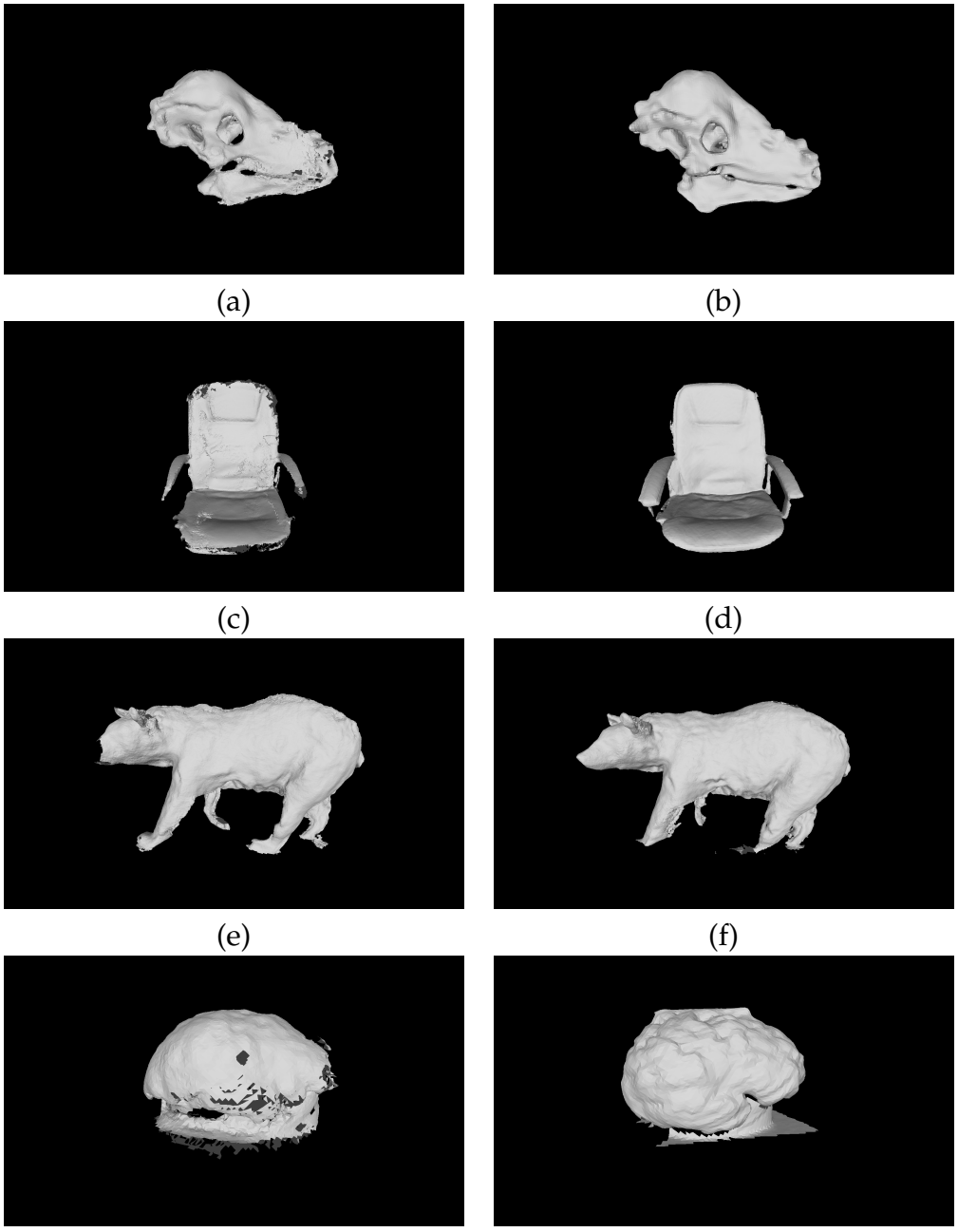


Figure 4.13: Comparisons of object reconstructions, versus manually segmented (after scene reconstruction) ground truth models:
(a, b) Dinosaur Head.
(c, d) Chair.
(e, f) Bear.
(g, h) Brain.
Reconstructions from the proposed system are on the left, ground truth is on the right.

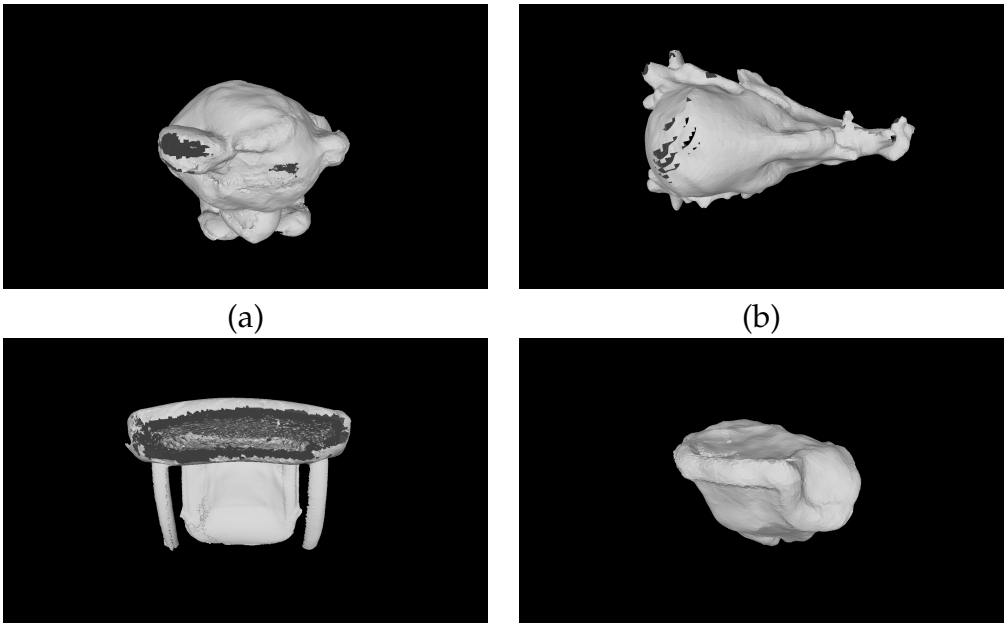


Figure 4.14: Closed reconstructions of the following sequences: (a) Teddy.
(b) Dinosaur Head.
(c) Chair.
(d) Rock sequences.

reconstruction output when not utilising the multiple sub-volume representation and it's accompanying enforcement of consistency constraints.

The gaps in the isosurface depicted in Figure 4.15 (a, b) demonstrate the impact that drift in the pose estimation stage can have on the resultant reconstruction; drift can introduce irregularities in the extraction and rendering of the objects isosurface. Note that the reconstructions given in Figure 4.15 (c, d) do not suffer such irregularities when utilising the object reconstruction pipeline presented in this chapter.

A final set of examples of the system in use for face reconstruction are given in Figure 4.16, the objective of which is to demonstrate tracking with mostly uniform appearance.

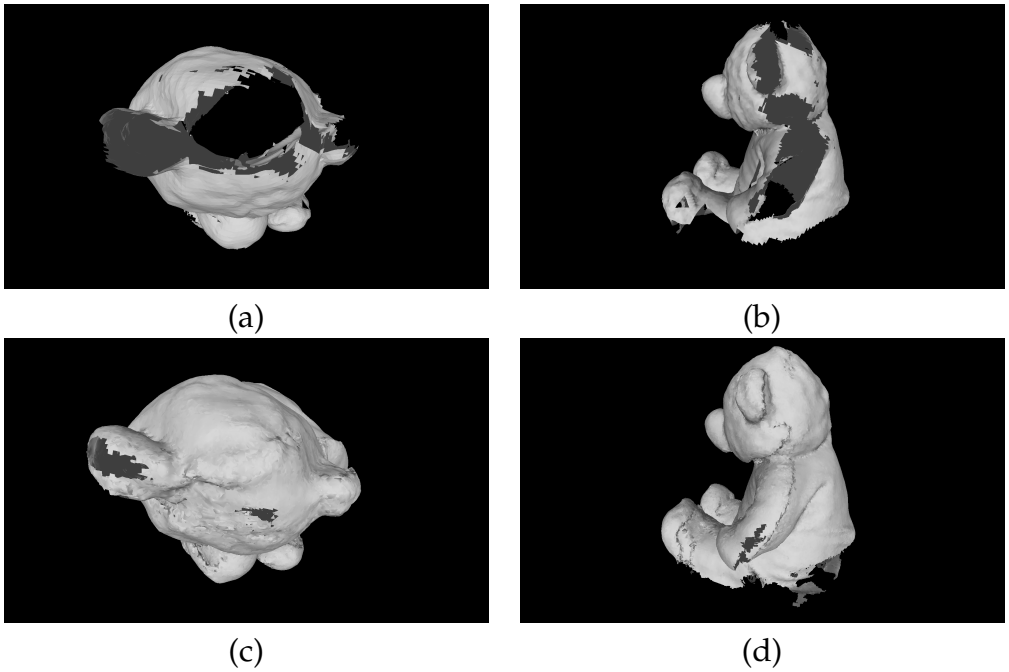


Figure 4.15: Teddy reconstruction with InfiniTAM versus the system proposed in this work:
(a, b) InfiniTAM.
(c, d) Approach of this work.

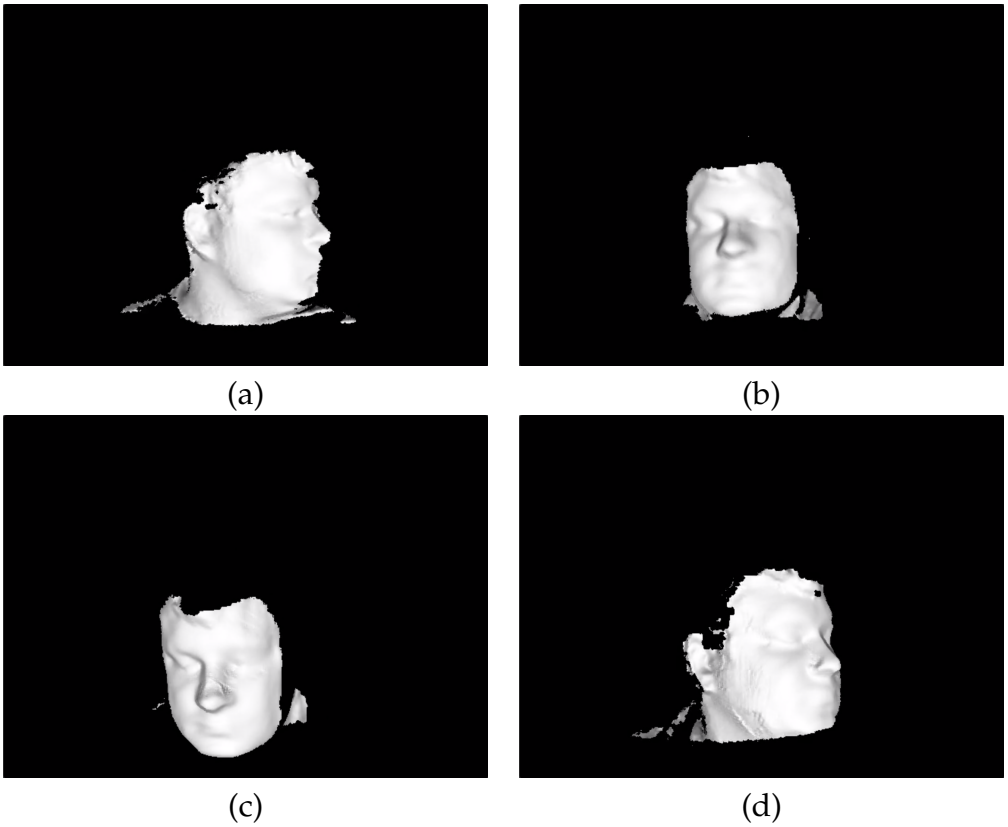
4.8 QUANTITATIVE RESULTS

In this section the performance of the proposed system is evaluated quantitatively with respect to both pose estimation accuracy and reconstruction accuracy.

For pose evaluation, the *3D Object Reconstruction* subset of the *RGB-D SLAM Dataset and Benchmark* [106] is used[1]. The objects of interest in the dataset vary largely in both geometry and appearance. As with the quantitative evaluation presented in Section 3.5, the ATE is the primary metric of interest.

It can be seen from Table 4.1 that on the *freiburg3_cabinet* and *freiburg3_teddy* sequences the approach proposed in this work yields a marked improvement in ATE over that of the standard KinectFusion pipeline with object segmentation.

[1] Technical University of Munich, RGB-D SLAM Dataset and Benchmark.
https://vision.in.tum.de/data/datasets/rgbd-dataset



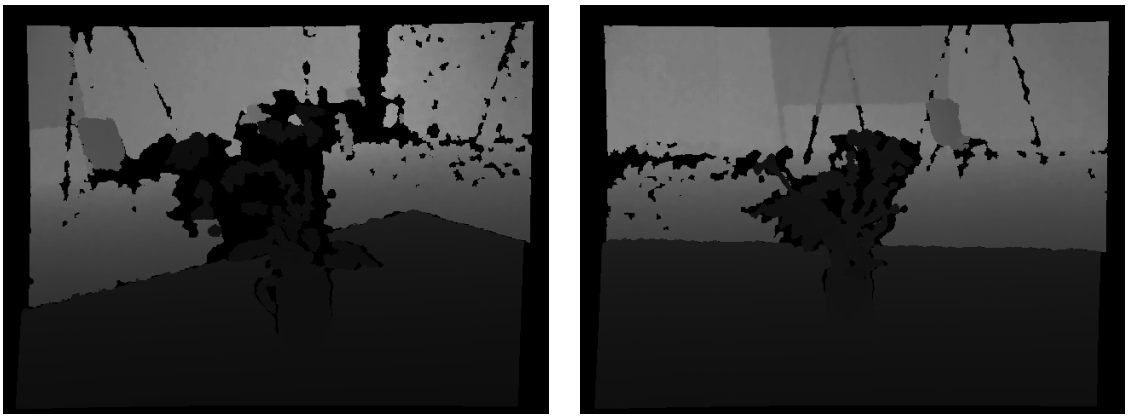
(a) (b) (c) (d)

Figure 4.16: Tracking and reconstructing a face at various poses.

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

Table 4.1: ATE results achieved by the proposed approach versus InfiniTAM with object segmentation.

The remaining sequences in the *3D Object Reconstruction* subset of the *RGB-D SLAM Dataset and Benchmark* [106] did not yield an interpretable reconstruction; these sequences did not reconstruct in *neither* the proposed approach, *nor* the baseline approach (InfiniTAM). In the case of the *freiburg2_metallic_sphere (2)* sequences, the object of interest is purely spherical in geometry. Such a shape is problematic for ICP like tracking algorithms due to it's inherent ambiguity. Ambiguous geometry was also troublesome for the *freiburg2_coke* sequence in *both* systems. The *freiburg2_flowerbouquet (brownbg)* sequences also proved troublesome for *both* approaches. When inspecting the depth stream for this sequence it is apparent that much of the flower bouquet does not have consistent depth data, in some cases with a large amount of descriptive geometrical information missing. An example of such frames from the sequence are given in Figure 4.17

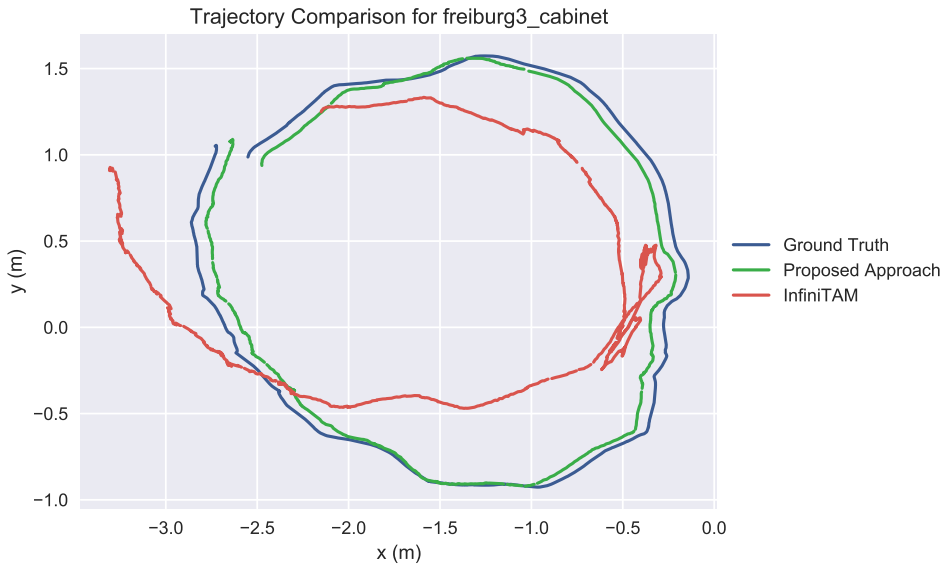


(a) (b)

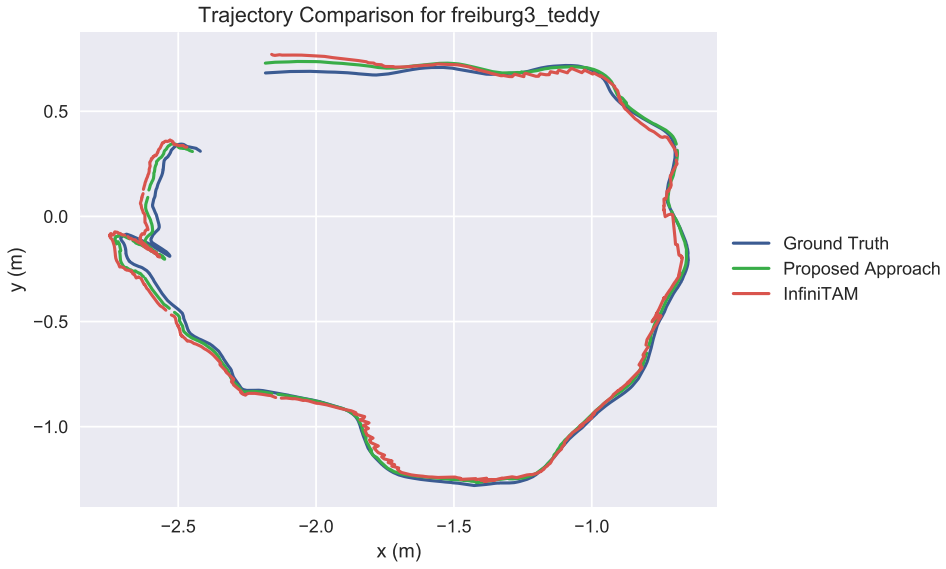
Figure 4.17: (a) Starting frame with much missing depth data.
(b) End frame with much more depth data.

The trajectories of the proposed object reconstruction system are evaluated on the *freiburg3_cabinet* and *freiburg3_teddy* sequences of Table 4.1, and are plotted versus the ground truth trajectories in Figure 4.18.

The proposed system is additionally evaluated quantitatively with respect to reconstruction quality. Reference models are obtained by reconstructing both the object of



(a)



(b)

Figure 4.18: Trajectory plots for the following sequences:
(a) *freiburg3_cabinet*.
(b) *freiburg3_teddy*.

interest and it's surrounding scene (for maximal pose estimation accuracy), followed by a manual segmentation of the object of interest in 3-space. To quantify the reconstruction quality the Hausdorff Distance [113] for subsets of metric spaces is used, where in this case the metric space is Euclidean. The Hausdorff Distance is defined as follows in Equation 4.43

$$d_H(X,Y) = \max\left[\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\right] \tag{4.43}$$

In Equation 4.43, X is the ground truth dense SLAM reconstruction, Y is the reconstruction outputted by the proposed system and $d(.)$ is the Euclidean distance.

The resultant quantitative comparisons may be found in Table 4.2.

Sequence	*Min Dist* (m)	*Max Dist* (m)	*Mean Dist* (m)	*RMS* (m)
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 4.2: Minimum, maximum, mean and RMSE distances between the reconstructions yielded by the proposed system and the baseline.

As can be seen by the similarity measures presented, the proposed system is capable of yielding reconstructions to a high quality despite the markedly more difficult pose estimation scenario of utilising the observations of a single object rather than those of an entire scene. It can be seen that the presented output reconstructions are geometrically close to those reconstructed with a dense SLAM system [30] following the KinectFusion [1] pipeline that is modelling and tracking the entire scene and thus has much more geometrical data with which to estimate pose.

Figure 4.19 presents renderings of the *Teddy*, *Brain*, *Chair* and *Dinosaur Head* sequences textured on the Hausdorff Distance between the output reconstructions of the

presented system and the base reconstructions extracted from the KinectFusion like pipeline.

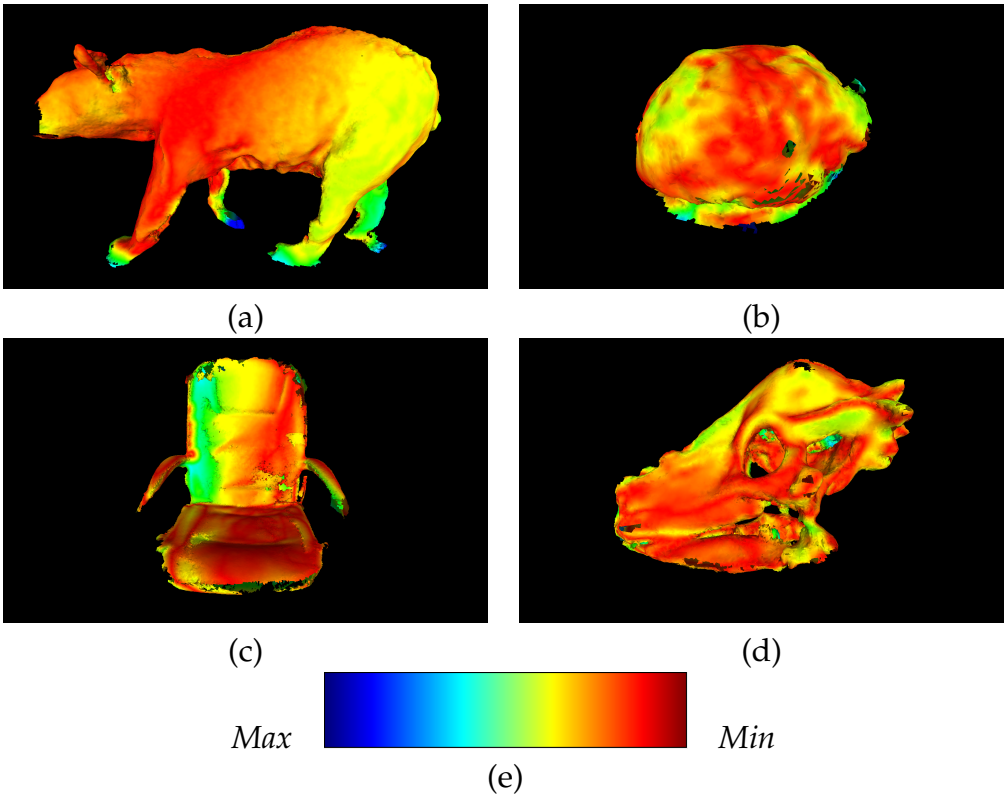


Figure 4.19: (a, b, c, d) Renderings of the sequences evaluated in Table 4.2 textured on Hausdorff Distance.
(e) Intensity mapping corresponding to the minimum and maximum values given in Table 4.2.

4.9 PERFORMANCE EVALUATION

This section provides an analysis of the performance of the proposed approach. The research objectives outlined in Section 1.4 highlight the requirement that commodity hardware be used for ease of applicability. As such, the proposed system has been designed with real time performance in mind. Figure 4.20 gives performance statistics on a subset of the evaluation sequences.

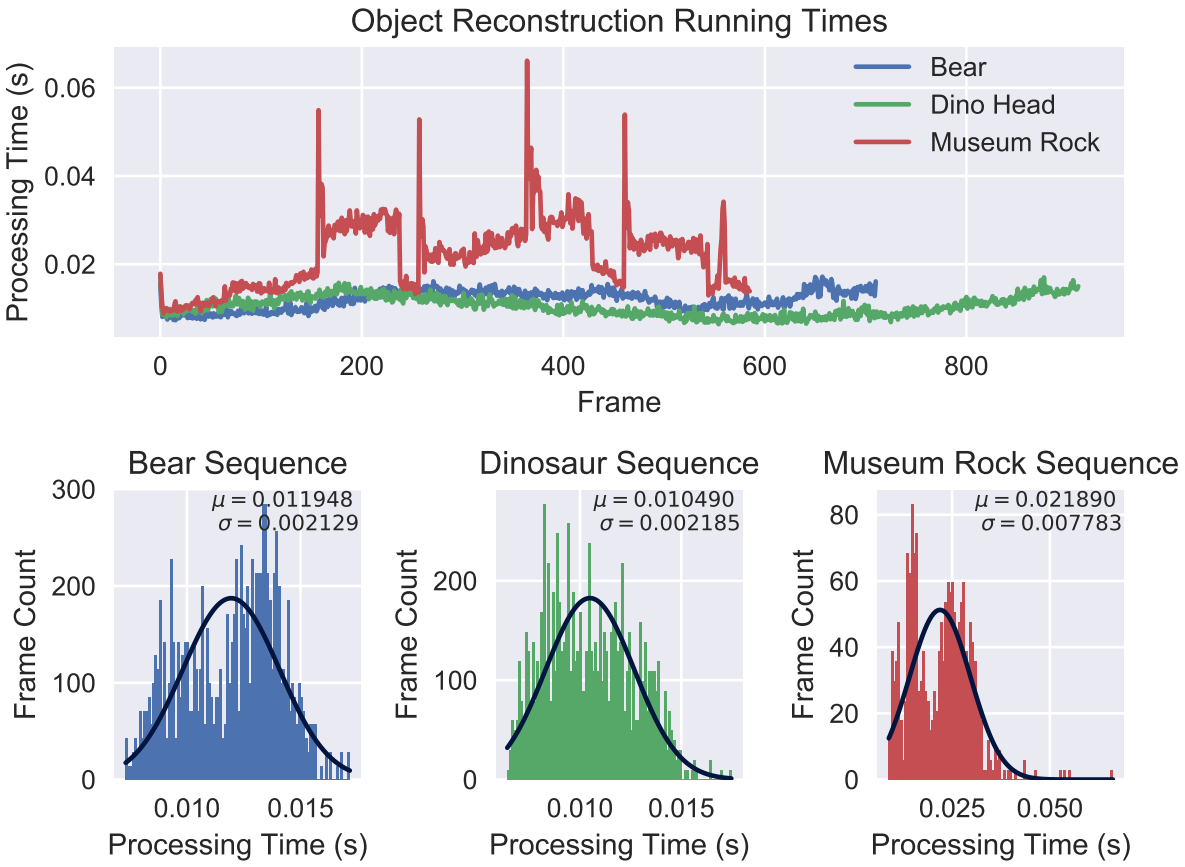


Figure 4.20: Performance of the proposed approach on the *Bear*, *Dinosaur Head* and *Museum Rock* sequences.

It can be seen in Figure 4.20 that for the *Bear* and *Dinosaur Head* sequences there are mean per-frame running times of 0.011948 and 0.010490 seconds, respectively. It can also be seen that the Standard Deviations of the frame running times are small (0.002129 and 0.002185 seconds respectively). As these sequences focus on objects with sufficient geometry to converge to a consistent model in a relatively short period, the fluctuations in runtime due to online correction are minimal. However, it may be seen in Figure 4.20 that for the *Museum Rock* sequence, the runtime is less stable. This sequence demonstrates a mean per frame runtime of 0.021890 seconds, with a larger standard deviation of 0.007783 seconds. This behaviour is likely to be caused by the lack of descriptive geometry of the object, resulting in less stable pose estimation. As such, for this sequence, the online correction procedure requires more iterations to converge to a consistent model. The spikes in Figure 4.20 reflect the practical implications of this; though correction is performed asynchronously, occasionally the correction thread blocks the main pipeline.

It can be seen in Figure 4.20 that for the *Bear* and *Museum Rock* sequences that there is a multimodality in the timing distributions. On further analysis it is apparent that this is due to the non-consistent sizing of the reconstructed subvolumes. Thus, a subvolume with high occupancy in the view frustum requires more processing time due to the increase of surface data. This is apparent when a new subvolume is created when viewing a smaller portion of an object. For example, if a new subvolume is created whilst viewing the side of the *Bear*, that subvolume will naturally contain a larger amount of distance and surface data than the case of viewing from the front, for example.

However, in all cases the system demonstrates real time performance with frame-rates of $\approx$ 84Hz, $\approx$ 95Hz and $\approx$ 45Hz for the *Bear*, *Dinosaur Head* and *Museum Rock* sequences, respectively. As with the performance evaluation of Section 3.6, the quoted runtimes in this section are excluding on-screen rendering with GLUT.

4.10 SUMMARY

It is evident from Sections 4.7 and 4.8 that the proposed approach is capable of densely reconstructing a range of rigid objects, providing closed and consistent object models, versus the state-of-the-art approach of *Ren et al* [74], which failed to reconstruct any of the objects with non-trivial geometry that it was evaluated on. Recall from Section 1.4 that a primary research objective of this work is to provide a means to reconstruct arbitrary object in a globally consistent manner.

Additionally, the proposed approach shows an improvement over direct segmentation in image space of the object of interest in terms of pose estimation. When evaluating the output reconstructions of the proposed system versus those extracted from a standard *KinectFusion* like pipeline (post reconstruction), the proposed system is capable of producing reconstructions that are geometrically similar. Again, noting the research objective of producing reconstructions of a comparable quality to the state of the art, as outlined in Section 1.4.

The probabilistic formulation proposed in this work is intended to be easily generalised, such that it may be adapted for a range of use cases. For example, when utilising stereo sensors where a noise model may not be known a-priori, the framework allows for a suitable prior to be used. Additionally, the proposed approach is intentionally decoupled from the object segmentation model, allowing for ease of adaptation for multiple object scenarios, or those where a continuous distribution over voxel labels is required.

In summary, the presented approach has demonstrated efficacy for the reconstruction of arbitrary objects in a geometrically consistent manner. As such, this work tackles one of the central research challenges outlined in Chapter 1, the difficulty in obtaining geometrically accurate 3D object data. As previously outlined, such data

collection is imperative for the advance from purely 2D features in learning systems, to the use of 3D geometry.

5

STEREO SHAPE AND POSE REGRESSION

This chapter introduces an approach to the simultaneous prediction of object shape and pose from stereo image pairs. A novel, data driven approach is taken that utilises the representational power of Convolutional Neural Networks with the generative power of Gaussian Processes.

5.1 INTRODUCTION

In the computer vision literature, there has been much research on the reconstruction of objects from observed points in a range image, such as those obtained with an RGBD sensor (like the Microsoft Kinect). As outlined in Chapter 4, progress has been made on the techniques used, such that globally consistent models of an object of interest can easily be obtained.

Though the traditional reconstruction paradigm is suitable for tasks such as 3D data collection, it is less applicable in scenarios where full sensor coverage of the object of interest is not possible, as outlined in the research problems in Section 1.4. When a full view of an object is not available, only a partial reconstruction may be built. Data driven, learning based approaches that yield reconstructions as predictions from a

generative model do not have such a limitation. However, for application as a direct replacement for manual reconstruction, pose estimation must also be performed.

There has been much progress on learning based methods for both pose estimation and shape prediction, as outlined in Section 2.5. However, much of this progress has been on the two problems when decoupled. In this work, an approach is proposed to solve the problem of simultaneous shape and pose prediction. The first difference in this work to those outlined in Chapters 3 and 4 is the use of monocular RGB frames versus RGBD. The central reason for the change of input data format is that the use of RGBD is prohibitive when considering outdoor environments where lighting conditions may prevent RGBD sensors, such as the Microsoft Kinect, from producing a meaningful depth map.

The second major deviation from the contributions of Chapters 3 and 4 is the removal of the dependence on temporally consistent input sequences. Rather, in this work the proposed system performs shape and pose prediction from an instantaneous, monocular RGB frame, requiring no temporal consistency in both the training and prediction phases. As such, there is no iterative solving for shape and pose per frame, rather, the process is framed as an instantaneous regression task. However, this work makes use of volumetric shape representations, as with the other works outlined in Chapters 3 and 4.

The approach outlined in this chapter makes use of CNN's and GPLVM's to jointly regress shape and pose given an RGB frame with an object (or objects) of interest present in the view frustum. The training of the model is part supervised (pose) and part semi-supervised (3D shape), requiring a ground truth 6DoF pose and a segmentation of the object(s) of interest. The proposed system regresses rotational and translational parameters directly from the neural network component, whilst the object shape drawn from a GPLVM prior. The use of a GPLVM for latent space embeddings of 3D shape allows for a simple way to learn complex 3D geometrical features, thus

simplifying the learning tasks of the neural network. In the proposed approach, the neural network component need only learn a mapping from observation to latent space, rather than from observation to full geometry for the shape of interest.

The system is based on the *Faster-RCNN* approach of *Ren et al* [114], trained on a multi-task loss over detection, classification, pose and shape. For training of shape, the loss is evaluated in terms of a rendering of the predicted shape under it's associated pose, against a ground truth segmentation. As such, the shape component of the model is trained in a weakly-supervised manner, as there is no ground truth 3D shape. Though such an approach presents a greater challenge than in the case of known ground truth for 3D shape, it does widen the applicability of the approach. In many *real world* scenarios it is not possible to obtain such ground truth data without first solving the problem of this work.

As outlined previously, the proposed approach differs from that of Chapters 3 and 4 in that it is data driven. Rather than predetermined algorithmic procedures being performed at each frame, the result of the input frames is derived from a trained model. As outlined in Sections 1.3 and 1.4, the research interest is in shape and pose prediction for objects in larger scale, *real world* environments. However, due to the difficulty faced with earlier experiments with a fully semi-supervised approach, a ground truth 6DoF pose is highly desirabe. As such, the dataset used in this work is the *Virtual KITTI (VKITTI)* dataset [115], a photo-realistic, synthetic rendering of the *The KITTI Vision Benchmark Suite* [96, 116, 117]. The advantage of using the aformentioned dataset is the balance between being a real world dataset and having a reliable and accurate ground truth for pose. The object class of interest in this work is cars.

The remainder of this chapter is structured as follows; Section 5.2 introduces the high level structure of the proposed model and the structure of it's neural network components. Section 5.3 introduces and derives the GPLVM that provides a generative model over shape. Following the introduction of the GPLVM, Section 5.4 outlines the

process of extracting a candidate shape for the observed object segmentation, from the GPLVM. Sections 5.5 and 5.6 outline the attitude representation of pose, and the shape rendering method used. Though, these are akin to those of Sections 3.2.1 and 3.2.3 respectively, the rendering stage is modified to introduce the property of differentiability. Sections 5.7 and 5.8 outline the loss function of the model and it's gradient for backpropagation, respectively. Sections 5.9 and 5.10 provide qualitative and quantitative results on the aforementioned dataset, respectively. Finally, Section 5.11 provides a summary of the approach taken in this work and the preliminary results of it's use.

5.2 ALGORITHMIC OVERVIEW

The proposed model takes as input a monocular RGB frame with the object(s) of interest present within the view frustum. From this RGB frame, a ResNet-101 [118] extracts feature descriptors which are mapped to a number of candidate object detections [4, 95]. For each detection, the extracted feature set is used to regress a latent space point for shape, and six $\mathbb{SE}(3)$ pose parameters, forming a Lie Algebra. The generated shape corresponding to the proposed latent space point is drawn from the distribution of the Gaussian Process (GP) prior conditioned on the latent space point. The form of the pose is the Lie Group mapping of the parameters, as given in Sections 3.2.1 and 4.4.2.

The loss function over the regressed shape and pose is quantified by rendering the generated shape under the predicted pose and computing a comparitive loss between the rendered region versus the provided semantic segmentation region. To maintain continuity, the rendering operation is formulated in a differentiable manner. As there

is often no ground truth 3D shape for *real world* data, training is performed in a weakly-supervised manner against a segmentation of the object of interest.

5.2.1 *Model Architecture*

The model of the proposed approach is of an RCNN [4] like architecture, where for a given input frame, a backbone network computes a feature map which is then provided to a Region Proposal Network (RPN), responsible for providing candidate Regions of Interest (RoI) in the input image. A high level depiction of this architecture is given in Figure 5.1. For the application outlined in this work, a true positive RoI corresponds to an object for which pose and shape should be regressed. From end to end, the model contains a ResNet-101 [118] backbone, an RPN, linear layers, batch normalisation layers, non-linear activations, a mapping from Lie algebra to Lie group (for pose), a GPLVM followed by an Inverse Discrete Cosine Transform (IDCT), a ray-caster and and a loss layer. As previously outlined, the loss consists of supervised and semi-supervised terms.

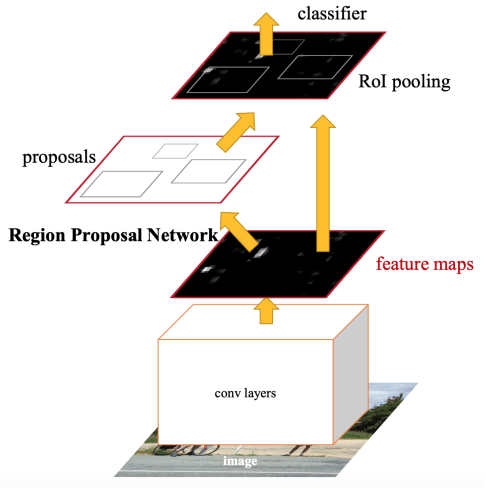


Figure 5.1: Faster R-CNN Architecture.[1]

The purpose of the ResNet-101 and RPN components is to extract, for each region proposal (object proposal), feature maps from the input frame that are descriptive of the object of interest in relation to the scene. Each proposed feature map is input into two sub-networks consisting of linear transforms and non-linear activations. The purpose of these sub-networks is to regress a latent space point for object shape, and a 6DoF pose parameter vector for the object pose. Following these two sub-networks is the aforementioned GPLVM and Lie algebra mapping. The GPLVM generates a posterior mean over shape for a given latent space point, whilst the Lie algebra mapping generates an $\mathbb{SE}(3)$ transform.

The IDCT decompresses the posterior mean output of the GPLVM to generate a valid SDF. Both the resultant SDF and $\mathbb{SE}(3)$ transform are passed to the ray-casting module, which generates a rendering for the candidate shape under the predicted pose. Finally, both this rendering and the ground truth segmentation mask are passed to the loss layer which computes a similarity metric between the two masks (ground truth and rendered), providing a tractible proxy loss for 3D shape. The topology of the proposed model is outlined in Figure 5.2.

[1] Image copyright: *Ren et al* [114].

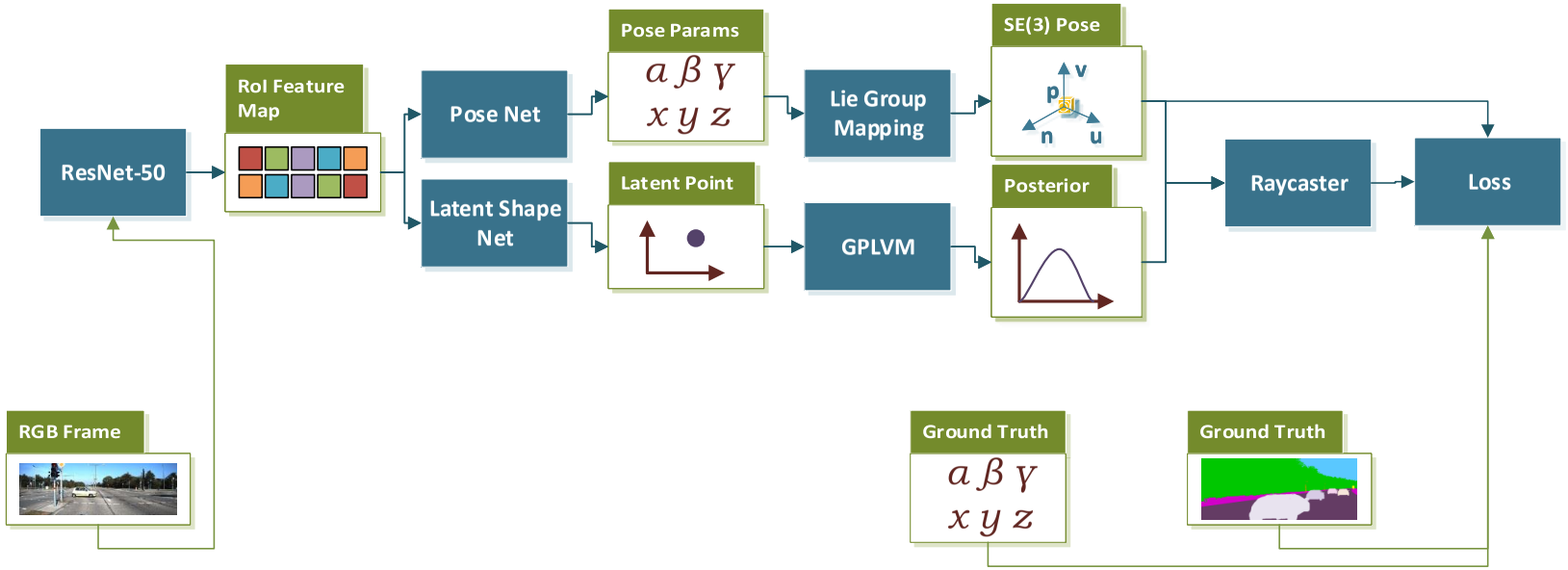


Figure 5.2: The proposed shape and pose prediction network. Note that the *ResNet-50*, *Shape Net* and *Latent Net* components have shared parameters.

Backbone Component

The ResNet architecture, introduced by *He at al* [118], is designed to overcome the convergence challenges of very "deep" neural network models. The central reformulation in the ResNet model is the notion that layers learn *residual functions* with respect to their inputs. Instead of a layer learning a mapping $\mathcal{H}(\mathbf{x})$ of it's input $\mathbf{x}$, it learns a *residual* mapping, as given in Equation 5.1.

$$\mathcal{F}(\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \mathbf{x} \tag{5.1}$$

$$\mathcal{F}(\mathbf{x}) + \mathbf{x} = \mathcal{H}(\mathbf{x}) \tag{5.2}$$

The formulation of Equation 5.1 is depicted in Figure 5.3.

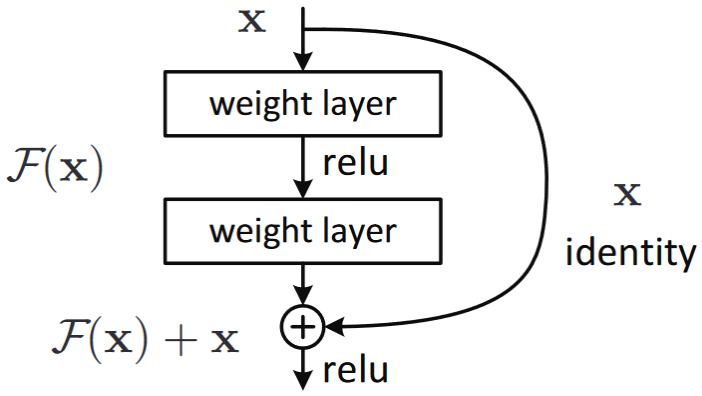


Figure 5.3: The central building block of the ResNet architecture[2], for some input $\mathbf{x}$ and transform $\mathcal{F}$.

Region Proposal, Classification and Bounding Box Components

As outlined in the *Faster-RCNN* [49] literature, the standard *R-CNN* [4, 95] architecture contains a region proposal network, providing pooled object RoI's in the input image, a

[2] Image copyright: *He et al* [118].

classification network and a bounding box regression network, as depicted in Figure 5.1. The standard *R-CNN* components are trained in a supervised manner with ground truth bounding boxes and classification labels. In the standard approach, the overall network loss is an aggregate of these individual task oriented losses.

Pose Regression Component

For each feature map obtained from an RoI object proposal, a 6DoF pose is regressed. For this pose regression, a separate pose branch of the network is used, taking as input for proposal n, a feature vector $\mathbf{x}_n \in \mathbb{R}^{1024}$ and providing output vector $\mathbf{y}_n \in \mathbb{R}^4$. The 4-dimensional output vector contains three rotational parameters and a single depth value. The architecture given in Figure 5.4 consists of linear transforms and nonlinear activation functions. For all but the first three output nodes, the Rectified Linear Unit (ReLU) is used. The first three output nodes for the objects rotational parameters however is the Hyperbolic Tangent (Tanh) scaled by 2π.

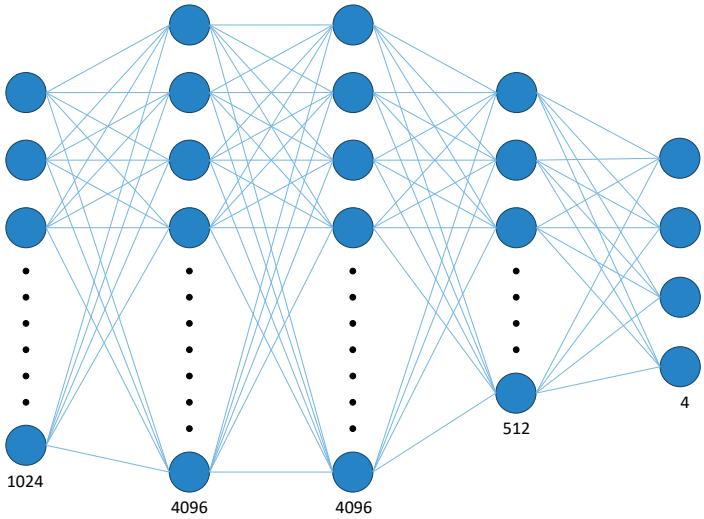


Figure 5.4: Pose regression network taking a 1024 dimensional input and providing a 4 dimensional output (three rotational parameters and a z coordinate).

It is not necessary to regress the entire 6DoF pose directly in the case of *known* camera intrinsic parameters. For a given depth value z, the x and y coordinates of a detected object may be recovered as follows in Equations 5.5 and 5.6 by making use of the 2D bounding box coordinates regressed by the network, as in Section 5.2.1.

To recover the 3D position of an object of interest, a simple camera unprojection may be performed. However, first the pixel coordinates of the object must be computed. Taking the object's x image location to be the centre of the x dimension of the bounding box, and the the y coordinate to be the bottom of the bounding box, the pixel coordinates may be computed as follows in Equations 5.3 and 5.4, for some bounding box $\mathbf{b}$.

$$\bar{x} = \mathbf{b}_x^{tl} + \frac{1}{2}\left[\mathbf{b}_x^{br} - \mathbf{b}_x^{tl}\right] \tag{5.3}$$

$$\bar{y} = \mathbf{b}_y^{br} \tag{5.4}$$

Where in Equations 5.3 and 5.4, the indices $\mathtt{tl}$ and $\mathtt{br}$ represent the top left and bottom right bounding box coordinates, respectively. Finally, the camera space x and y coordinates of the object of interest may be recovered as follows in Equations 5.5 and 5.6.

$$x = \frac{z}{f_x}\left[\bar{x} - c_x\right] \tag{5.5}$$

$$x = \frac{z}{f_y}\left[\bar{y} - c_y\right] \tag{5.6}$$

Latent Shape Space Regression Component

As outlined in Section 5.2, the GPLVM component of the network provides a generative model over 3D shape for the object class of focus in this work. However, to draw a 3D shape from the GP distribution, the GP must be conditioned on a 2D latent space point. The purpose of the shape regression network is to regress, for a given feature map corresponding to an RoI, a latent space point on which the GP is conditioned to predict a 3D shape descriptor.

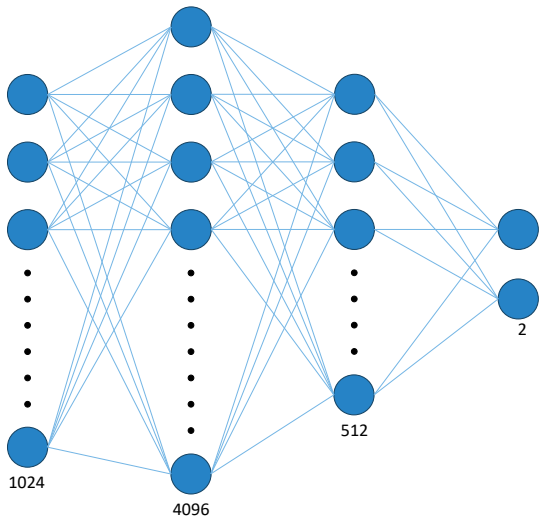


Figure 5.5: Shape regression network taking a 1024 dimensional input and providing a 3 dimensional output (a 2d point in the latent shape space, plus a scale factor).

The architecture of the shape network is analogous to the pose network of Section 5.2.1. Much like the pose network, the shape network consists of a set of linear transforms and nonlinear activations. The domain of the GP shape prior is $[0, 1]$, so the two outputs of the shape network are given by the sigmoid activation function.

5.3 GAUSSIAN PROCESS LATENT VARIABLE MODEL

The use of a GPLVM for the embedding of 3D shape is motivated by the assumption that for a given category of object, cars for example, there is enough shared geometric structure that a comprehensive generative model may be formed. In the linear case, the GPLVM is equivalent to a probabilistic formulation of Principal Component Analysis (PCA), where a linear mapping from observed to hidden, latent space is derived.

Under a Bayesian formulation, the often intractable latent variables pertaining to the linear mapping may be marginalised, such that the latent embedding itself may be optimised directly. The given framework allows for complex, non-linear embeddings to be learnt via the use of kernel functions, though the optimisation in this case is often non-convex and highly non-linear.

A GP may be viewed as a prior distribution over continuous functions. The posterior may be obtained by conditioning on observed data, to obtain a function estimate for each data point. A visual example of a GP is given in Figure 5.6.

5.3.1 *Gaussian Process Marginal Likelihood*

The first step is to define a latent variable model of the form given in Equation 5.7.

$$P(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{W}, \beta) = P(\mathbf{Y} \mid \boldsymbol{W}\mathbf{X}^T, \beta^{-1}\mathbf{I}) \tag{5.7}$$

In Equation 5.7 the observed data $\mathbf{Y} \in \mathbb{R}^{N \times D}$ is mapped to a lower dimensionality manifold $\mathbf{X} \in \mathbb{R}^{N \times P}$, by parameters $\boldsymbol{W} \in \mathbb{R}^{P \times D}$ and variance β. In Equation 5.7, the latent variable model $P(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{W}, \beta)$ provides a linear mapping defined by the aforementioned latent variables $\boldsymbol{W}$. As $\boldsymbol{W}$ is unobservable and thus not directly

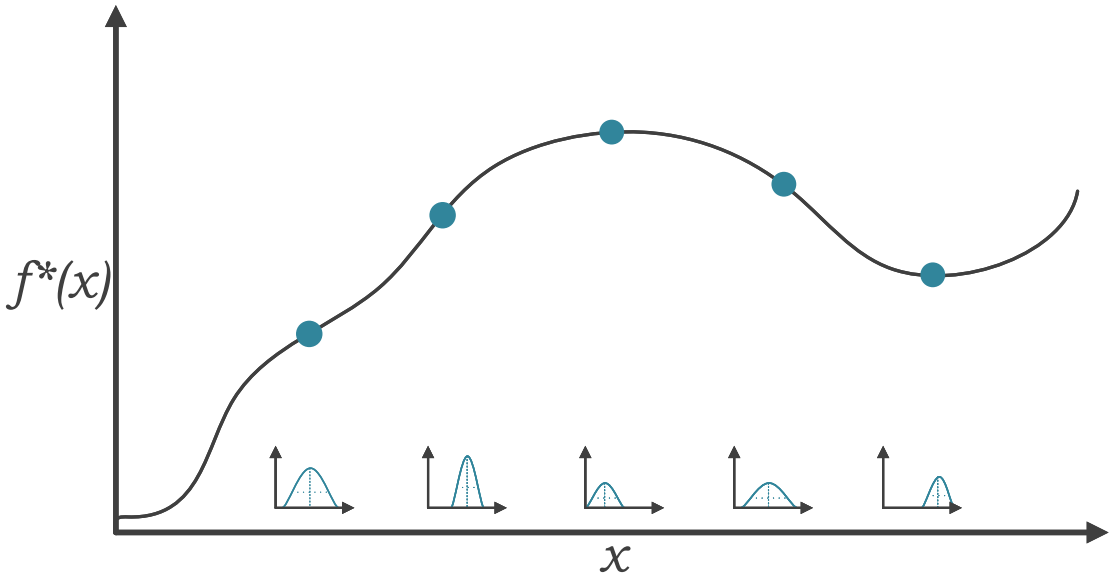


Figure 5.6: For each point along x, a function $\mathbf{f} \sim \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ drawn from a Gaussian Process $\mathcal{GP}$ is evaluated at x.

tractable, it may be analytically marginalised out, given a suitable likelihood and prior. The marginal likelihood of $\mathbf{X}$ is of the form outlined in Equation 5.8.

$$\mathrm{P}(\mathbf{Y} \mid \mathbf{X}, \beta) = \int \mathrm{P}(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{W}, \beta)\mathrm{P}(\boldsymbol{W})\mathrm{d}\boldsymbol{W} \tag{5.8}$$

In Equation 5.8, $\mathrm{P}(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{W}, \beta)$ is of a multivariate Gaussian form and $\mathrm{P}(\boldsymbol{W})$ is a multivariate Gaussian conjugate prior of the form $\mathcal{N}(\boldsymbol{W} \mid \mathbf{0}, \mathbf{I})$.

To find the marginal distribution outlined in Equation 5.8, it's form may first be simplified as follows in Equation 5.9.

$$P(\mathbf{Y} \mid \mathbf{X}, \beta) = \int \mathcal{N}(\mathbf{Y} \mid \boldsymbol{W}\mathbf{X}^T, \beta^{-1}\mathbf{I})\mathcal{N}(\boldsymbol{W} \mid \mathbf{0}, \mathbf{I})d\boldsymbol{W} \tag{5.9}$$

$$= \int \frac{1}{\sqrt{|2\pi\beta^{-1}\mathbf{I}|}} e^{-\frac{\beta}{2}(Y-WX)^T(Y-WX)} \frac{1}{\sqrt{|2\pi\mathbf{I}|}} e^{-\frac{1}{2}W^TW} d\boldsymbol{W} \tag{5.10}$$

$$= \frac{1}{\sqrt{|2\pi\beta^{-1}\mathbf{I}|}} \frac{1}{\sqrt{|2\pi\mathbf{I}|}} \int e^{-\frac{\beta}{2}(Y-WX)^T(Y-WX)-\frac{1}{2}W^TW} d\boldsymbol{W} \tag{5.11}$$

$$\propto \int e^{-\frac{\beta}{2}(Y-WX)^T(Y-WX)-\frac{1}{2}W^TW} d\boldsymbol{W} \tag{5.12}$$

$$\propto \int e^{-\frac{1}{2}\left[\beta(Y-X^TW)^T(Y-X^TW)+W^TW\right]} d\boldsymbol{W} \tag{5.13}$$

$$\propto e^{-\frac{\beta}{2}Y^TY} \int e^{-\frac{1}{2}\left[-\beta(Y^TX^TW)-\beta(W^TXY)^T+\beta W^TXX^TW+W^TW\right]} d\boldsymbol{W} \tag{5.14}$$

$$\propto e^{-\frac{\beta}{2}Y^TY} \int e^{-\frac{1}{2}\left[-2\beta Y^TX^TW+\beta W^TXX^TW+W^TW\right]} d\boldsymbol{W} \tag{5.15}$$

$$\propto e^{-\frac{\beta}{2}Y^TY} \int e^{-\frac{1}{2}\left[W^T(\beta XX^T+\mathbf{I})W-2\beta Y^TX^TW\right]} d\boldsymbol{W} \tag{5.16}$$

To make the integral over $\boldsymbol{W}$ tractable in Equation 5.9, the distribution $P(\mathbf{Y} \mid \mathbf{X}, \beta)$ must be a Gaussian; an exponential of a quadratic form. Completing the square in

$\boldsymbol{W}$ allows the marginal to be expressed in such a form. First, a change of variables is made, as in Equation 5.17.

$$\mathbf{A} = \beta \mathbf{X}\mathbf{X}^{\mathsf{T}} + \mathbf{I} \tag{5.17}$$

$$\mathbf{b} = \beta \mathbf{Y}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}} \tag{5.18}$$

The procedure to integrate over $\boldsymbol{W}$ and transform $\mathrm{P}(\mathbf{Y} \mid \mathbf{X}, \beta)$ into a valid multivariate Gaussian form is as follows in Equation 5.19.

$$\mathrm{P}(\mathbf{Y} \mid \mathbf{X}, \beta) \propto e^{-\frac{\beta}{2}\mathbf{Y}^{\mathsf{T}}\mathbf{Y}} \int e^{-\frac{1}{2}\left[\mathbf{W}^{\mathsf{T}}\mathbf{A}\mathbf{W} - 2\mathbf{b}\mathbf{W}\right]} \mathrm{d}\boldsymbol{W} \tag{5.19}$$

$$\propto e^{-\frac{\beta}{2}\mathbf{Y}^{\mathsf{T}}\mathbf{Y}} \int e^{-\frac{1}{2}\mathbf{W}^{\mathsf{T}}\mathbf{A}\mathbf{W} - 2\mathbf{b}\mathbf{W} - \mathbf{b}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{b}} \mathrm{d}\boldsymbol{W} \tag{5.20}$$

$$\propto e^{-\frac{\beta}{2}\mathbf{Y}^{\mathsf{T}}\mathbf{Y}} \int e^{-\frac{1}{2}\mathbf{W}^{\mathsf{T}}\mathbf{A}\mathbf{W} - 2\mathbf{b}\mathbf{A}\mathbf{A}^{-1}\mathbf{W} + \mathbf{b}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{A}\mathbf{A}^{-1}\mathbf{b}} \mathrm{d}\boldsymbol{W} \tag{5.21}$$

$$\propto e^{-\frac{\beta}{2}\mathbf{Y}^{\mathsf{T}}\mathbf{Y}} \int e^{-\frac{1}{2}\left[\left(\mathbf{W} - \mathbf{A}^{-1}\mathbf{b}\right)^{\mathsf{T}}\mathbf{A}\left(\mathbf{W} - \mathbf{A}^{-1}\mathbf{b}\right) - \mathbf{b}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{b}\right]} \mathrm{d}\boldsymbol{W} \tag{5.22}$$

$$\propto e^{-\frac{\beta}{2}\mathbf{Y}^{\mathsf{T}}\mathbf{Y}} e^{-\frac{1}{2}\left[\sqrt{|2\pi\mathbf{A}|} - \mathbf{b}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{b}\right]} \tag{5.23}$$

$$\propto e^{\frac{1}{2}\left[\beta\mathbf{Y}^{\mathsf{T}}\beta\mathbf{Y} - \mathbf{b}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{b}\right]} \tag{5.24}$$

$$\propto e^{-\frac{1}{2}\mathbf{Y}^{\mathsf{T}}\left(\beta\mathbf{I} - \beta^2\mathbf{X}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{X}\right)\mathbf{Y}} \tag{5.25}$$

With the distribution $P(\mathbf{Y} \mid \mathbf{X}, \beta)$ derived as being proportional to an exponentiated quadratic form as in Equation 5.19, it is clear that the inverse covariance matrix $\mathbf{\Sigma}^{-1}$ of the Gaussian distribution corresponding to $P(\mathbf{Y} \mid \mathbf{X}, \beta)$ is as follows in Equation 5.26.

$$\mathbf{\Sigma}^{-1} = \beta \mathbf{I} - \beta^2 \mathbf{X}^\top \mathbf{A}^{-1} \mathbf{X} \tag{5.26}$$

$$= \beta \mathbf{I} - \beta^2 \mathbf{X}^\top \left(\beta \mathbf{X} \mathbf{X}^\top + \mathbf{I} \right)^{-1} \tag{5.27}$$

To obtain the covariance matrix of the distribution $P(\mathbf{Y} \mid \mathbf{X}, \beta)$, the form of it's inverse may be simplified by the use of the matrix inversion lemma (also known as the Woodbury Identity) [119]. First making a change of variables in the Woodbury Identity as in Equation 5.28.

$$\mathbf{A} = \beta^{-1} \mathbf{I} \tag{5.28}$$

$$\mathbf{C} = \mathbf{I} \tag{5.29}$$

$$\mathbf{U} = \mathbf{X}^\top \tag{5.30}$$

$$\mathbf{V} = \mathbf{X} \tag{5.31}$$

in

$$(\mathbf{A} + \mathbf{U}\mathbf{C}\mathbf{V}) = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} \tag{5.32}$$

The simplified form of $\mathbf{\Sigma}^{-1}$ is thus given as follows in Equation 5.33.

$$\mathbf{\Sigma}^{-1} = \beta \mathbf{I} - \beta^2 \mathbf{X}^\top \left(\beta \mathbf{X} \mathbf{X}^\top + \mathbf{I} \right)^{-1} \tag{5.33}$$

$$= \beta^{-1} \mathbf{I} + \mathbf{X}^\top \mathbf{X} \tag{5.34}$$

It follows from Equation 5.33, that the covariance matrix $\mathbf{\Sigma}$ takes the form of Equation 5.35.

$$\mathbf{\Sigma} = (\mathbf{\Sigma}^{-1})^{-1} \tag{5.35}$$

$$= \mathbf{X}^T\mathbf{X} + \beta^{-1}\mathbf{I} \tag{5.36}$$

As such, the form of the normalized marginal likelihood $P(\mathbf{Y} \mid \mathbf{X}, \beta)$ of $\boldsymbol{W}$ is given in Equation 5.37.

$$P(\mathbf{Y} \mid \mathbf{X}, \beta) = \mathcal{N}(\mathbf{Y} \mid \mathbf{0}, \mathbf{X}^T\mathbf{X} + \beta^{-1}\mathbf{I}) \tag{5.37}$$

5.3.2 *Gaussian Process Fitting*

As outlined in Section 5.3, the given formulation of the marginal likelihood in Equation 5.37 can be optimised directly for the latent embedding $\mathbf{X}$. For a mapping from $\mathbb{R}^{N\times D}$ space to $\mathbb{R}^{N\times P}$ space (for $P < D$), the latent embedding $\mathbf{X}$ may be initialized by applying an orthogonal linear transform to the observed data and reducing dimensionality. One such approach is to apply PCA to the observed data, taking the first P reverse sorted eigenvalues of the covariance matrix. Additionally, in the non-linear case, where the covariance matrix $\mathbf{\Sigma}$ is generated by a given kernel function κ, the hyperparameters of κ may also be optimised.

To find the most probable latent space embedding, the latent variables $\mathbf{X}$ may be found by directly optimising the marginal likelihood of Equation 5.37. As such, the

natural logarithm may also be optimised for the latent variables $\mathbf{X}$ and is given in Equation 5.38.

$$\mathcal{L} = -\frac{DN}{2}\ln(2\pi) - \frac{D}{2}\ln(|\mathbf{\Sigma}|) - \frac{1}{2}\text{tr}(\mathbf{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^{T}) \tag{5.38}$$

$$= -\frac{DN}{2}\ln(2\pi) - \frac{D}{2}\ln(|\mathbf{\Sigma}|) - \frac{1}{2}\mathbf{Y}^{T}\mathbf{\Sigma}\mathbf{Y} \tag{5.39}$$

The gradient of the log marginal of Equation 5.38 is derived as follows in Equation 5.40.

$$\frac{\partial\mathcal{L}}{\partial\mathbf{X}} = -\frac{1}{2}\left[\left(D\frac{\partial}{\partial\mathbf{\Sigma}}\ln\left(|\mathbf{\Sigma}|\right)\right)\frac{\partial\mathbf{\Sigma}}{\partial\mathbf{X}} + \left(\frac{\partial}{\partial\mathbf{\Sigma}}\mathbf{Y}^{T}\mathbf{\Sigma}^{-1}\mathbf{Y}\right)\frac{\partial\mathbf{\Sigma}}{\partial\mathbf{X}}\right] \tag{5.40}$$

$$= -\frac{1}{2}\left[D\mathbf{\Sigma}^{-1}\frac{\partial\mathbf{\Sigma}}{\partial\mathbf{X}} - \mathbf{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^{T}\mathbf{\Sigma}^{-1}\frac{\partial\mathbf{\Sigma}}{\partial\mathbf{X}}\right] \tag{5.41}$$

$$= -\frac{1}{2}\left[D\mathbf{\Sigma}^{-1}2\mathbf{X} - \mathbf{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^{T}\mathbf{\Sigma}^{-1}2\mathbf{X}\right] \tag{5.42}$$

$$= -D\mathbf{\Sigma}^{-1}\mathbf{X} + \mathbf{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^{T}\mathbf{\Sigma}^{-1}\mathbf{X} \tag{5.43}$$

The gradient derived in Equation 5.40 holds for the case when $\mathbf{\Sigma} = \mathbf{X}^{T}\mathbf{X} + \beta^{-1}\mathbf{I}$. However, for $\mathbf{\Sigma} = \kappa(.)$ where κ is a given kernel function, the result of the derivation of Equation 5.40 is applicable. When substituting $\frac{\partial\mathbf{\Sigma}}{\partial\mathbf{X}}$ with $\frac{\partial\kappa}{\partial\mathbf{X}}$, the gradient is thus given in Equation 5.44.

$$\frac{\partial\mathcal{L}}{\partial\mathbf{X}} = -\frac{1}{2}\left[D\mathbf{\Sigma}^{-1}\frac{\partial\kappa}{\partial\mathbf{X}} - \mathbf{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^{T}\mathbf{\Sigma}^{-1}\frac{\partial\kappa}{\partial\mathbf{X}}\right] \tag{5.44}$$

It should be noted that the gradient $\frac{\partial\kappa}{\partial\mathbf{X}}$ may also be substituted for $\frac{\partial\kappa}{\partial\theta}$, for some hyperparameter θ of a given kernel κ. A common non-linear covariance kernel function

in the GP literature is the exponentiated quadratic [82], which takes the form given in Equation 5.45.

$$\kappa(\mathbf{x}_i, \mathbf{x}_j, \theta_0, \theta_1, \theta_2, \lambda) = \theta_0 e^{-\frac{\lambda}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2} + \theta_1 + \theta_2 \delta\Big(-\frac{\lambda}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\Big) \tag{5.45}$$

$$= \theta_0 e^{-\frac{\lambda}{2}\sum_n^D (\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2} + \theta_1 + \theta_2 \delta\Big(-\frac{\lambda}{2}\sum_n^D (\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2\Big) \tag{5.46}$$

The gradient of the Exponentiated Quadratic kernel of Equation 5.45, $\frac{\partial \kappa}{\partial \mathbf{x}_{i,n}}$ for the n^{th} variable of $\mathbf{x}_i$ can be derived as follows in Equation 5.47.

$$\frac{\partial \kappa}{\partial \mathbf{x}_{i,n}} = \frac{\partial}{\partial \mathbf{x}_{i,n}} \theta_0 e^{-\frac{\lambda}{2}\sum_{n=0}^D -\frac{\lambda}{2}(\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2} \tag{5.47}$$

$$= \frac{\partial}{\partial \mathbf{x}_{i,n}} \theta_0 e^{-\frac{\lambda}{2}\sum_{n=0}^D (\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2} \frac{\partial}{\partial \mathbf{x}_{i,n}} \sum_{n=0}^D -\frac{\lambda}{2}(\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2 \tag{5.48}$$

$$= \theta_0 e^{-\frac{\lambda}{2}\sum_{n=0}^D (\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2} \frac{\partial}{\partial \mathbf{x}_{i,n}} \sum_{n=0}^D -\frac{\lambda}{2}(\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2 \tag{5.49}$$

$$= \lambda \theta_0 e^{-\frac{\lambda}{2}\sum_{n=0}^D (\mathbf{x}_{i,n} - \mathbf{x}_{j,n})^2} (\mathbf{x}_{i,n} - \mathbf{x}_{j,n}) \tag{5.50}$$

Following the derivation of Equation 5.47, the remaining gradients of $\kappa(.)$ may be trivially derived, as in Equation 5.51.

$$\frac{\partial \kappa}{\partial x_{j,n}} = -\frac{\partial \kappa}{\partial x_{i,n}} \tag{5.51}$$

$$\frac{\partial \kappa}{\partial \theta_0} = e^{-\frac{\lambda}{2}\sum_{n=0}^{D}\left(x_{i,n}-x_{j,n}\right)^2} \tag{5.52}$$

$$\frac{\partial \kappa}{\partial \theta_1} = 1 \tag{5.53}$$

$$\frac{\partial \kappa}{\partial \theta_2} = 0 \tag{5.54}$$

$$\frac{\partial \kappa}{\partial \lambda} = -\frac{1}{2}\sum_{n=0}^{D}\theta_0\left(x_{i,n}-x_{j,n}\right)^2 e^{-\frac{\lambda}{2}\sum_{n=0}^{D}\left(x_{i,n}-x_{j,n}\right)^2} \tag{5.55}$$

5.4 LATENT SPACE SHAPE ESTIMATION

The form of the model outlined in Section 5.3.1, when trained as outlined in Section 5.3.2, defines a GP prior over the latent space embedding $\mathbf{X}$. When regressing a DCT compressed 3D shape, it is necessary to condition the DCT compressed 3D shape of a given latent space point on the GP prior. This conditioning yields a posterior mean estimation over 3D shape for a given latent space point.

The estimated posterior mean provides a Discrete Cosine Transform (DCT) compressed representation of an SDF shape volume $\mathbf{\Phi} \in \mathbb{R}^{N\times N\times N}$. The DCT compressed form of $\mathbf{\Phi}$ given by the aforementioned posterior mean is obtained by GP regression [119]. Finally, the true, uncompressed form of $\mathbf{\Phi}$ is obtained by taking the IDCT of the posterior mean. The granularity of the geometric properties captured by the GP

shape prior is governed by the number of DCT harmonics used in the compression and decompression processes at training and prediction.

5.4.1 *Shape Posterior Mean Estimation*

With the optimised latent variables $\mathbf{X}$, the formulation outlined in Equation 5.37 defines a GP prior over functions of $\mathbf{X}$, as follows.

$$\mathbf{f}(\mathbf{X}) \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{XX}}) \tag{5.56}$$

A GP prior may similarly be constructed for observed latent space points $\mathbf{L}$, as follows.

$$\mathbf{f}^{\star}(\mathbf{l}) \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{LL}}) \tag{5.57}$$

It follows from Equations 5.56 and 5.57 that the joint distribution over $\mathbf{X}$ and $\mathbf{L}$ can be formulated as follows in Equation 5.58.

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^{\star} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{XX}} & \boldsymbol{\Sigma}_{\mathbf{XL}} \\ \boldsymbol{\Sigma}_{\mathbf{LX}} & \boldsymbol{\Sigma}_{\mathbf{LL}} \end{bmatrix} \right) \tag{5.58}$$

The posterior of $\mathbf{f}^{\star}$ conditioned on $\mathbf{f}$ is given by the distribution in Equation 5.59.

$$\mathrm{P}(\mathbf{f}^{\star} \,|\, \mathbf{X}, \mathbf{L}, \mathbf{f}) = \mathcal{N}\left(\boldsymbol{\Sigma}_{\mathbf{LX}} \boldsymbol{\Sigma}_{\mathbf{XX}}^{-1} \mathbf{f}, \boldsymbol{\Sigma}_{\mathbf{LL}} - \boldsymbol{\Sigma}_{\mathbf{LX}} \boldsymbol{\Sigma}_{\mathbf{XX}}^{-1} \boldsymbol{\Sigma}_{\mathbf{XL}} \right) \tag{5.59}$$

As such, to obtain a posterior mean prediction for a latent space point $\mathbf{l}$, a draw from the distribution outlined in Equation 5.59 is taken as follows in Equation 5.60.

$$\mathbf{f}^{\star} \sim \mathrm{P}(\mathbf{f}^{\star} \,|\, \mathbf{X}, \mathbf{L}, \mathbf{f}) \tag{5.60}$$

Given the form of $P(\mathbf{f}^\star \mid \mathbf{X}, \mathbf{L}, \mathbf{f})$ in Equation 5.60, the posterior mean $\mathbf{f}^\star$ and variance $\mathbb{V}^\star$ are given as follows in Equation 5.61.

$$\mathbf{f}^\star = \boldsymbol{\Sigma}_{XL}^T \boldsymbol{\Sigma}_{XX}^{-1} \mathbf{Y} \tag{5.61}$$

$$\mathbf{f}^\star = \boldsymbol{\Sigma}_{LL} - \boldsymbol{\Sigma}_{LL}^T \boldsymbol{\Sigma}_{XL}^{-1} \boldsymbol{\Sigma}_{LL} \tag{5.62}$$

5.4.2 *Gaussian Process Posterior Mean Gradient*

Due to the gradient based optimisation of the approach outlined in this chapter, each component outlined in Figure 5.2 must be differentiable. The formulation of the posterior mean of Equation 5.61 is differentiable with it's gradient derived as follows in Equation 5.63.

$$\frac{\partial \mathbf{f}^\star}{\partial \mathbf{L}} = \frac{\partial}{\partial \mathbf{L}} \boldsymbol{\Sigma}_{XL}^T \boldsymbol{\Sigma}_{XX}^{-1} \mathbf{Y} \tag{5.63}$$

$$= \frac{\partial \boldsymbol{\Sigma}_{XL}^T}{\partial \mathbf{L}} \boldsymbol{\Sigma}_{XX}^{-1} \mathbf{Y} + \boldsymbol{\Sigma}_{XL}^T \frac{\partial \boldsymbol{\Sigma}_{XX}^{-1}}{\partial \mathbf{L}} \mathbf{Y} + \boldsymbol{\Sigma}_{XL}^T \boldsymbol{\Sigma}_{XX}^{-1} \frac{\partial \mathbf{Y}}{\partial \mathbf{L}} \tag{5.64}$$

$$= \frac{\partial \boldsymbol{\Sigma}_{XL}^T}{\partial \mathbf{L}} \boldsymbol{\Sigma}_{XX}^{-1} \mathbf{Y} \tag{5.65}$$

5.4.3 *Signed Distance Function Extraction*

As outlined in Section 5.4, the SDF volume $\boldsymbol{\Phi}$ is generated by taking the IDCT of the posterior mean of the GP prior conditioned on a given latent space point, as given by Equation 5.59. In the formulation presented in this work, the 2D latent point is generated by the shape network component following the RPN of Figure 5.2. The

output of the shape network is constrained to the interval $[0, 1]$ by the use of a sigmoid activation function.

The latent variable model outlined in Section 5.3 is a distribution over a latent space embedding of 3D shape. However, the 3D shape data on which the GP prior is modelled is compressed with the DCT. The intuition behind the use of the DCT is the property that the number of harmonics used in the transform impacts on the granularity of the resultant 3D shape geometry. It has been shown [120] that the lower frequency harmonics of the DCT capture general structure and shape, whilst higher frequency harmonics capture finer detailed geometric features. An example of a decompressed SDF of a car is given in Figure 5.7.

The voxels of an input SDF $\mathbf{V}$ under the DCT (for model training) are thus given as follows in Equation 5.66.

$$\boldsymbol{\Psi}_{x,y,z} = \mathbf{V}_{x,y,z}\left[\sum_{x=0}^{N-1}\cos\left[\frac{\pi}{N}\left[x+\frac{1}{2}\right]x\right]\sum_{y=0}^{N-1}\cos\left[\frac{\pi}{N}\left[y+\frac{1}{2}\right]y\right]\sum_{z=0}^{N-1}\cos\left[\frac{\pi}{N}\left[z+\frac{1}{2}\right]z\right]\right] \tag{5.66}$$

$$= \mathbf{V}_{x,y,z}\left[\sum_{x=0}^{N-1}\cos\left[\frac{\pi\left(x^2+\frac{x}{2}\right)}{N}\right]\sum_{y=0}^{N-1}\cos\left[\frac{\pi\left(y^2+\frac{y}{2}\right)}{N}\right]\sum_{z=0}^{N-1}\cos\left[\frac{\pi\left(z^2+\frac{z}{2}\right)}{N}\right]\right] \tag{5.67}$$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1}\mathbf{V}_{x,y,z}\cos\left[\frac{\pi\left(x^2+\frac{x}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(y^2+\frac{y}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(z^2+\frac{z}{2}\right)}{N}\right] \tag{5.68}$$

It follows that for a predicted posterior mean $\mathbf{f}^\star$, as outlined in Equation 5.60, the voxels of the SDF corresponding to the estimation $\mathbf{f}^\star$ may be extracted via the IDCT as follows in Equation 5.69.

$$\boldsymbol{\Phi}_{x,y,z} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1}\mathbf{f}^\star_{x,y,z}\cos\left[\frac{\pi\left(x^2+\frac{x}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(y^2+\frac{y}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(z^2+\frac{z}{2}\right)}{N}\right] \tag{5.69}$$

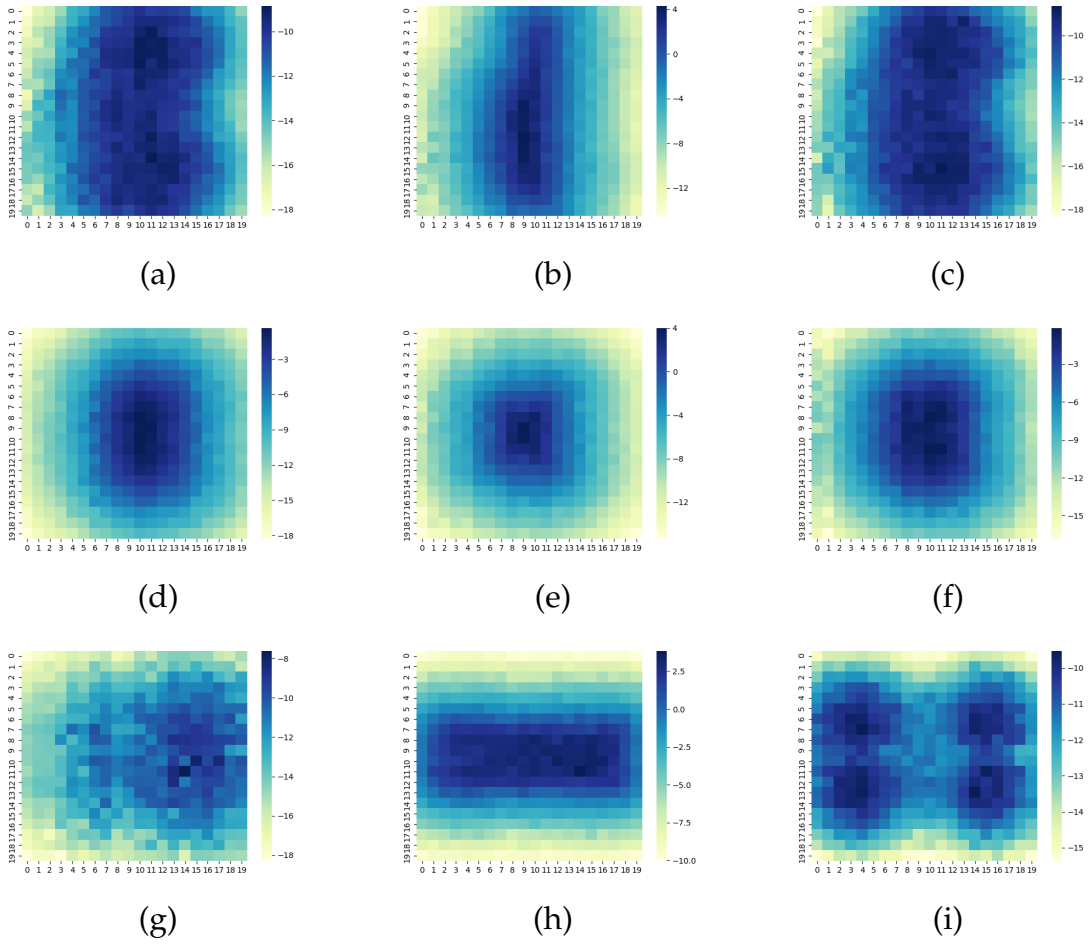


Figure 5.7: Slices of a Signed Distance Function embedding $\mathbf{\Phi} \in \mathbb{R}^{20\times20\times20}$ of a car at slices 0, 10 and 19:
(a, b, c) Along the x axis.
(d, e, f) Along the y axis.
(g, h, i) Along the z axis.

An example of how the geometric detail of the decompressed model varies with the number of harmonics used, N, is given in Figure 5.8.

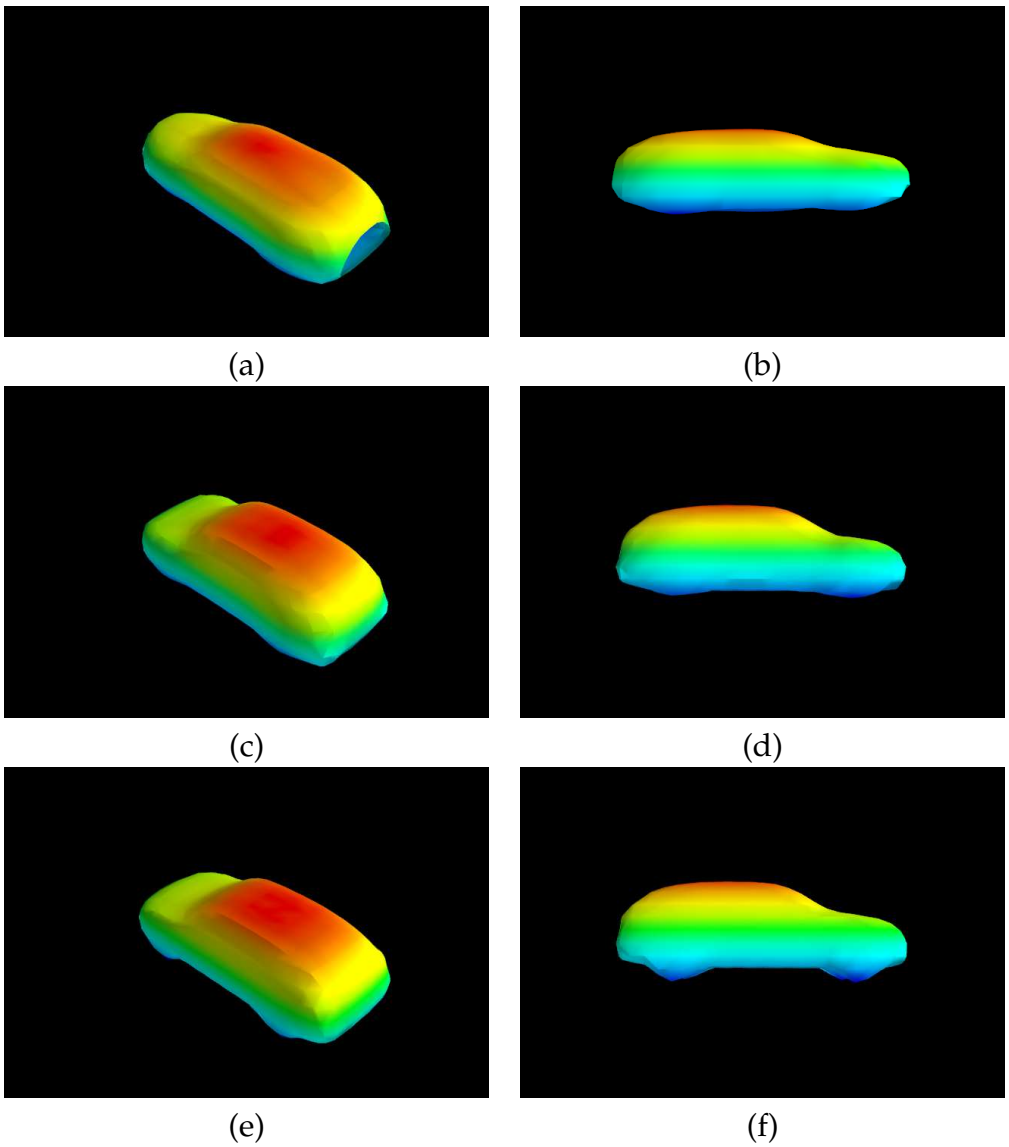


Figure 5.8: The effect of varying the number of DCT harmonics on the resultant 3D shape is shown for:
(a, b) One DCT harmonic.
(c, d) Three DCT harmonics.
(e, f) All DCT harmonics.

5.4.4 *Signed Distance Function Gradient*

The SDF $\mathbf{\Phi}$, generated by the IDCT given in Equation 5.69, must be differentiable with respect to the posterior mean $\mathbf{f}^\star$ for backpropagation training of the model. The gradient of the SDF $\mathbf{\Phi}$ outlined in Equation 5.69 is derived as follows in Equation 5.71. For notational clarity, the DCT coefficients are defined as follows in Equation 5.70

$$\zeta(x,y,z) = \cos\left[\frac{\pi\left(x^2+\frac{x}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(y^2+\frac{y}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(z^2+\frac{z}{2}\right)}{N}\right] \tag{5.70}$$

$$\frac{\partial \mathbf{\Phi}}{\partial \mathbf{f}^\star_{x,y,z}} = \frac{\partial}{\partial \mathbf{f}^\star_{x,y,z}} \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} \mathbf{f}^\star_{x,y,z}\zeta(x,y,z) \tag{5.71}$$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} \frac{\partial}{\partial \mathbf{f}^\star_{x,y,z}} \mathbf{f}^\star_{x,y,z}\zeta(x,y,z) \tag{5.72}$$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} \left[\left[\frac{\partial}{\partial \mathbf{f}^\star_{x,y,z}} \mathbf{f}^\star_{x,y,z}\right]\zeta(x,y,z) + \mathbf{f}^\star_{x,y,z}\frac{\partial \zeta}{\partial \mathbf{f}^\star_{x,y,z}}\right] \tag{5.73}$$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} \nabla \mathbf{f}^\star_{x,y,z}\zeta(x,y,z) \tag{5.74}$$

$$= \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} \nabla \mathbf{f}^\star_{x,y,z} \cos\left[\frac{\pi\left(x^2+\frac{x}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(y^2+\frac{y}{2}\right)}{N}\right]\cos\left[\frac{\pi\left(z^2+\frac{z}{2}\right)}{N}\right] \tag{5.75}$$

From Equation 5.71 it is evident that the partial derivative $\frac{\partial \mathbf{\Phi}}{\partial \mathbf{f}^\star_{x,y,z}}$ is trivial to compute, as the derivative of the IDCT is simply the IDCT of the derivative. Furthermore, the gradient of the posterior mean is similarly trivial to compute, as shown in Equation 5.63.

5.5 POSE ESTIMATION

The parameters of the $\mathbb{SE}(3)$ pose applied to the predicted shape are obtained from a fully connected component of the model, as outlined in Figure 5.2, as with the latent pose point of Section 5.4.3.

The three rotational parameters α, β and γ are predicted by the pose network component of the model on the interval $[-2\pi, 2\pi)$. The translation parameters however do not have their output range restricted (except the implicit nonnegativity enforced by the ReLU).

5.6 RENDERING

The dynamic SLAM and object reconstruction works of Chapters 3 and 4, both rely on ray-casting to obtain a rendering of a surface embedded within an implicit volume, under some estimated pose. The approach taken in this chapter is very similar.

Ray-casting is first and foremost used in the model outlined in Section 5.2 for visualisation purposes, to render the estimated shape under the estimated pose for an instantaneous scene view. In Chapters 3 and 4, the output rendering is also used for frame-wise evaluation and optimisation of the pose energy function. However, as the approach outlined in this chapter does not assume temporal consistency, the requirements for optimisation differ.

As the model outlined in Section 5.2 relies on the backpropagation of gradients for it's training, the rendering module must be differentiable for the backward pass. As such, this section outlines a simple formulation of differentiable ray-casting, inspired by the work of *Prisacariu et al* [65]. In addition to generating a shaded rendering,

the proposed approach generates a probabilistic map. For a review of the generic ray-casting algorithm, refer to Section 3.2.3.

By accumulating log probabilities of voxels visited on traversal of the ray, a differentiable representation of the process can be obtained. However, this relies on the use of a differentiable PDF. As with the approach to object reconstruction outlined in Chapter 4, the desired behaviour is that the distribution shall encode the probability of a voxel being very close to the isosurface embedded within the volume. As such, the log-sigmoid of Equation 4.14 is used. Thus, for a pixel location $[x, y]$, the output probabilistic value is as follows in Equation 5.76.

$$\mathcal{R}(x, y) = \sum_{v \in \mathcal{V}} \ln P(v) \tag{5.76}$$

In Equation 5.76, $\mathcal{V}$ is the set of voxels in the SDF $\mathbf{\Phi}$ that intersect the ray from the given pixel in the image frame. The gradient of the $\ln P(v)$ term in Equation 5.76 if the form given in Equation 4.33.

5.7 MULTIPLE TASK LOSS

The architecture outlined in Section 5.2.1 contains multiple branches following the RoI extraction, each with a different task and associated loss function. As the approach outlined in this work is based on the *Faster R-CNN* [114] architecture, the overall network loss is an aggregate of the individual task oriented losses, as outlined in Section 5.2.1. As such, the overall network loss may be defined as follows in Equation 5.77.

$$\mathcal{L} = \mathcal{L}_{cls}(z_{cls}, \mathbf{y}_{cls}) + \mathcal{L}_{bb}(z_{bb}, \mathbf{y}_{bb}) + \mathcal{L}_{p}(z_{p}, \mathbf{y}_{p}) + \mathcal{L}_{s}(z_{s}, \mathbf{y}_{s}) \tag{5.77}$$

In Equation 5.77, the four loss terms measure the models performance on classification, bounding box regression, pose regression and shape regression for predictions $\mathbf{z}$ and targets $\mathbf{y}$.

Note that the $\mathcal{L}_s$ is a proxy loss over shape, which measures the similarity between the rendering of the predicted 3D shape under the predicted pose.

5.7.1 *Pose and Shape Losses*

As the approach in this work assumes a known ground truth pose, a simple L_2 loss over the network predicted pose parameters and the ground truth is sufficient. As outlined in Section 5.2.1, only the z component of the translation vector must be regressed by the network as the x and y components may be recovered from the predicted bounding box coordinates.

As such, the pose loss term is given in Equation 5.78, for rotational parameter vector $\mathbf{z}_\rho$ and depth parameter z.

$$\mathcal{L}_p = \sum_{n=1}^{N} \left[\mathbf{y}_\rho - \mathbf{z}_\rho\right]^2 + \sum_{n=1}^{N} \left[\mathbf{y}_z - \mathbf{z}_z\right]^2 \tag{5.78}$$

The loss over shape is given by the pixel-wise binary cross entropy between the rendering of the current predicted shape under the current predicted pose, and the ground truth segmentation for the given detection. Note that the aforementioned rendering is the output of the differentiable raycast outlined in this work.

5.8 GRADIENTS FOR TRAINING WITH BACKPROPAGATION

With the gradients of the model components derived, the gradient update to the neural network component may be given. As the proposed model is optimised using

the backpropagation algorithm, the gradient update for layer $n - m$ is computed by applying the chain rule to layer n, successively working backwards to layer $n - m$. As such, the gradient for the non CNN components with respect to the CNN output for pose $\mathcal{O}_p$, is given as follows in Equation 5.79.

$$\frac{\partial \mathcal{L}}{\partial \mathcal{O}_p} = \frac{\partial \mathcal{L}}{\partial \mathcal{R}} \frac{\partial \mathcal{R}}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial \mathcal{O}_p} \tag{5.79}$$

In Equation 5.79, $\mathcal{L}$ is the loss given in Equation 5.77 of Section 5.7, $\mathcal{R}$ is the rendering of the predicted shape $\mathbf{\Phi}$ under the predicted pose $\mathbf{T}$ outlined in Section 5.6. $\mathbf{T}$ is the $\mathbb{SE}(3)$ pose outlined in Section 5.5. The gradient with respect to the latent shape point generated by CNN output $\mathcal{O}_l$ may also be found in the same manner and is given in Equation 5.80.

$$\frac{\partial \mathcal{L}}{\partial \mathcal{O}_l} = \frac{\partial \mathcal{L}}{\partial \mathcal{R}} \frac{\partial \mathcal{R}}{\partial \mathbf{\Phi}} \frac{\partial \mathbf{\Phi}}{\partial \mathbf{f}^\star} \frac{\partial \mathbf{f}^\star}{\partial \mathcal{O}_l} \tag{5.80}$$

In Equation 5.80, $\mathcal{L}$ and $\mathcal{R}$ are as in Equation 5.79. Φ is the SDF given by the IDCT, as given in Equation 5.69 of Section 5.4.3. $\mathbf{f}^\star$ is the posterior mean of the GPLVM, given by Equation 5.60.

5.9 QUALITATIVE RESULTS

This section provides a qualitative evaluation of the approach outlined in this chapter, both with respect to shape and combined shape and pose. For the experiments in this section and Section 5.10 two data sources were used. For the task of pose prediction, the *VKITTI* dataset was chosen for it's accurate ground truth and relative *real world* nature (photorealistic renderings of real world scenes). For the shape embedding and prediction component, the GPLVM model outlined in Section 5.3 was trained on a

collection of 3D shapes obtained from the online *ShapeNet* collection of 3D Computer Aided Design (CAD) models[3].

For the training of the generative model, 70 CAD models of cars were obtained from *ShapeNet*, chosen to cover a variety of common geometries for the class. The obtained CAD models were then manually processed to remove inner geometry, voxelised into binary occupancy volumes and finally converted to SDF's by the computation of the Eucledian distance transform to the binarised contour.

When training the latent shape point network, outlined in Section 5.2.1, a combination of the *VKITTI* dataset and the shape drawn from the generative model of Section 5.3 (derived from the *ShapeNet* CAD models) is used. As outlined, the *VKITTI* dataset contains both ground truth 6DoF pose and ground truth semantic segmentation masks. As outlined in Section 5.1, the loss of Section **??** is minimised over the rendering of the shape drawn from the GP under the predicted pose, with respect to the ground truth mask. The 3D CAD models were also reoriented to align with the *VKITTI* coordinate system of y pointing downwards, x pointing to the right and z directly forwards.

5.9.1 *Gaussian Process Latent Shape Embedding*

The GPLVM based embedding of 3D shape outlined in Section 5.3 was trained on a small but varied dataset of 3D CAD models. Qualitatively, the trained model appears capable of representing common geometries of the class of interest in this work. Given in Figure 5.9 are examples of shapes drawn from the shape distribution at uniformly sampled latent space points.

It can be seen in Figure 5.9, the trained shape model is capable of embdedding a variety of 3D geometries relevant to the problem addressed in this work. Note that

[3] ShapeNet: https://www.shapenet.org/

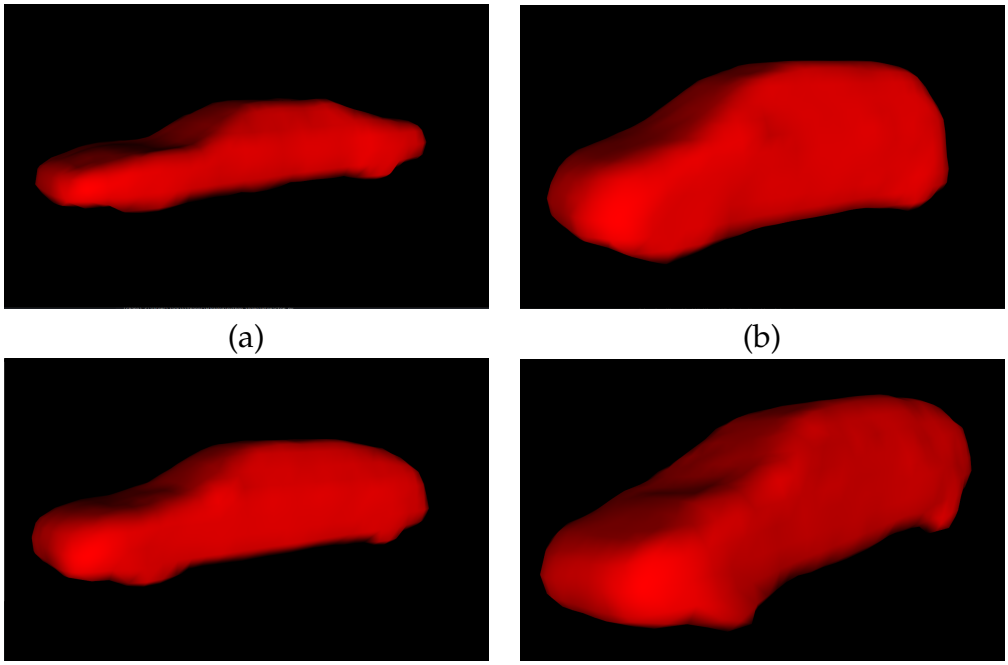
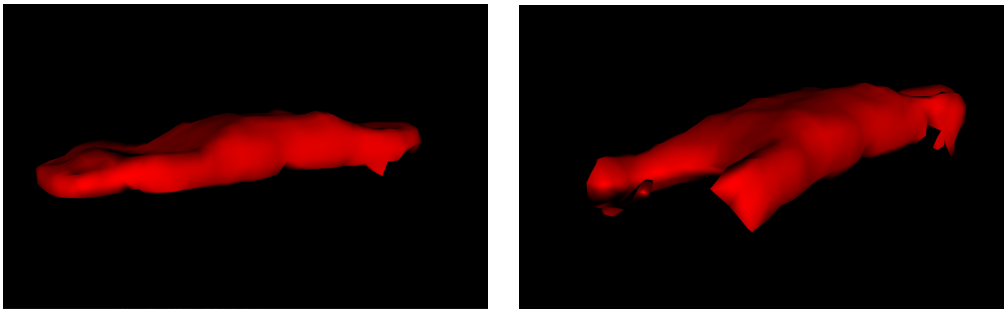


Figure 5.9: (a, b, c, d) IDCT extracted, random draws from the trained GP.

due to the probabilistic nature of the shape model, for a predicted shape (a posterior mean), there is additional covariance information for each estimated posterior mean. The shapes given in Figure 5.9 were drawn from a region of the latent space for which there is low covariance, thus higher probabilistic certainty. Figure 5.10 demonstrates shapes drawn from the shape distribution on regions of the latent space for which there is higher covariance and thus increased estimation uncertainty.

Given in Figure 5.11 is a visual representation of the uncertainty in estimated shape with respect to the latent space point on which the GP is conditioned. The examples provided in Figures 5.9 and 5.10 correspond to the low uncertainty (dark) and higher (light) uncertainty regions of Figure 5.11, respectively.



(a) (b)

Figure 5.10: (a, b) IDCT extracted, random draws from the trained GP in regions of the latent space that have high covariance.

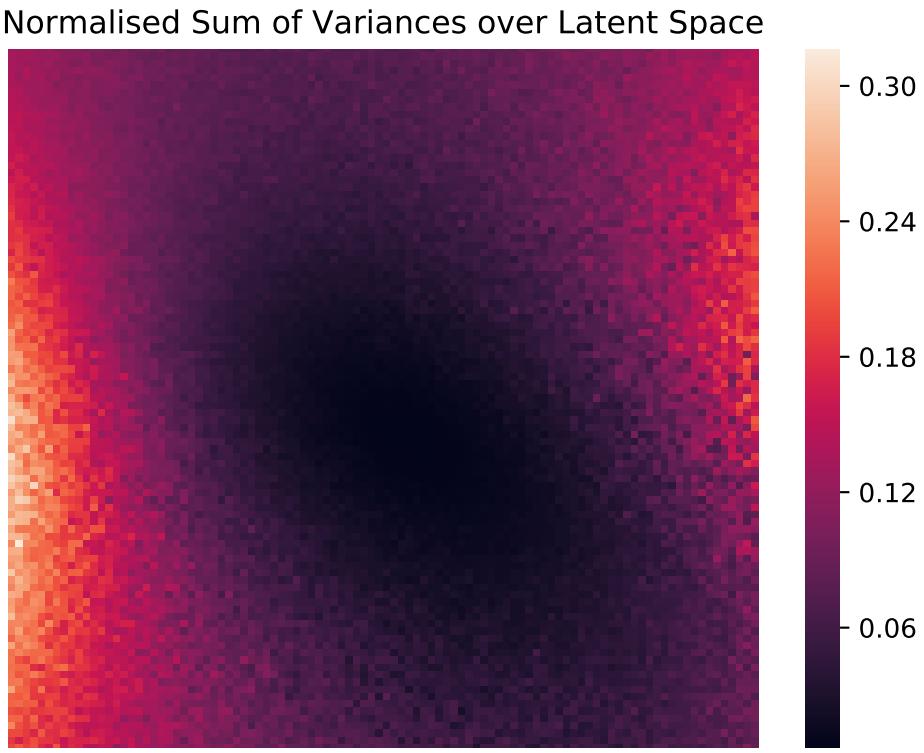


Figure 5.11: Normalised sum of the embedding variance for estimated shape on the latent space, for regularly sampled latent points on the interval $[-5, 5]$. Lower values indicate lower estimation uncertainty.

5.10 QUANTITATIVE RESULTS

Following the qualitative results presented in Section 5.9, this section provides a quantitative analysis of the performance of the model outlined in this work.

First, an overview of the models performance during training is provided, followed by quantitative results on the pose prediction performance of the model. Finally, a quantitative measure of the shape prediction performance of the model is provided. In the training sections that follow, the supervised and weakly supervised tasks have been trained and tested on a randomised $80:20$ training and validation data split, respectively. During training, gradient clipping was used to circumvent the problem of exploding gradients, with the clipping threshold set to 0.9.

5.10.1 *Transfer Learning for Car Detection on VKITTI*

The training of the model has been performed in multiple stages. Firstly, the standard *Faster R-CNN* is trained on the dataset outlined in Section 5.1 for the tasks of classification and bounding box regression. This stage of training may be thought of as a simple application of transfer learning for the specific problem domain outlined in this work; the backbone and RPN components of the network have been pretrained on the *Microsoft Common Objects in Context (COCO)* [121] dataset for the tasks of object classification and bounding box regression. Due to the specialisation of the detection requirements in this work, the RPN and feature extraction components must be retrained.

Figure 5.12 depicts the short retraining of the RPN and finetuning of the feature extraction network of the model outlined in Section 5.2. It can be seen that in a relatively short training period, the *Microsoft COCO* pretrained network rapidly becomes

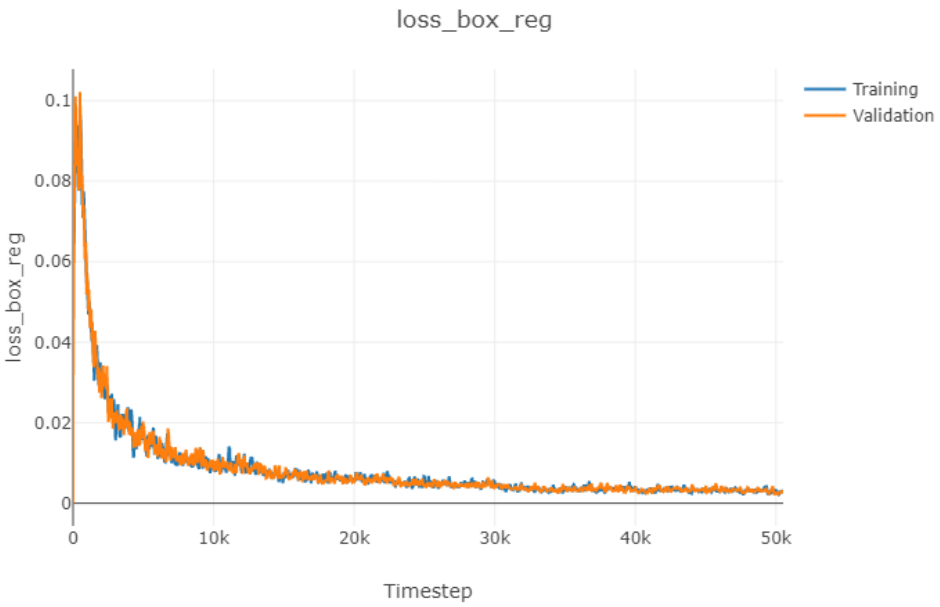


Figure 5.12: Bounding box regression loss on the *VKITTI* dataset for the standard *Faster R-CNN* network.

performant on the *VKITTI* dataset used in this work, for the task of specific object detection. Additionally, similar behaviour may be observed in Figure 5.13 for the coupled task of object classification.

For the training procedures of Figures 5.12 and 5.13, the Stochastic Gradient Descent (SGD) optimisation routine was used, with a learning rate of 0.001. The training session consisted of 50,000 batches, for which a batch consists of the class relevant ground truth detecions for a given frame.

5.10.2 *GPLVM Training*

The training procedure of the GPLVM outlined in Section 5.3 is an unsupervised routine and as such, no training/testing split of the 3D CAD model data introduced in Section 5.9 is required. It can be seen in Figure 5.14 that the GPLVM model requires

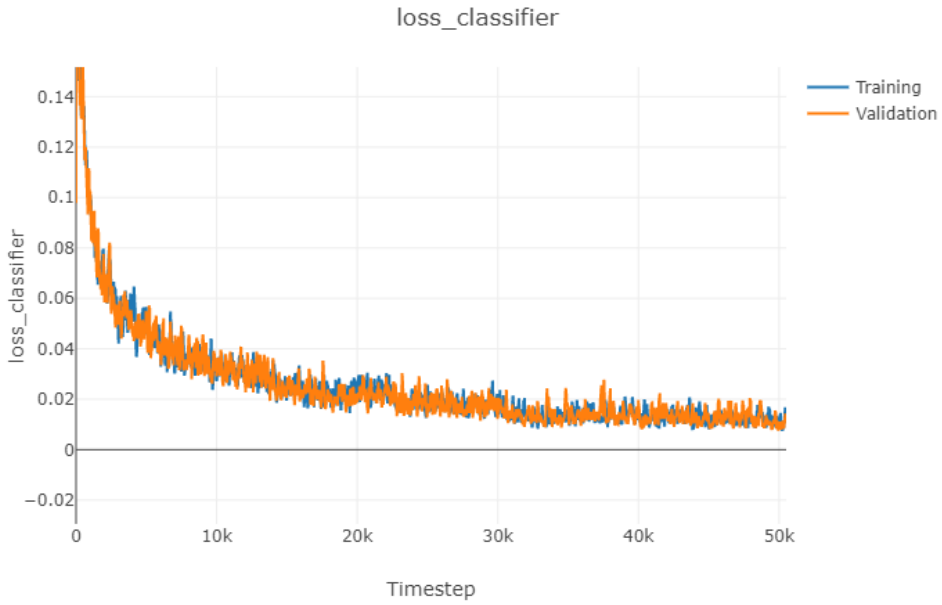


Figure 5.13: Classification loss on the *VKITTI* dataset for the standard *Faster R-CNN* network.

very few iterations to converge to an optimal latent embedding. The model was trained by minimising the negative log-likelihood, as is standard in the GP literature [119].

For the training procedure given in Figure 5.14, the Adam [122] optimiser was used with a base learning rate of 0.01. The model was trained for 1500 epochs, though as can be seen in Figure 5.14, the model converges considerably prior to this epoch limit.

5.10.3 *Supervised Training of Pose*

The training procedure of the pose regression network, outlined in Section 5.5 is supervised in nature, with the loss criterion given in Equation 5.78 measuring the MSE between the model predicted pose and the ground truth pose given in the *VKITTI* dataset. The training of the pose regression network was split over two training sessions, with differing learning rates. Figure 5.15 provides the training and validation error losses for the rotation and z coordinate for 150,000 training steps.

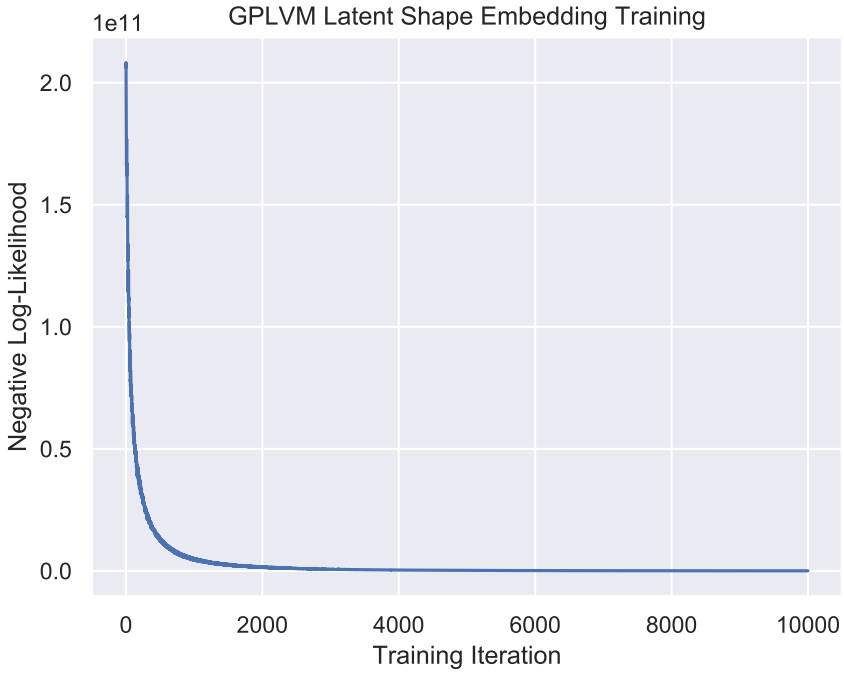
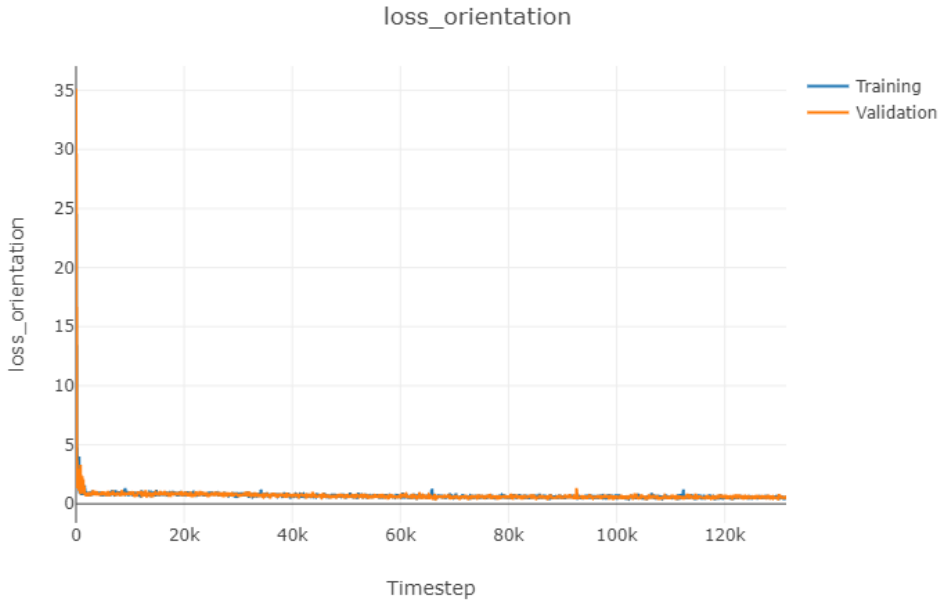


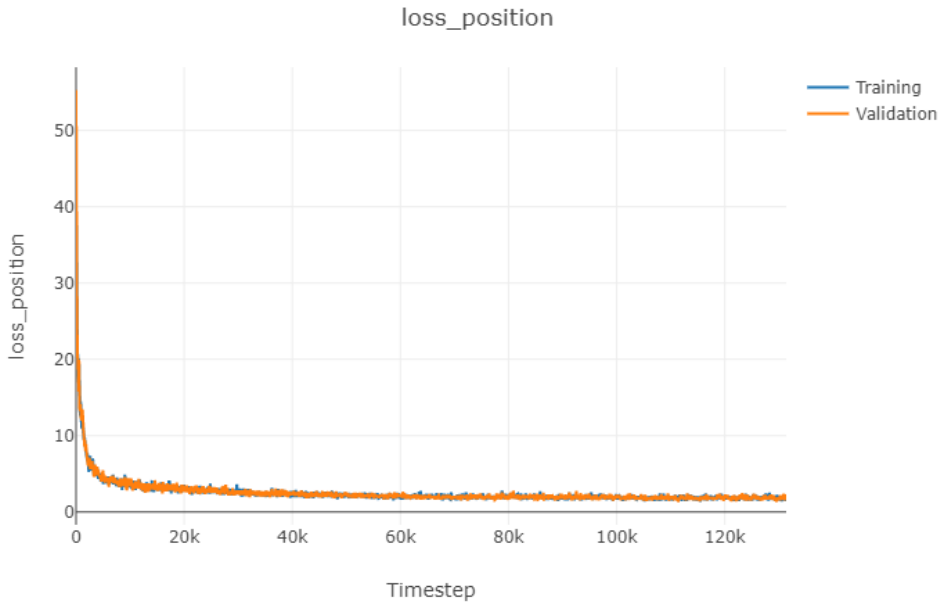
Figure 5.14: Classification loss on the *VKITTI* dataset for the standard *Faster R-CNN* network.

It can be seen in Figure 5.15 that the model rapidly trains to a configuration that dramatically reduces the magnitude of error. This behaviour is due to the pretraining of the backbone and RPN networks, as outlined in Section 5.10.1. Additional experiments that do not perform this pretraining phase do not reduce as rapidly. The loss plots given in Figure 5.15 are of the first training session, with a learning rate of 0.001. Figure 5.16 provides the losses over a second training session, in which the learning rate is reduced to 0.0001.

Though it can be seen in Figure 5.16 that the orientation and z component losses do decrease on the second training session, the magnitude of change is not comparable to that of the first training session, of Figure 5.15. This is to be expected, due to the combination of the network having been previously trained and the use of a small learning rate. However, experimentation with a second training session in which higher learning rates were used did not yield a loss reduction trend.



(a)



(b)

Figure 5.15: (a, b) Training and validation losses for the two pose components, over 150,000 training iterations, with a learning rate of 0.001.

(a)

(b)

Figure 5.16: (a, b) Training and validation losses for the two pose components, over 50,000 training iterations, with a learning rate of 0.0001.

5.10.4 *Detection, Classification and Pose Accuracy*

As specified earlier in this section, the *VKITTI* dataset was split on an 80 : 20 ratio of training to testing data. This section provides a quantitative evaluation of the pose prediction branch of the model outlined in this work, as well as the object detection and classification branches, on which the pose prediction branch is dependant. There are two components to the pose prediction branch of the network, the rotational component and the z translation component. As such, during training, there are two measures of performance pertaining to the systems pose accuracy. Table 5.1 provides MSE measures for the rotational and z translation components for both the training and testing portions of the *VKITTI* dataset, as well as the related MSE measures for the detection and classification branches.

Quantity	Mean Training MSE	Mean Testing MSE
Classification Error	0.060825	0.059888
Detection (Bounding Box) Error	0.039616	0.039755
Rotational Error	0.552863	0.512188
z Coordinate Error	1.798331	1.709215

Table 5.1: Mean Squared Error(s) over classification, detection, orientation and z coordinate estimates for the training and testng subsets of the *VKITTI* dataset.

It can be seen in Table 5.1 that the training and testing errors are similarly low, indicating a good bias/variance trade-off. In some cases, the testing data error is slightly lower than it's training counterpart, though by a small margin. As the split between training and testing data was randomised on an 80 : 20 ratio, it is possible that this is due to the testing subset containing a lower number of outliers. In addition to the MSE metrics of Table 5.1, density estimates over the MSE metrics are provided in Figures 5.17 and 5.18.

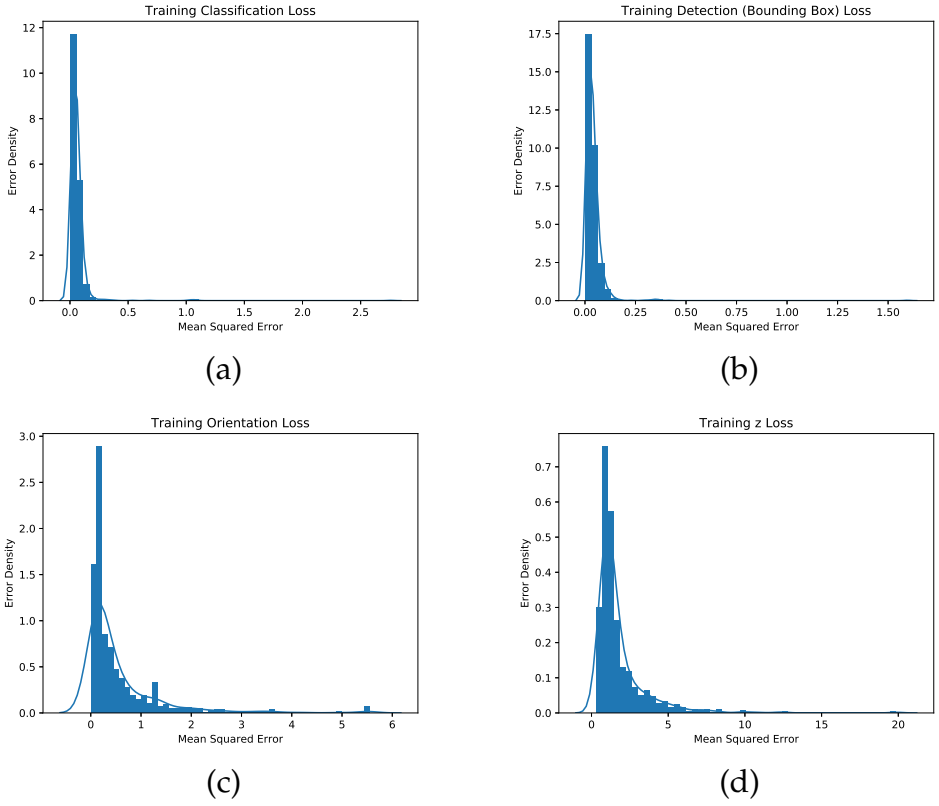


Figure 5.17: (a, b, c, d) Training MSE Kernel Density Estimates (KDE's).

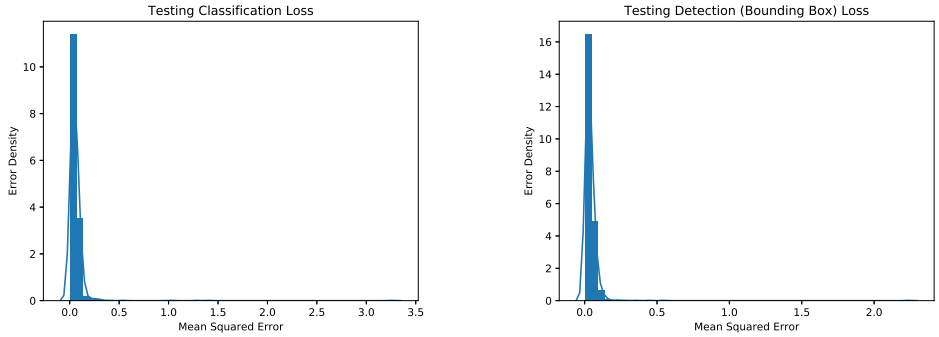


Figure 5.18: (a, b, c, d) Testing MSE Kernel Density Estimates (KDE's).

As can be seen in Figures 5.17 and 5.18, the MSE distributions have a low standard deviation, indicating that the MSE over the training and testing sets is consistent. This implies that the performance of the pose estimation network is consistent and that the network has indeed converged on a general solution.

5.10.5 *Weakly Supervised Training of Latent Shape*

Contrary to the training approach taken for the pose branch of the network, the training of the latent space shape point predictor is weakly-supervised in nature. As outlined in Section 5.2.1, the loss with respect to the latent space shape point is minimised in terms of the 2D rendering of the predicted shape under the predicted pose, against the ground truth semantic segmentation mask. Thus, the loss as outlined in this work may be considered a proxy loss for true 3D shape.

However, experimentation has proven the proposed shape evaluation pipeline to be difficult to train, largely due to the very high computational overhead involved in rendering an occupancy map for each detection in a given input frame (for a frame in which there are many instances of the object class of interest, this cost can become very high). As such, the proposed approach was unable to converge to a meaningful mapping from a given arbitrary detection, to a latent space shape point and more crucially, scale factor. In addition to the computational overhead involved, there is the inherent difficulty in scale estimation from a single view.

5.11 SUMMARY

The approaches introduced in this work to the problems of latent embedding of 3D shape and the regression of orientation and depth (and thus, pose for a known camera

configuration, as outlined in Section 5.5) provide useful tools for the understanding of 3D scenes and their components (object classes of interest), in the wild. The pose estimation approach outlined in this work has direct potential application in many scenarios in which it is necessary to estimate an object's pose (including translation, rather than just orientation [123]) from a single view.

Additionally, the 3D latent shape embedding approach outlined in this work, though not completely novel, is certainly of theoretical and practical interest. Despite the difficulties in end-to-end training of pose *and* shape in this work, the use of the GPLVM as a component within a 3D shape generating architecture is certainly a promising approach for domain specific applications, due to the ease and short duration of training, as shown in Figure 5.14. Purely CNN based approaches to the problem of learning 3D shape are typically far more computatioanlly expensive and complex (making them difficult to train).

6

DISCUSSION

This chapter reviews the proposed algorithms in this work within the context of the initially outlined research objectives of this thesis.

6.1 SUMMARY

This work has outlined novel approaches to various challenges in 3D computer vision, including handling dynamics in dense SLAM, reconstructing 3D objects and predicting both shape and pose. This chapter evaluates the methods, contributions and outcomes of these approaches, with the research objectives outlined in Section 1.4 as a point of reference.

This chapter proceeds as follows; first, Section 6.1.1 provides a summary of the dynamic SLAM and motion segmentation approach outlined in Chapter 3. Following this, Section 6.1.2 provides a similar evaluation of the approach to object reconstruction outlined in Chapter 4. Section 6.1.3 evaluates the approach taken to shape and pose prediction in Chapter 5.

Following the evaluations given in Sections 6.1.1, 6.1.2 and 5.11, consideration is given to the limitations of the approaches of this work and potential future directions in Section 6.2. Finally, a high level conclusion and closing remarks are given in Section 6.3.

6.1.1 *Real Time Motion Segmentation for Dense Volumetric Fusion*

One of the central research objectives outlined in Section 1.4 is the development of an algorithm which facilitates the dense reconstruction of dynamic environments. As highlighted in Sections 1.1 and 2.3, this has remained a challenging problem in 3D computer vision, due to the reliance of prominent pose estimation techniques on reliant point correspondences between frames. The novel approach taken in Chapter 3 mitigates the restrictions incurred by this dependence by actively excluding dynamic scene components from the pose estimation phase of the pipeline, as shown in Figure 3.6.

As demonstrated in Sections 3.4 and 3.5, the approach taken shows improvements in pose estimation quality over an open implementation [30] of the standard KinectFusion [1] pipeline when evaluated on the *Dynamic Objects* subset of the TUM RGBD dataset [106]. This improved performance is evident in Tables 3.1 and 3.2, and Figures 3.10 and 3.11. The improvement in the ability to accurately track sensor pose during reconstruction fulfils the research objective of designing an algorithm that provides an improvement in pose estimation versus static dense SLAM in dynamic scenes.

Additionally, it is evident from Section 3.4 that the proposed approach yields high quality reconstructions in dynamic environments that are comparable to their static counterparts when there is no motion in the scene. Again, this result is directly

satisfying of the research objective to design a system that is capable of providing comparable quality reconstructions in previously troublesome environments.

Section 3.7 demonstrates how the dynamic segmentation ability of the proposed approach may be leveraged for object recognition purposes. It is shown that the use of the dynamics information may be used to indicate to the system an object of interest, such that 3D features may be extracted and used for training of, and prediction with simple classifiers in an interactive manner. In Section 1.4 potential use of dynamics for object recognition purposes is outlined as an additional research objective.

The demonstrated improvements in pose estimation and the ability to perform simple scene understanding in dynamic scenes are facilitated by the following central contributions of Chapter 3. First is the introduction of the novel dual representation of the scene, in which a stable version of the scene is maintained as both the resultant reconstruction and the source of depth map to which live frames are registered. The dynamic scene representation has all observed data points integrated. This dual representation allows for the separation and thus segmentation of dynamic, moving scene components from their stable counterparts, thus allowing the pose estimation phase to use a reliable, non corrupted scene model.

In addition to the dual scene representation, the system outlined in Chapter 3 introduces a novel online adjustment schema for the TSDF truncation region. This online adjustment of the truncation region allows for live integration and removal of surface data in the dynamic model by facilitating real time space carving such that changes in the scene are reflected instantly in the dynamic model. This technique combined with the dual representation is the basis of the proposed approach.

The system and results outlined in Chapter 3 have the potential to impact significantly on real world applications of 3D vision systems. Due to the approach utilising volumetric representations versus less scalable alternatives, there is potential for application in large scale robotics. Such applications are ordinarily inhibited by the static

nature of dense SLAM approaches. Additionally, the ability to operate in dynamic environments when coupled with the ability to exploit dynamics has the potential to greatly impact on the fields of semantic dense SLAM and 3D scene understanding.

6.1.2 *Probabilistic Object Reconstruction with Online Drift Correction*

The second major research objective outlined in Section 1.4 is the development of a system that allows for the reconstruction of arbitrary objects in a globally consistent manner. As highlighted in Sections 1.2 and 2.4, object centric dense reconstruction is a markedly difficult problem compared to it's scene scale counterpart. A prominent technical challenge in object reconstruction is the enforcement of object consistency when there are erroneous pose estimation results. The system outlined in Chapter 4 mitigates object inconsistencies by the use of a novel object reconstruction pipeline that includes an online correction procedure.

The system outlined in Chapter 4 when compared to the state-of-the-art object reconstruction method of *Ren et al* [74], demonstrates a vast improvement in efficacy for general object reconstruction. The latter system was unable to reconstruct the test sequences used in Sections 4.7 and 4.8.

Furthermore, it is shown in Table 4.2 and Figure 4.19 of Section 4.8 that the approach presented in this work is capable of yielding high quality reconstructions, relative to manually extracted reconstructions obtained with a standard, scene scale dense SLAM approach. As outlined in Section 1.4, a principal requirement of the object reconstruction research objective is to develop an approach that provides high quality reconstructions, utilising only observations belonging to the object of interest.

Whereas many dedicated object reconstruction systems rely on either known poses or poses for which there is a strong prior (such as an object on a turntable in front of a

laser scanner), the approach proposed in this work is able to perform pose estimation online for arbitrary trajectories. In Section 1.4 one of the requirements of the central research objective of the work of Chapter 4 is that the system is capable of operating with no a-priori known trajectory. This competency may be observed in Table 4.1 and Figure 4.13 of Section 4.8. This competency allows the system to be used with commodity equipment, thus facilitating ease of use.

The ability of the proposed object reconstruction system to produce globally consistent reconstructions is demonstrated in Section 4.7 and is one of the key research objectives of the work of Chapter 4. The potential impact of a system that can be used to easily obtain such reconstructions is particularly evident for fields that are heavily data dependent. Such a field is machine learning, where as outlined in Section 1.2, there is an abundance of real world 2D image data available for the learning of vision tasks, but not a comparable amount for 3D geometry.

6.1.3 *Shape and Pose Prediction*

The work presented in Chapter 5 proposes an ambitious approach to simultaneous pose and 3D shape prediction for a domain specific problem, namely class specific inference for real world scenes. Firstly, the work presents an approach to the pose estimation of instantaneous object detections, without requiring an explicit tracking procedure; pose is inferred as a one-shot regression. The evaluation of this pose regression performance demonstrates consistently low error over a randomised split of the dataset used, prividing pose estimates for the difficult case of monocular vision.

Secondly, the proposed approach makes use of generative probabilistic modelling to learn latent embeddings of complex 3D geometry for the class of interest, namely cars. This latent embedding demonstrates the ability to generate a range of domain

specific geometries, given a low dimensional latent space input (2D). It is demonstrated that the model is trivial to train, requiring little data and computation time when compared to CNN based approaches.

Though the driving motivation of Chapter 5 is the unification of these two problems (pose estimation and 3D shape estimation), experimentally this proved to be computationally difficult. As the architecture presented is based on the *RCNN* approach, the model is trained on each proposed object in the input frame. When regressing 3D shape and attempting to optimise the latent space point that generated the 3D shape, there is considerable computational overhead involved in performing the Gaussian Process Regression step. Though this is not an issue in the case of a small number of confirmed detections, when performing this inference step for each proposal of an *RCNN* like architecture, it may need to be computed many times (experimentally, for some frames this incurred upwards of 170 GP regressions).

As this requires a lot of memory, it is not feasible to run this inference step on a single GPU simultaneously with the other neural network components. As the GPU hardware available during the development of this work was a single NVIDIA GTX1060 6GB card, this phase was run on an Intel Core i5 Quad Core CPU. For some frames the running time was upwards of 1 minute for a single frame. However, there is no theoretical reason that the rich features provided by CNN models such as that used in Chapter 5 can not be used to learn a mapping from detection to 2D latent point. The problems encountered in the development of this work are computational.

In summary, the approach outlined in Chapter 5 contains multiple components that are of direct use and theoretical interest. However, computational difficulties were encountered when amalgamating the pose and shape prediction components within the proposed *RCNN* framework.

6.2 FUTURE WORK

The system of Chapter 3 provides a strong foundation for further research into both dense SLAM for dynamic environments and semantic SLAM. However, there are potential directions of subsequent research that follow on directly from the approach of this work. Firstly, at present the system presented in this work currently maintains two full TSDF volumes; there is a volume for the static scene representation and a volume for the dynamic scene representation. An improvement that could be introduced in a later iteration of this work is the use of a sparse representation of the dynamic scene volume, which would improve the space complexity of the system.

A second potential improvement to the work of Chapter 3 is the expansion of the motion segmentation itself to be semantically aware. At present, the system is capable of determining which elements of the world are undergoing motion and is able to leverage this information to improve pose estimation and thus the quality of the resultant reconstruction. With further semantic capabilities at the lower level, a revised system could have the ability to separate observed motion into instances of dynamic objects.

The object reconstruction system presented in Chapter 4 provides a simple means to obtain geometrically consistent reconstructions of 3D objects. Though the proposed system demonstrates an improvement over comparative methods, it still requires the end user to indicate which object is to be reconstructed. An alternative approach would be to leverage some of the advancements in salient object detection [124] to allow for automation of the process.

The probabilistic framework on which the system is based could additionally be extended in a later iteration of the work to estimate the geometry of any missing sections of the surface of the object being reconstructed. Such an extension would

further increase the consistency of the reconstructions in the case where full coverage with the sensor has not been achieved.

Another potential line of research following on from the contributions outlined is the extension to the multiple object case. The ability to track and reconstruct multiple objects in the sensor view combined with the potential salient object detection would greatly enhance the efficiency of 3D data collection.

Finally, future research may be carried out on the method of simultaneous shape and pose prediction outlined in Chapter 5. Predominantly, this work would entail investigation into making the simultaneous pose and shape inference tractable. Potential directions include changing the core architecture to another that does not require the entire prediction pipeline to be run for each detection. Additionally, the computational cost of the GP regression may be reduced by the use of sparse GP's. However, this would entail a new direction of research into sparse GPLVM modelling of 3D shape.

6.3 CLOSING REMARKS

This work has introduced novel approaches to a range of non-trivial research problems in 3D computer vision. The contributions of this work span the extension of the traditional dense SLAM pipeline to the dynamic case, the consistent reconstruction of arbitrary objects and the prediction of real world objects, for which only a single view pair is available. The remit of this work has direct relevance to the increasingly automated world of machine perception and intelligence.

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 3.2.1 are given as follows.

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

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

$$\frac{\partial \mathbf{R}}{\partial \gamma} = \begin{bmatrix} -\frac{2\gamma\left(\alpha^2-\beta^2-\gamma^2+1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2}-\frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} & -\frac{2\gamma(2\alpha\beta+\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2}+\frac{1}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1}-\frac{2\gamma(2\alpha\gamma-\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} \\ -\frac{2\gamma(2\alpha\beta-\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2}-\frac{1}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\gamma\left(\alpha^2-\beta^2+\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2}-\frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} & \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1}-\frac{2\gamma(\alpha+2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} \\ \frac{2\alpha}{\alpha^2+\beta^2+\gamma^2+1}-\frac{2\gamma(2\alpha\gamma+\beta)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} & \frac{2\beta}{\alpha^2+\beta^2+\gamma^2+1}+\frac{2\gamma(\alpha-2\beta\gamma)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2} & \frac{2\gamma\left(\alpha^2+\beta^2-\gamma^2-1\right)}{\left(\alpha^2+\beta^2+\gamma^2+1\right)^2}+\frac{2\gamma}{\alpha^2+\beta^2+\gamma^2+1} \end{bmatrix} \tag{3}$$

.2 MOTION SEGMENTATION RESULTS APPENDICES

.2.1 *Motion Segmentation figure: ATE results*

In this section, figure: results for the Motion Segmentation system to complement those outlined in Section 3.5 are given. The results in this section assess Absolute Trajectory Error on the TUM Dynamic Objects *Validation* set. Quantitative results are given in Table 1 and visualised in Figure 1.

TUM Standard Sequence Name	*MoSeg* ATE (m)	*Baseline* ATE (m)
fr3-sitting-static	0.044	**0.030**
fr3-sitting-xyz	**0.044**	0.048
fr3-sitting-halfsphere	**0.026**	0.028
fr3-sitting-rpy	**0.043**	0.044
fr3-walking-static	**0.121**	0.466
fr3-walking-xyz	**0.082**	0.633
fr3-walking-halfsphere	**0.401**	0.525
fr3-walking-rpy	**0.073**	0.561

Table 1: The Absolute Trajectory Error (ATE) results (in metres, lower is better) achieved by the proposed approach in comparison to the baseline InfiniTAM [30] framework on a variety of the standard sequences from the TUM RGBD *Validation* dataset [106]. Results are in the format mean $\pm$ standard deviation. The better result (by mean) on each sequence is highlighted in bold.

.2.2 *Motion Segmentation figure: RTE results*

In this section, figure: results for the Motion Segmentation system to complement those outlined in Section 3.5 are given. The results in this section assess Relative Trajectory

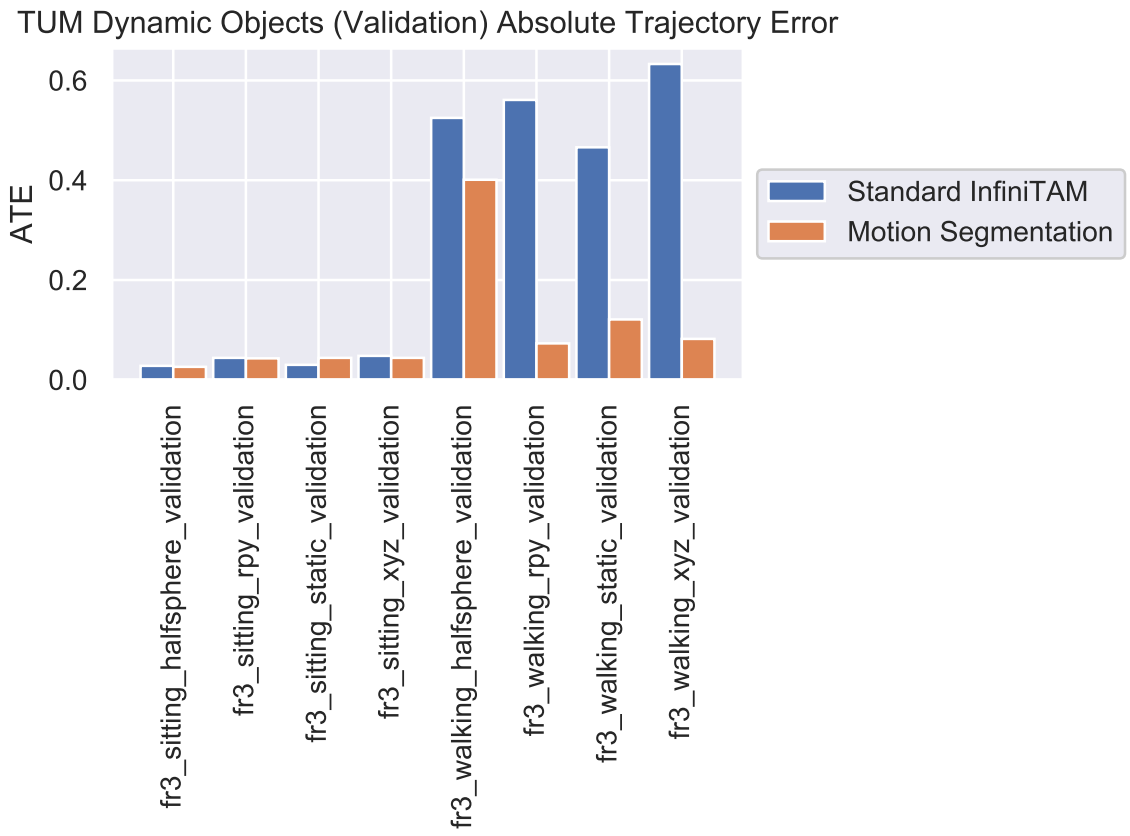


Figure 1: Absolute Trajectory Error for the TUM Dynamic Scenes *Validation* dataset.

Error on the TUM Dynamic Objects *Validation* set. Quantitative results are given in Table 2 and visualised in Figure 2.

TUM Standard Sequence Name	*MoSeg* RTE (m)	*Baseline* RTE (m)
fr3-sitting-static	**0.011**	**0.011**
fr3-sitting-xyz	**0.031**	0.034
fr3-sitting-halfsphere	0.024	**0.022**
fr3-sitting-rpy	0.051	**0.048**
fr3-walking-static	**0.083**	0.163
fr3-walking-xyz	**0.067**	0.285
fr3-walking-halfsphere	**0.167**	0.211
fr3-walking-rpy	**0.121**	0.194

Table 2: The Relative Trajectory Error (RTE) results (in metres, lower is better) achieved by the proposed approach in comparison to the baseline InfiniTAM [30] framework on a variety of the standard sequences from the TUM RGBD *Validation* dataset [106]. Results are in the format mean $\pm$ standard deviation. The better result (by mean) on each sequence is highlighted in bold.

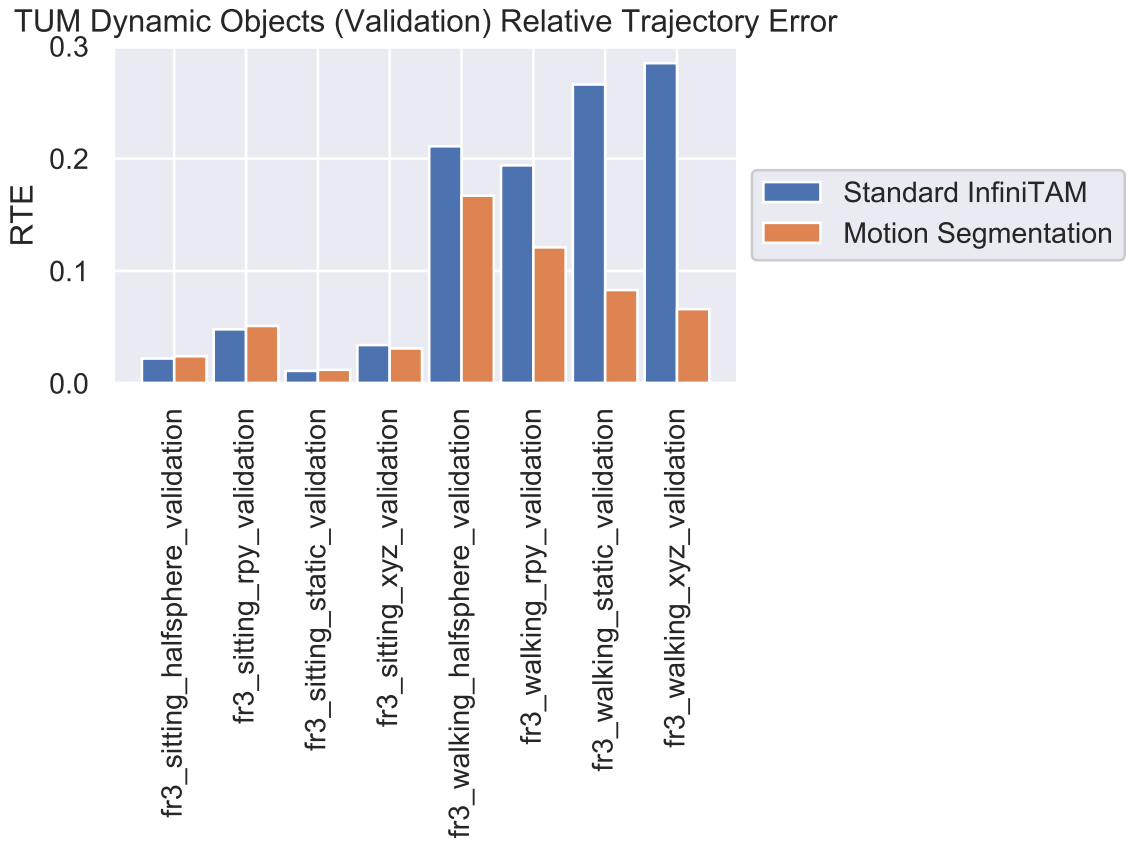


Figure 2: Relative Trajectory Error for the TUM Dynamic Scenes *Validation* dataset.

BIBLIOGRAPHY

[1] 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, Oct. 2011.

[2] O. Kähler, V. A. Prisacariu, and D. W. Murray, *Real-Time Large-Scale Dense 3D Reconstruction with Loop Closure*, pp. 500–516. Cham: Springer International Publishing, 2016.

[3] 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," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.

[4] 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, (Washington, DC, USA), pp. 580–587, IEEE Computer Society, 2014.

[5] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 640–651, Apr. 2017.

[6] 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.

[7] 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," *ACM Trans. Graph.*, vol. 34, pp. 154:1–154:17, Nov. 2015.

[8] 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," *CoRR*, vol. abs/1510.03727, 2015.

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

[10] 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, June 2015.

[11] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, Feb. 1992.

[12] 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, (New York, NY, USA), pp. 303–312, ACM, 1996.

[13] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, (New York, NY, USA), pp. 163–169, ACM, 1987.

[14] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, (Nara, Japan), November 2007.

[15] 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, 2008.

[16] T. J. Purcell, I. Buck, W. R. Mark, and P. Hanrahan, "Ray tracing on programmable graphics hardware," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, (New York, NY, USA), pp. 703–712, ACM, 2002.

[17] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, (New York, NY, USA), pp. 143–150, ACM, 1986.

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

[19] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, (Washington, DC, USA), pp. 2320–2327, IEEE Computer Society, 2011.

[20] M. Niessner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, pp. 169:1–169:11, Nov. 2013.

[21] 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, Dec. 2013.

[22] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart, "Spectral surface quadrangulation," in *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, (New York, NY, USA), pp. 1057–1066, ACM, 2006.

[23] 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.

[24] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," *J. Vis. Comun. Image Represent.*, vol. 25, pp. 137–147, Jan. 2014.

[25] H. Pfister, M. Zwicker, J. Baar, and M. Gross, "Surfels: Surface elements as rendering primitives," May 2000.

[26] S. Laine and T. Karras, "Efficient sparse voxel octrees," in *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, (New York, NY, USA), pp. 55–63, ACM, 2010.

[27] R. Mukundan, "Quaternions: From classical mechanics to computer graphics, and beyond," in *Proceedings of the 7 th Asian Technology Conference in Mathematics, 2002*, 2002.

[28] 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, Sept. 2014.

[29] B. Glocker, J. Shotton, A. Criminisi, and S. Izadi, "Real-time rgb-d camera relocalization via randomized ferns for keyframe encoding," *TVCG*, September 2014.

[30] 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," *CoRR*, vol. abs/1410.0925, 2014.

[31] 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," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, pp. 1241–1250, Nov. 2015.

[32] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," in *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, (New York, NY, USA), pp. 1134–1141, ACM, 2005.

[33] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, pp. 31–43, winter 2010.

[34] 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, June 2015.

[35] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, (New York, NY, USA), pp. 417–424, ACM Press/Addison-Wesley Publishing Co., 2000.

[36] R. Smith, M. Self, and P. Cheeseman, *Estimating Uncertain Spatial Relationships in Robotics*, pp. 167–193. New York, NY: Springer New York, 1990.

[37] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), (Berlin, Heidelberg), pp. 404–417, Springer Berlin Heidelberg, 2006.

[38] 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.

[39] T. K. Ho, "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, (Washington, DC, USA), pp. 278–, IEEE Computer Society, 1995.

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

[41] 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., 2003.

[42] 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., 2011.

[43] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Synthcam3d: Semantic understanding with synthetic indoor scenes," *CoRR*, vol. abs/1505.00171, 2015.

[44] C.-Y. Liou, J.-C. Huang, and W.-C. Yang, "Modeling word perception using the elman network," *Neurocomputing*, vol. 71, no. 16, pp. 3150 – 3157, 2008. Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006).

[45] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, June 2015.

[46] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4628–4635, May 2017.

[47] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[48] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," *CoRR*, vol. abs/1808.08378, 2018.

[49] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[50] L. V. Tsap, D. B. Goldof, and S. Sarkar, "Nonrigid motion analysis based on dynamic refinement of finite element models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 526–543, May 2000.

[51] J. Chen, X. Wu, M. Y. Wang, and F. Deng, *Human Body Shape and Motion Tracking by Hierarchical Weighted ICP*, pp. 408–417. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[52] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," vol. 604-613, 10 2000.

[53] 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, June 2012.

[54] P. Lammich and S. R. Sefidgar, "Flow networks and the min-cut-max-flow theorem," *Archive of Formal Proofs*, June 2017. http://isa-afp.org/entries/Flow_Networks.html, Formal proof development.

[55] C. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, Springer, 2006.

[56] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[57] S. Vicente and L. Agapito, "Soft inextensibility constraints for template-free non-rigid reconstruction," in *Computer Vision – ECCV 2012* (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), (Berlin, Heidelberg), pp. 426–440, Springer Berlin Heidelberg, 2012.

[58] 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, June 2012.

[59] A. L. M. Levada, N. D. A. Mascarenhas, and A. Tannús, "Improving potts mrf model parameter estimation in image analysis," *2008 11th IEEE International Conference on Computational Science and Engineering*, pp. 211–218, 2008.

[60] S. Boyd, S. Boyd, L. Vandenberghe, and C. U. Press, *Convex Optimization*. Berichte über verteilte messysteme, Cambridge University Press, 2004.

[61] 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, May 2013.

[62] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, pp. 25–36. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[63] 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)*, 2013.

[64] 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, (Washington, DC, USA), pp. 1–8, IEEE Computer Society, 2013.

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

[66] S. Perera, N. Barnes, X. He, S. Izadi, P. Kohli, and B. Glocker, "Motion segmentation of truncated signed distance function based volumetric surfaces," IEEE, Jan. 2015.

[67] L. Kavan, S. Collins, and J. Zara, "Dual quaternions for rigid transformation blending," tech. rep., 2006.

[68] 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, 2006.

[69] 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, Sept. 2009.

[70] X. Lladó, A. D. Bue, A. Oliver, J. Salvi, and L. Agapito, "Reconstruction of non-rigid 3d shapes from stereo-motion," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 1020 – 1028, 2011.

[71] V. A. Prisacariu and I. D. Reid, "Pwp3d: Real-time segmentation and tracking of 3d objects.," *International Journal of Computer Vision*, vol. 98, no. 3, pp. 335–354, 2012.

[72] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *Proceedings of European Conference on Computer Vision*, 2008.

[73] R. Garg, A. Roussos, and L. Agapito, "Dense variational reconstruction of non-rigid surfaces from monocular video," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1272–1279, June 2013.

[74] 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, Dec. 2013.

[75] A. Agudo, J. M. M. Montiel, L. Agapito, and B. Calvo, "Online dense non-rigid 3d shape and camera motion recovery," in *BMVC*, 2014.

[76] 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, June 2015.

[77] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, (London, UK, UK), pp. 298–372, Springer-Verlag, 2000.

[78] R. Yu, C. Russell, N. D. F. Campbell, and L. Agapito, "Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 918–926, 2015.

[79] T. Gupta, D. Shin, N. Sivagnanadasan, and D. Hoiem, "3dfs: Deformable dense depth fusion and segmentation for object reconstruction from a handheld camera," *CoRR*, vol. abs/1606.05002, 2016.

[80] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2 ed., 2001.

[81] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel, "Sequential non-rigid structure from motion using physical priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 979–994, May 2016.

[82] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *J. Mach. Learn. Res.*, vol. 6, pp. 1783–1816, Dec. 2005.

[83] 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, 2013.

[84] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," 1992.

[85] 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.

[86] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[87] 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.

[88] N. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," vol. 46, pp. 175–185, 08 1992.

[89] 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], 2015.

[90] 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.

[91] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[92] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[93] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, pp. 1345–1359, Oct. 2010.

[94] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *ArXiv e-prints*, Nov. 2017.

[95] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[96] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[97] 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.

[98] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[99]

[100] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[101] A. Pumarola, A. Agudo, L. Porzi, A. Sanfeliu, V. Lepetit, and F. Moreno-Noguer, "Geometry-aware network for non-rigid shape prediction from a single view," *CoRR*, vol. abs/1809.10305, 2018.

[102] 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.

[103] M. D. Shuster, "Survey of attitude representations," *Journal of the Astronautical Sciences*, vol. 41, pp. 439–517, Oct. 1993.

[104] H. Anton, *Elementary Linear Algebra*. Wiley, 1991.

[105] S. D. Roth, "Ray casting for modeling solids," *Computer Graphics and Image Processing*, vol. 18, no. 2, pp. 109 – 144, 1982.

[106] 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.

[107] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A, Optics and Image Science*, vol. 4, pp. 629–642, Apr 1987.

[108] Y. Singer and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *In ICML*, pp. 807–814, 2007.

[109] 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.

[110] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4471–4478, May 2017.

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

[112] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," in *Proceedings of the Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EMMCVPR '01, (Berlin, Heidelberg), pp. 359–374, Springer-Verlag, 2001.

[113] G. Rote, "Computing the minimum hausdorff distance between two point sets on a line under translation," *Inf. Process. Lett.*, vol. 38, pp. 123–127, May 1991.

[114] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 91–99, Curran Associates, Inc., 2015.

[115] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtualworlds as proxy for multi-object tracking analysis," pp. 4340–4349, 06 2016.

[116] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[117] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[118] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[119] C. E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006.

[120] C. Ren, V. Prisacariu, and I. Reid, "Regressing local to global shape properties for online segmentation and tracking," *International Journal of Computer Vision*, vol. 106, no. 3, pp. 269–281, 2014.

[121] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.

[122] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[123] M. Braun, Qing Rao, Y. Wang, and F. Flohr, "Pose-rcnn: Joint object detection and pose estimation using 3d object proposals," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1546–1551, Nov 2016.

[124] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey.," *arXiv preprint arXiv:1411.5878*, vol. 2, no. 4, 2014.