Thesis Title

Department of Engineering Science

University of Oxford

Mr Jack Miles Hunt

Declaration

Acknowledgements

Abstract

Notation and Abbreviations Used

Contents

Chapter 1

Introduction

Chapter 2

Literature Review

2.1 Tracking and Mapping

Besl and McKay [1]

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

Curless and Levoy [2]

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

Zhou et.al. [3]

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

Censi [4]

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

Newcombe et.al. [5]

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

Neissner et.al. [6]

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

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

Thomas et.al. [7]

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

Salas-Moreno et.al. [8]

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

Stuckler et.al. [9]

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

- Pose estimation by maximising observation likelihood with uncertainty measure.

Salas-Moreno et.al. [11]

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

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

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

Whelan et.al [14]

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

Zhou et.al. [15]

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

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

Kahler et.al [16]

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

2.2 Semantic SLAM

Civera et.al. [17]

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

Stuckler et.al [18]

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

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

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

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

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

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

Cavallari et.al. [27]

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

2.3 Dynamic SLAM, Motion Segmentation and Optical Flow

Tsap et.al. [29]

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

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

Chen et.al [30]

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

Sun et.al. [31]

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

Unger et.al. [32]

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

- Again, motion only. No reconstruction.

Herbst et.al. [33]

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

Stuckler et.al. [35]

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

Keller et.al. [36]

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

Perera et.al. [37]

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

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

Newcombe et.al. [38]

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

2.4 Object Reconstruction

Curless and Levoy [2]

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

Kolev et.al. [40]

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

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

Weise et.al. [41]

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

Ren et.al. [42]

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

Dou et.al. [44]

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

Gupta et.al. [45]

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

2.5 Shape and Pose Prediction

Prisacariu et.al. [46]

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

Dame et.al. [48]

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

Toshev et.al. [49]

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

- No reconstruction, pose estimation only.

Wohlhart et.al. [50]

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

Chang et.al. [51]

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

Rock et.al. [52]

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

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

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

Gwak et.al [54]

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

Chapter 3

Real Time Motion Segmentation for Dense Volumetric Fusion

3.1 Introduction

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

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

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

Once a part of the dynamic model has been stable for a sufficient period, it's volumetric data is integrates in to the static model and it is used for the tracking phase of the pipeline. The use of Volumetric structures in this work is motivated by previous works on Voxel Block Hashing [6], providing efficient, real time lookup operations. The presented approach exploits the abstraction that Voxel Blocks provide; a block of voxels is interpreted as a region of space that can be either static or dynamic. From a survey of the literature, it appears that this approach is the first to utilise such a dual representation for the motion segmentation problem.

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

3.2 Related Work

3.3 Static Volumetric Fusion

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

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

- Camera Tracking.
- Model Integration.
- Rendering.

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

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

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

where v denotes a TSDF Voxel.

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

3.3.1 Camera Tracking

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

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

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

and has the following form

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

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

Attitude Representation

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

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

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

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

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

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

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

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

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

Pose Recovery Formulation

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

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

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

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

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

For the Gradient update phase of the ICP algorithm, the Partial Derivatives $\frac{\partial E}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}}$ may be derived as follows in Equation 3.11. As a first step, the following definition is made

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

With this substitution in place, the derivation proceeds as follows

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

Note that the derivation for the Partial Derivative $\frac{\partial E}{\partial \mathbf{t}}$ is analogous and has been omitted. Note that for the Translation Gradient, the term $\frac{\partial \mathbf{R}}{\partial\{\alpha,\beta,\gamma\}}$ is replaced with $\frac{\partial \mathbf{t}}{\partial\{t_x,t_y,t_z\}}$.

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 of Appendix .1.

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

$$\mathbf{J}_R = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{3.12}$$

likewise for the Translation Partial Derivatives $\frac{\partial \mathbf{t}}{\partial\{t_x,t_y,t_z\}}$

$$\mathbf{J}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

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

$$\begin{aligned}\mathbf{J} &= \left.\frac{\partial E}{\partial \mathbf{R}_{\{\alpha,\beta,\gamma\}}, \mathbf{t}_{\{x,y,z\}}}\right|_{\alpha,\beta,\gamma=0} \\ &= \begin{bmatrix} 0 & -z & y & 1 & 0 & 0 \\ z & 0 & -x & 0 & 1 & 0 \\ -y & x & 0 & 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{3.14}$$

Pose Recovery Optimisation

3.4 Volumetric Fusion with Dynamic Scenes

3.5 Qualitative Results

3.6 Quantitative Results

3.7 Application to Semantic Scene Understanding

Chapter 4

Probabilistic Object Reconstruction with Online Drift Correction

4.1 Introduction

4.2 Related Work

4.3 Probabilistic Forumlation of Object Reconstruction

4.4 Online Model Correction

4.5 Volumetric Segmentation and Explicit Loop Closure Detection

4.6 Qualitative Results

4.7 Quantitative Results

Appendices

.1 Mathematical Appendices

.1.1 Rodriguez Paramaterisation Partial Derivatives

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

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

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

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

.2 Misc Appendix

Bibliography

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

[2] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’96, pp. 303–312, ACM.

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

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

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

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

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

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

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

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

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

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

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

[14] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense rgb-d slam with volumetric fusion," vol. 34, no. 4-5, pp. 598–626.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[36] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3d reconstruction in dynamic scenes using point-based fusion," in *Proceedings of the 2013 International Conference on 3D Vision*, 3DV '13, pp. 1–8, IEEE Computer Society.

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

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

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

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

[41] T. Weise, T. Wismer, B. Leibe, and L. V. Gool, "In-hand scanning with online loop closure," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 1630–1637.

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

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

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

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

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

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

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

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

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

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

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

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

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