

A Regularized Fusion based 3D Reconstruction Framework: Analyses, Methods and Applications

vorgelegt von

M.Sc.

Muhammad Asif Ali Rajput

geb. in Sukkur, Pakistan.

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Henning Sprekeler

Gutachter: Prof. Dr.-Ing. Olaf Hellwich

Gutachter: Dr.-Ing. Anko Börner

Gutachter: Prof. Dr.-Ing Reinhard Koch

Tag der wissenschaftlichen Aussprache: 24. September 2018

Berlin 2018

Abstract

Recent developments in depth sensing technologies enabled mobile robots to perceive surroundings with high accuracy. Robotic applications, equipped with depth perception technology, enable the capability of autonomous navigation to self-driving cars, assist in critical surgical procedures, or reconstruct the 3D model of a potentially hazardous environment. There exists a variety of 3D sensors ranging from highly accurate laser-based range sensors to low-range active depth cameras; the selection of a 3D sensor, however, directly affects the capabilities of robotic applications to perceive surroundings. Unlike self-driving autonomous vehicles, which are equipped with high-cost LiDAR 3D sensors to ensure safety, mobile robots are usually equipped with low-cost active or passive depth sensors. This means that acquired depth information from low-cost 3D sensors is prone to accumulating estimation noise. In principal, existing 3D reconstruction frameworks employ multiple instances of erroneous depth samples in an incremental fashion to produce high-quality 3D models.

In this thesis, the research objective is focused on reducing the effects of error-prone depth information by employing a proposed 3D reconstruction framework capable of reducing accumulated noise, using a regularizing 3D integration system. The underlying principal of existing state-of-the-art volumetric reconstruction techniques is unchanged since the introduction in 1996. The novelty of the proposed framework lies in the use of a *prior* smoothing constraint that represents, on a small-scale, that the surface of the perceived object is smooth. The application of this smoothing constraint on depth information, acquired from low-cost 3D sensors, can enhance the quality of 3D information without sacrificing fine details in surface geometry.

Critical experimentation and empirical evaluation of the new reconstruction framework have shown a significant increase in accuracy and quality of reconstructed shapes compared to state-of-the-art methods. Furthermore, by quantitative assessments it has been observed that

employing smoothing constraints to an incremental 3D fusion process accelerates the surface estimation process. Therefore, comparatively fewer depth samples are required to generate high-quality 3D surfaces. These properties of the proposed research link well with robotic applications which rely on somehow inaccurate (say, because low-cost) image sensors.

Zusammenfassung

Der heutige Stand der Technik in der mobilen Robotik ermöglicht es, die Umgebung mit hoher 3D Genauigkeit abzutasten. Dies ist von Vorteil für viele Anwendungen in den Bereichen Autonomes fahren, Medizinische Operationen oder Inspektion von schwer zugänglichen Gebieten. Die Tiefensensoren lassen sich zwischen hoch-akkurate 3D Scanner und kostengünstige Tiefenkameras klassifizieren. Grundlegend ist das Ziel kostengünstige Tiefensensoren zu nutzen und inkrementell die Genauigkeit der über die Zeit aufgebauten 3D Modelle zu verbessern.

Diese Arbeit untersucht die Reduzierung der Fehlereinflüsse durch fehlerhafte und ungenaue Tiefenmessungen. Der vorgestellte technische Ansatz ist in der Lage das Rauschen mithilfe von Regularisierung im 3D Raum stark zu verringern. Die Neuerung des Ansatzes liegt in der Integration des Vorwissens (engl. Prior) über die differentielle Glattheit von beobachteten Oberflächen. Die Arbeit demonstriert, dass mithilfe des Ansatzes die 3D Modellierungsgenauigkeit stark verbessert werden kann, ohne den Detailgrad der beobachteten Geometrien zu verlieren.

Experimente und empirische Auswertungen haben gezeigt, dass mithilfe der vorgestellten Methode die erreichten Genauigkeiten sich stark von den bekannten Ansätzen hervorheben. Zusätzlich, führt die Anwendung des Ansatzes zur effizienteren Rekonstruktion der Geometrien. Im Vergleich zu existierenden Arbeiten, erfordert die Methode weniger Datenpunkte (geringere Bildauflösungen), um dennoch vergleichbare Genauigkeit zu erreichen. Der Mehrwert der Arbeit erstreckt sich auf alle robotischen Anwendungen, wo die Wahrnehmung und Rekonstruktion der Umweltgeometrien mit kostengünstigen Tiefensensoren erreicht werden soll.

Acknowledgments

First and foremost, I would like to sincerely thank my thesis advisors, Prof. Dr. Olaf Hellwich and Dr. Anko Börner for their guidance, support and recommendations throughout this study and appreciate their confidence in me.

I would also like to thank Dr. Eugen Funk for helping me as a friend and mentor, his continuous guidance is one of the reason behind this work.

Finally, I would like to pay my gratitude to Higher Education Commission (HEC) Pakistan for providing me with the financial support and opportunity to follow my dreams of higher education in a world-renowned university.

Dedications

I would like to dedicate this work to my wife *Maria*, who provided tireless support and motivation so that I can obtain the highest academic qualification without losing the hope. This work is also dedicated to my parents, who raised me to question everything which allowed me to think critically in this research phase.

Contents

Abstract	iii
Abstract-DE	iv
Acknowledgments	vi
Dedications	viii
1 Introduction	1
1.1 Depth Imaging for 3D Reconstruction	2
1.2 Challenges and the Research Question	4
1.3 Contributions	6
1.4 Thesis outline	7
1.5 Summary	8
2 Related Work	9
2.1 General Shape Representation	10
2.1.1 Simplex Representation	10
2.1.2 Parametric Representation	11
2.1.3 Implicit Representation	12
2.1.4 Surface Splatting	13
2.2 Prior Based Shape Approximation	14
2.2.1 Regular Priors	14
2.2.2 Local Smoothing Priors	16
2.2.3 Global Smoothing Priors	18

2.3	Depth Map Smoothing	22
2.4	Incremental 3D Fusion	24
2.5	Depth Outliers removal	30
2.6	Key Considerations	32
2.7	Summary	34
3	Methodology	37
3.1	Framework Structure	37
3.2	Validation and Evaluation	39
3.2.1	Evaluation Framework	39
3.2.2	Performance Metrics	41
3.3	Datasets	43
3.3.1	Synthetic Piecewise Function	43
3.3.2	Synthetic 3D Complex Environment	44
3.3.3	Realistic 3D Complex Environment	45
3.4	3D Sensors	48
3.5	Summary	49
4	Fundamentals of Volumetric 3D Integration	51
4.1	Signed Distance Function	51
4.1.1	Definition	52
4.2	SDF from depth images	55
4.3	Effects of incremental 3D fusion	56
4.3.1	Relation of convergence with weights	57
4.4	Semi-dense voxel grid	61
4.5	Summary	62
5	Concept and Design	63
5.1	Recursive least squares as 3D fusion approach	63
5.1.1	Weighted least squares and standard derivation of ML-Estimate	65
5.1.2	Depth fusion with recursive 3D fusion	70
5.1.3	Properties of RLSFusion	71
5.2	Regularized Recursive Fusion	74

5.2.1	Derivation of regularized least squared 3D fusion	75
5.2.2	Faster Convergence with regularized fusion	79
5.3	Outliers removal using spatial information	81
5.4	3D reconstruction framework	86
5.5	Summary	88
6	Evaluation	91
6.1	Quantitative evaluation	91
6.1.1	Outliers removal and memory utilization	96
6.2	Qualitative evaluation	98
6.3	Running time analysis	100
6.4	Summary	107
7	Conclusion and Future work	109
7.1	Future Directions	111
7.1.1	Adaptive depth denoising	111
7.1.2	Automated scene understating	111
7.1.3	Efficient data structure for large environments	112
	Bibliography	120
	List of Figures	125
	List of Tables	127
A	Appendix	129
A.1	Formulation of D and C matrices	129
A.2	Technical Requirements	131

Chapter 1

Introduction

The importance of robotic applications in everyday lives (besides industrial automation) has increased significantly over the years. Various time-consuming, sequential and tedious tasks are automated by the help of autonomous robotics. The usability of autonomous robots in everyday life is spread on a broad spectrum ranging from miniature floor cleaning robots to fully autonomous vehicles driving in unknown territories. Regardless of the size, environment and application domain of autonomous robots in general, the efficiency of a robot for a given task depends greatly on the real-time understanding of the surrounding environment.

Modern robots are equipped with depth sensor systems, such as laser-based range scanners, which allow them to perceive an environment as a 3-dimensional (3D) surrounding. In fact, the reconstruction and analysis of 3D depth data, and their representation in form of 3D maps, allows robotic applications to perform precise tasks such as navigation of autonomous vehicle without collision. Thus, research domains, dealing with applications of depth perception, are intensively studied. Developments in this domain have typically potentials with respect to social or economical impacts. Automated decisions (e.g. how to interact with objects using actuators, or how to avoid collision while navigating) depend on the modeled environment based on 3D depth data.

Unlike in a theoretically ideal system in which a depth sensor provides accurate 3D information, sensed depth values are prone to accumulating unwanted measurement noise. Although depth sensors acquire depth noise, the degree of noise in depth samples depends on various factors such as the distance between sensor and object, extreme lightning conditions, reflective surfaces, or multiple sensor-corrupting projective patterns. For handling additive

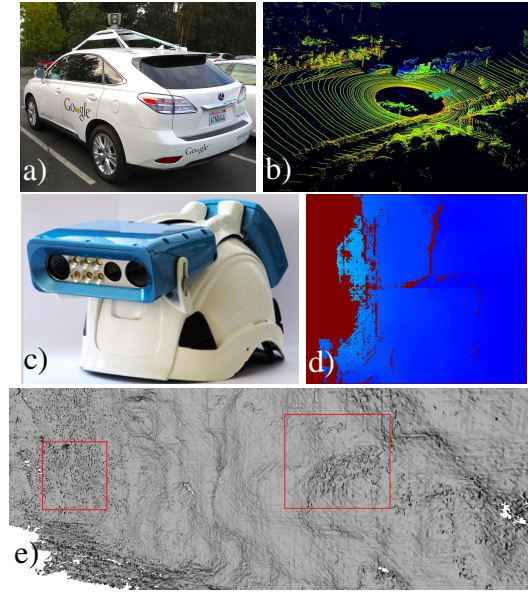


Figure 1.1: a) Laser depth sensor (LiDAR) mounted on top of autonomous vehicle, b) 3D points acquired from LiDAR, c) Stereo camera based IPS system, d) Color-coded depth map from IPS and e) Reconstructed 3D model of *mine* using stream of depth images with highlighted surface deformities.

noise in depth samples, various strategies have been proposed and implemented such as kernel-based filtering or variational de-noising methods. However, not all de-noising techniques can be implemented to handle noise in real-time.

1.1 Depth Imaging for 3D Reconstruction

3D laser scanners are used as depth perception systems in robotic applications in which accuracy and real-time availability of 3D data is crucial. Typical laser scanning systems sample the geometry of environment with a rotating head which result in a 3D point cloud which captures 360° surrounding area around vehicle as shown in Figure 1.1.b. Despite the accurate depth measurements by laser 3D scanners, usability of such sensors is restricted in mobile robotics due to their weight, cost and high power consumption.

For this reason, a low-cost stereo-scopic camera based 3D reconstruction has gained the interest of research communities. Mobile robots are required to process 3D information from depth cameras into understandable 3D reconstructed models in real-time. Although computationally expensive 3D reconstruction techniques such as *Structure from Motion* (SfM)

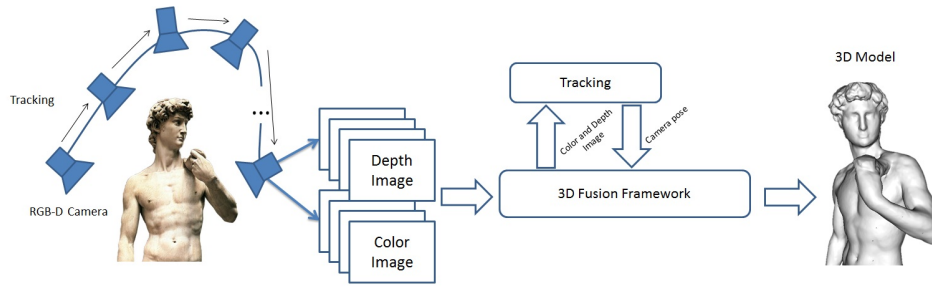


Figure 1.2: Incremental 3D fusion and reconstruction process.

algorithms allows high-quality approximation of the 3D geometry, applying these techniques in real-time scenarios is challenging. In general, low-cost depth sensors acquire relatively higher amount of measurement noise and outliers due to their depth acquisition principal compared to laser based scanners, this phenomena further degrades the reconstruction of the 3D model as shown in Figure 1.1.e where a stereo-scopic passive depth sensor is used to acquire 3D information. As a result, aforementioned issues subverts the quality of reconstructed 3D models, therefore the problem of handling noise and outliers in real-time becomes further challenging. For these reasons, enhancing the quality of acquired depth data by the means of denoising is the key aim of this thesis.

Stereo-scopic based depth sensors produce a stream of depth and color images in real-time, these depth images can be integrated (also referred to as fusion) using a volumetric integration technique (described in Chapter 2) to produce globally consistent 3D models. This process of 3D modeling from stream of depth and color images is shown in Figure 1.2. First, a 3D sensor (RGB-D camera in this case) observer the scene (in this case Michelangelo's *David*) and generates a stream of color and depth image as shown in Figure 1.2. These image streams are fed to the 3D fusion framework in which a VisualSLAM algorithm estimates sensor ego-motion for each instance of depth and color image, estimated camera tracking information along-with color and depth image streams are processed to produce high-quality 3D model of the observed environment. A crucial feature of the 3D fusion framework in Figure 1.2 is to cater acquired depth noise and outliers at the time of incremental processing and integration.

1.2 Challenges and the Research Question

The ability to improve the quality of the reconstructed 3D model from error-prone 3D depth data would allow various mobile robotic applications to estimate an accurate geometry of the environment. This challenging objective attracted several research communities to investigate and implement novel depth noise and outliers reduction techniques (discussed in Chapter 2). In principal, a depth outlier is a 3D sample point which lies either in front or behind the actual surface while a depth noise is an error-prone 3D sample having close proximity to the surface, therefore result in distorting the geometry of reconstructed model.

In early research endeavours, triangulation techniques have been employed for surface reconstruction directly from 3D samples. These techniques interconnect acquired 3D samples to represent surface geometry using triangle meshes Cazals and Giesen (2004). The aforementioned meshing technique presumes a well defined spatial distribution of 3D samples, however in practice this assumption is violated specially when dealing with multiple depth samples of target surface. In such cases, triangulation meshing tries to accommodate all points by producing undesirable triangles at random position and orientation.

For such reasons, the research community has revived the use of volumetric representation and integration Curless and Levoy (1996) in which multiple depth samples are represented as signed distance values from the estimated surface and standard marching cubes Lorensen and Cline (1987) algorithm is applied to acquire a globally consistent 3D model.

Volumetric integration of depth images reduces the effects of depth noise at the time of integration by fusing multiple noisy depth samples to estimate better understanding of the object geometry. Implicit representation of the estimated surface is updated using a weighted addition, it is therefore expected that the estimated implicit surface will eventually *converge* to the true surface. Furthermore, formulating an optimal weighting function which respects sensor characteristics as well as geometry of environment is a significantly difficult problem and varies greatly among different types of sensors.

In principal, volumetric integration techniques presume a globally consistent model, therefore untreated depth outliers produce surface patches which are inconsistent to the boundary of the surface. Statistical strategies which test every depth sample for proximity test among neighboring 3D data Rusu et al. (2008) have proven to be effective but lacks

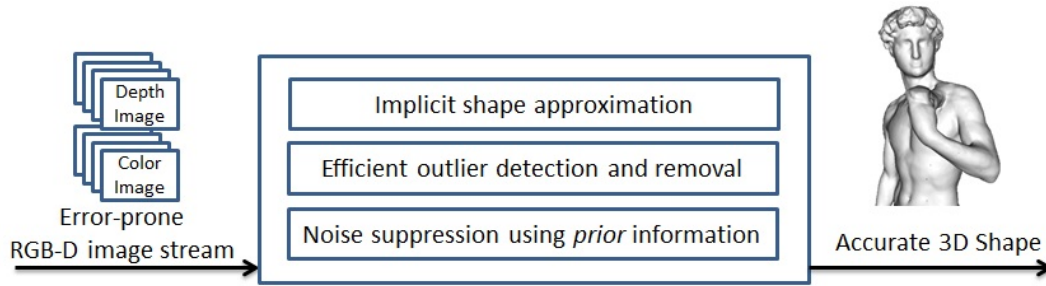


Figure 1.3: Proposed 3D reconstruction framework

computationally inexpensive profile which restrict these techniques to be implemented on real-time applications for mobile robots.

New strategies are therefore required to combine outlier removal techniques with noise reduction methods such that the process of integration is capable of handling erroneous depth information efficiently. In contrast to standard implementations of volumetric integration in which the weighting function is to be calculated prior to the reconstruction, new techniques are needed to handle spatial awareness of the environment to cater noise in 3D data. These challenges provided motivation to use relevant *a priori* information about sensor or environment, this leads to the ultimate question of this thesis:

How to efficiently reconstruct a 3D model of environment by reducing effects of noise and outliers inherently in real time scenarios?

This research question is addressed by a novel 3D reconstruction framework shown in Figure 1.3. Considering a set of error-prone 3D point clouds acquired from a depth camera, the framework applies geometry aware outlier removal filters which identifies and removes isolated depth samples. The proposed framework then applies novel total variation (TV) denoising on a novel implicit representation to enhance the quality of the reconstructed model while reducing the arbitrary surface deformities caused by depth noise. This resulting implicit representation of the depth data is then stored for integration of upcoming depth samples. This three stage process of acquisition, filtering and integration is repeated for each acquired depth image and implicit representation is updated respectively.

The research question can be subdivided into following objectives:

1. To develop a mathematical model which supports a priori information about environment

to support improved shape approximation and noise suppression in erroneous 3D samples.

2. To develop a computationally inexpensive numerical outlier removal filter which targets isolated depth samples on the basis of proximity to support real-time reconstruction.
3. To develop a reconstruction framework which accepts sequence of error-prone depth and color images and produce high quality 3D models of environment in real-time.

1.3 Contributions

In general, optimized depth images have the potential to increase the accuracy of robotic application in their respective tasks. However for our implementation we focus to apply depth optimization in robotic applications to aid autonomous navigation and understanding of environment in real-time. For this reason, the contributions of this thesis lies in both theoretical and practical domains to optimize the processing of depth images.

In summary, the main contributions of this thesis are:

- **Theoretical aspects & Implementation**

- Design and implementation of *Regularized Fusion (RFusion)*: A total variation filtering based 3D incremental fusion scheme is proposed, formulated and implemented which is capable of using *prior* smoothing knowledge to reduce depth noise at the time of integration in a real-time scenario.
- Design and implementation of *Spatial Outliers Removal Filter (SORF)*: A light-weight outlier detection scheme having linear complexity of $O(n)$ is proposed which uses spatial proximity cues to identify and remove explicit outliers.
- Design and implementation of *SmoothFusion*: A modular 3D reconstruction framework which encapsulates regularization aspect of *RFusion* to filter depth noise and utilize proposed *SORF* to remove explicit and isolated outliers in computationally efficient manner.

1.4 Thesis outline

The rest of this thesis is organized as follows:

- Chapter 2 provides a comprehensive literature review of contemporary 3D shape approximation and representation techniques. As it will be shown, modern techniques use *a priori* knowledge into shape approximation process to produce accurate 3D models with smooth surfaces. State-of-the-art 3D reconstruction frameworks are briefly reviewed and evaluated to highlight potential issues related with depth noise and outliers, this serves as problem statement for the proposed research. All reviewed techniques and frameworks are evaluated with respect to their robustness to noise, ability to deal with outliers, processing speed and overall accuracy.
- Chapter 3 describes aspects of methodology adopted in this thesis for prototyping and testing proposed framework. In addition, both quantitative and qualitative evaluation measures are presented highlighting details of test simulation, datasets and performance metrics used for benchmarks.
- Chapter 4 provides theoretical background focusing specifically on volumetric integration and presents analytical reasoning behind dense and semi-dense volumetric representation used in state-of-the-art frameworks.
- Chapter 5 provides theoretical aspects and design of proposed contributions. Firstly, a novel Recursive Least Square (RLS) based 3D fusion (RLSFusion) technique is introduced which enables possibility of functional extendability such as exponential forgetting and regularization in the 3D integration process. Secondly, the core concept of total variation based filtering for implicit shape regularization in incremental 3D reconstruction is presented and implemented in the form of *RFusion*. Finally, internal workings of *SORF* are introduced and importance of using robust spatial outliers removal filter in real-time 3D fusion framework is briefly discussed.
- Chapter 6 presents quantitative evaluation measures used to evaluate reconstructed 3D models from proposed frameworks with state-of-the-art techniques. In cases where quantitative evaluation is not possible, screenshots from both proposed and existing frameworks are provided to aid qualitative comparison. Finally, a comprehensive per-frame running time analysis is presented which compares the execution of proposed

framework in comprehensive detail.

- Chapter 7 summarizes the contributions of our systems and propose several directions for future work.

1.5 Summary

This chapter highlighted significance of incremental 3D fusion in modern robotic applications from 3D depth sensing systems. Key challenges such as depth outliers and noise were identified and research question is formulated which subdivides problem statement into three basic inter-related objectives, i.e. formulation of implicit shape approximation, efficient outliers detection and removal scheme and smoothing using *prior* knowledge.

As the first step to achieving these objectives, next chapter provides critical literature review of existing 3D shape approximation techniques and state-of-the-art 3D reconstruction frameworks.

Chapter 2

Related Work

This chapter will review traditional shape modelling techniques and fusion frameworks used to reconstruct models from 3D points. Initially, general shape representation methods are reviewed to establish baseline on their suitability for numerical shape approximation and modelling. Thereafter, modern approaches which process regularization and smoothing using a priori knowledge are reviewed and evaluated. At last, state of the art fusion frameworks which use 3D depth data to reconstruct a virtual environment are briefly discussed and evaluated.

3D shape approximation and modelling is a traditional yet active research problem which is addressed by computer graphics and vision community for past two decades. In early days, direct triangulation based techniques for modelling were proposed and implemented (see Edelsbrunner and Mücke (1994)), these techniques were straightforward and effective when the sampled 3D points were ordered and pre-sampled however lacked the capability to handle scattered and unordered depth data. A novel volumetric representation technique was presented by Curless and Levoy (1996) which illustrated the capability of integrating multiple range images in a volumetric fashion to construct complex models, due to lack of computation resources at that time this technique was unexplored for up-to 15 years and was revived by Newcombe et al. (2011). The difficulties of triangulation based modelling were later addressed by Alexa et al. (2003), Calakli and Taubin (2011) and Kazhdan and Hoppe (2013) by using *a priori* knowledge (also referred to as *prior*) in modelling phase, this allows the reconstruction framework to handle redundant and uneven samples and produce smooth 3D models. Many research groups integrated the use of smoothness priors in their application-specific methods, some considered repetitive structures Pauly et al. (2008), Berner et al. (2011)

while other experimented with geometry of the sampled scene Yingze Bao et al. (2013). The concept of smoothness priors is also extended to global and piecewise smooth geometries by Avron et al. (2010) and Calakli and Taubin (2011).

Section 2.1 provides a brief review of general shape representation techniques while focusing on their usability of handling unordered 3D points. Since the concept of using smoothing priors is basis of presented research objective, relevant notable contributions are briefly addressed in Section 2.2. Depth image smoothing techniques which are relevant to proposed research are briefly reviewed in Section 2.3 to establish baseline. Section 2.4 reviews current state-of-art reconstruction frameworks which utilize a volumetric representation for depth integration. The discussion is then summarized in Section 2.6 where generality, computational effectiveness, robustness and accuracy are elaborated.

2.1 General Shape Representation

2.1.1 Simplex Representation

Shape representation using polygon meshes or more generally simplexes is considered as a standard practice in computer graphics community. Many interactive visualization applications such as virtual reality, augmented reality and video games use simplexes to process and visualize 3D models. This concept originated from research by Bowyer (1981), in which a triangle meshes connecting tetrahedra via Delaunay-triangulation are used to approximate shapes from 3D points in an automated fashion. Edelsbrunner and Mücke (1994) proposed α -shapes algorithm (Figure 2.1) for creating topological correct surfaces from 3D points using polygon meshes. The method connects neighboring 3D points using triangles while the α value controls the acceptable euclidean distance between connected sample. Since the value of α is strongly dependent on factors such as detail and scale of the model, the selection of appropriate value for α requires user interaction by an expert. Later, a more adaptive region growing technique called *Ball-Pivoting Algorithm* (BPA) was proposed by Bernardini et. al (1999) in which a user defined a virtual ball having radius ρ is used to determine valid 3D points. In principal, α -shapes, BPA and all Delaunay-triangulation methods are incapable to handle noisy or redundant 3D samples Bodenmueller (2009). Since acquired 3D points are prone to collect various types of noise in the scanning process, the noise sensitive behaviour produces abrupt

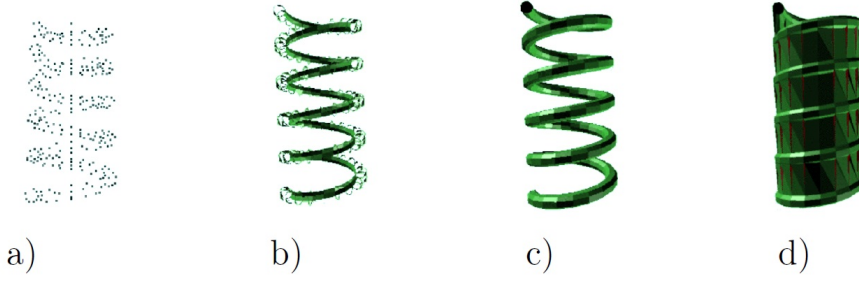


Figure 2.1: The α -shapes algorithm. a) Input samples, b-d) reconstruction with increasing α Edelsbrunner and Mücke (1994).

geometric defects in the reconstructed model. Henceforth, the computer vision communities avoid the use of simplexes based representation while approximating the shape with 3D points.

2.1.2 Parametric Representation

Parametric algorithms handle the problem of non-uniform sampled 3D points by fitting a spline to approximate contours of the surface, these methods are well-known for signal interpolation as well as approximation. For instance, a function $f(u, v) : \mathbb{R}^2 \mapsto \mathbb{R}$ which returns the height of the surface it is approximating at any given values of u and v (Figure 2.2.a). Various approximating functions can be stitched together to approximate complex globally consistent 3D models (Figure 2.2 b), however a traditional parametric surface reconstruction algorithm consists of two steps:

1. **Partitioning:** A clustering technique (for e.g. Sheffer et al. (2007)) is applied
2. **Parameterization:** A local surface with corresponding height parametrization model f is extracted via optimization

A standard practice is to employ a *least square* function for each segment which minimizes

$$\min \sum_i^N \|h(\mathbf{P}_i) - f(u_i, v_i)\|_2^2 \quad (2.1)$$

where $\mathbf{P}_i \in \mathbb{R}^3$ is the i^{th} sample 3D point, $h(\mathbf{P}_i) : \mathbb{R}^3 \mapsto \mathbb{R}$ is the segment height and $(u_i, v_i) = proj(\mathbf{P}_i)$ is the projection of \mathbf{P}_i on the corresponding segment plane partition from step 1.

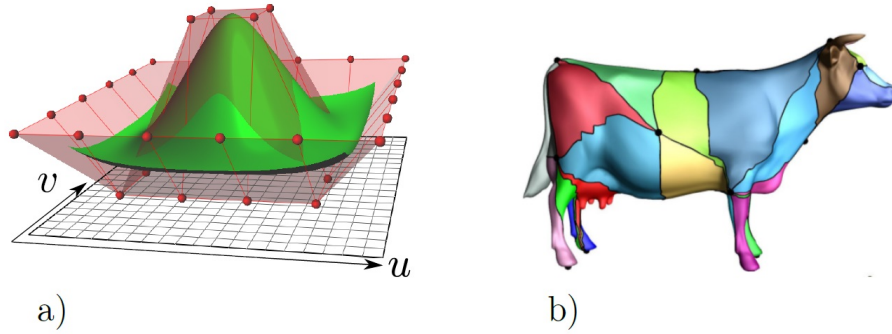


Figure 2.2: a) Smooth surface model via NURBS. b) A set of parametric shapes combined to a global consistent surface. Schreiner et al. (2004).

Bezier curves (see Agoston and Agoston (2005)) or *Non-Uniform Rational B-Splines* (NURBS) model are then employed to approximate model f for each segment.

Although parametric shape approximation and reconstruction has the capability to produce smooth 3D surfaces for non-uniform 3D point sets, however an extra computational task of combining local segments to produce a global continuous shape is required. This combinatorial task is computationally expensive as shown by Floater and Hormann (2005). For this reason, the local approximation and representation is adopted and will be examined in Section 2.2.

2.1.3 Implicit Representation

The *Signed Distance Field* (SDF) is a special case of shape representation having high potential of usability in applications such as motion planning (Hoff III et al. (1999)), multi-body dynamics (Guendelman et al. (2003)), collision detection and cloth animation (Bridson et al. (2003)) and camera movement tracking (Canelhas et al. (2013)). The shape of the desired object is represented by an implicit indicator function $f(x)$ which classifies space around the object surface as either *inside* $f(x) < 0$ or *outside* $f(x) > 0$ where $x \in \mathbb{R}^3$ is the spatial coordinate of sampled 3D space. In principal, a surface of the object is set of all x where f produces zero as illustrated in Figure 2.3. The 3D space is divided into smaller elements called *voxels* which contains implicit indicator value of SDF. Given enough computational and memory resources, the implicit modelling can be extended to work with streams of depth and color images from a RGB-D sensor (such as Microsoft Kinect) to produce high quality 3D models of environment (Newcombe et al. (2011)).

Main drawback of using implicit representation for 3D modelling comes from the division

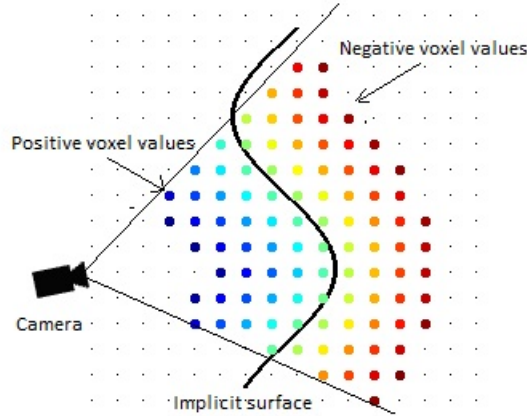


Figure 2.3: A Signed Distance Function (SDF) on a fine grid.

of 3D space into dense cells regardless of whether a cell contains meaningful SDF information or not. This inefficient memory utilization makes implicit shape modelling infeasible for large-scale environments. Representing an area of $100 \times 100 \times 100 \text{ m}^3$ using 1 cm *voxel* resolution using standard floating point values would require 4000 GB of memory to encode implicit SDF. State-of-the-art volumetric reconstruction frameworks (such as Steinbruecker et al. (2014), Kähler et al. (2015)) employ narrow-band surface localized voxels to facilitate large-scale environment reconstruction. Further explanation and incremental integration based applications of implicit representation are discussed in Section 2.4.

2.1.4 Surface Splatting

Surface splatting is a specialized case of the point based representation which is targeted to render millions of 3D points independently (more generally vertices) in real-time using modern rendering framework such as OpenGL. State-of-art reconstruction frameworks (such as Whelan et al. (2015), Keller et al. (2013)) employ surface splatting to accommodate dynamics of environment by adding or removing vertices in real-time. In principal, elliptical surfaces having associated confidence, color and normal information (called *splats*) are used to represent vertices. In order to integrate the curvature information using discrete splats, each splat is processed with *Elliptical Weighted Average* (EWA) to produce high quality texture blending while maintaining a low memory profile. Zwicker et al. (2003) demonstrated the potential of surface splatting to produce high-quality 3D models from both scan and synthetic objects (Figure 2.4)

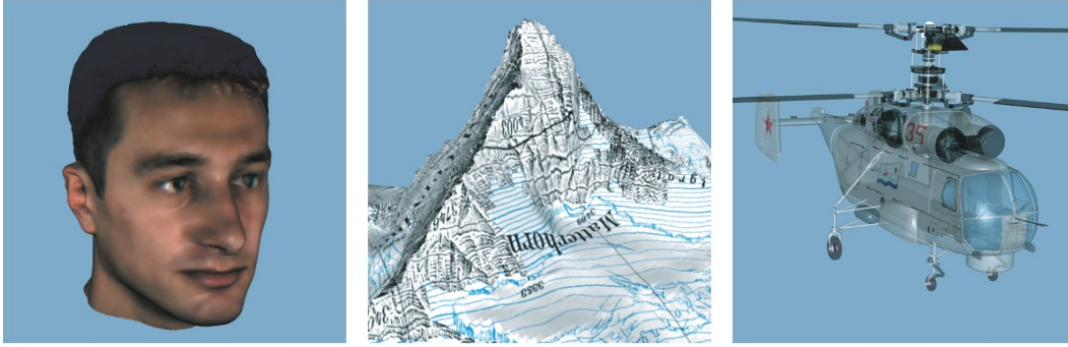


Figure 2.4: Surface splatting of a scan of a human face, textured terrain, and a complex point-sampled object with semi-transparent surfaces. (Zwicker et al. (2003))

Although surface splatting has a high potential for applications in real-time dynamic 3D modelling using high-quality depth sensor such as *Kinect* and *Kinect v2*, however inability to handle low-density and error prone depth data is main disadvantage for using this representation technique.

The shape representation techniques discussed so far enable the approximation of shape geometry from 3D points. In various applications, it is preferred to restrict the generality of representation approach in the favour of approximation quality. Examples in the upcoming section will discuss and illustrate the process of utilizing a priori information to cater error-prone measurements while producing smooth surfaces.

2.2 Prior Based Shape Approximation

In computer graphics and vision algorithms, prior information is used to aid reconstruction and rendering processes. This integration of prior knowledge is essential in automated 3D shape approximation since all depth sensing systems are prone to introduce measurement errors depending upon the type and working of the sensor system. To maintain relevancy with the research objective while avoiding exploration of inessential techniques, two general prior types are identified and briefly explained in the upcoming sections.

2.2.1 Regular Priors

Various everyday objects exhibit repetitive structures and 3D reconstruction frameworks can use this vital information to produce life-like 3D models, these repetitive structure are

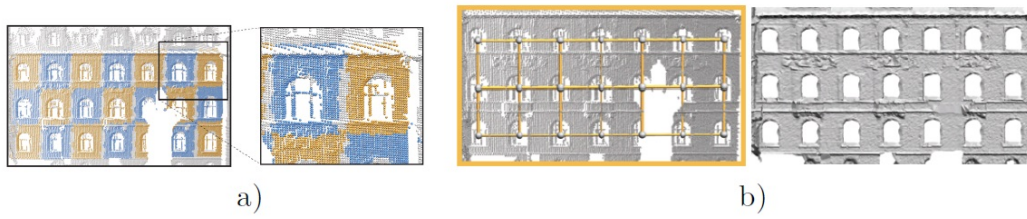


Figure 2.5: Detecting repetitive structures (a) enables hole filling (b) in structured environments. (Pauly et al. (2008)).



Figure 2.6: Similar objects are scaled to match repetitive patterns. Red: strong deformations. Green: small deformations. (Berner et al. (2011)).

commonly referred to as regular priors. Pauly et al. (2008) proposed a method to cluster point clouds into repetitive segments and employ this vital information for hole filling of structured environment. The potential of this scheme is illustrated in Figure 2.5 where a complete wall segment is inferred from repetitive structure. Berner et al. (2011) extended the concept of using regular priors to general partial symmetries, in which low dimensional shape space is represented and used for matching. In principal a basic structure is identified and non-rigid deformations of this portion are matched with similar areas, this is illustrated in Figure 2.6 where strong deformation matches are shown in red while small variations are shown in green. Berner et al. (2011) suggested the use of supervised segmentation where the matches are ambiguous Figure 2.6.

Supervised integration of prior information is further investigated by Arikan et al. (2013) and Sharf et al. (2007) in which relational based similarities are marked by expert user. Yingze Bao et al. (2013) proposed to perform semantic classification by relating observed images and sparse point cloud from known database (Figure 2.7) using pre-learned approaches to reduce interactive intervention from expert user. Aforementioned methods are designed to perform well in presence of known objects or fractals in shape subspace, however in common scenario when the scene consists of unknown objects, these method does not provide any

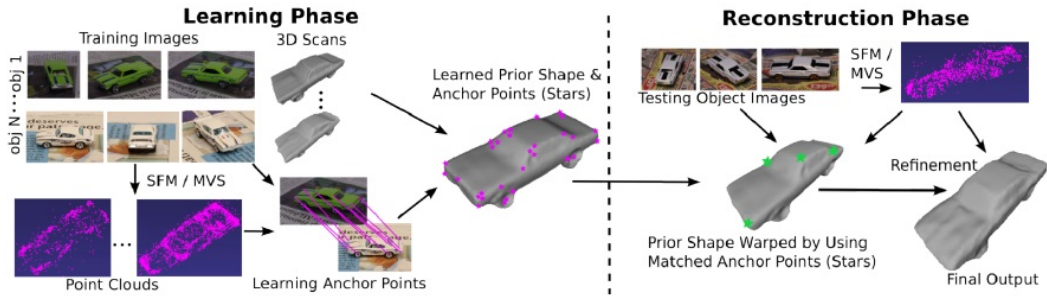


Figure 2.7: Learning priors from images for reconstruction (Yingze Bao et al. (2013)).

advantage. Thus instead of using a specialized structure, generalized and straightforward prior models are required which can be used in an unknown environment. Smooth and planar surfaces have been identified as common characteristics for a large variety of scenes, therefore using this basic information into the shape reconstruction approach does not confine underlying algorithms to specific environment. The smoothness assumption is divided into local and global smoothness priors, both are highly important and relevant to proposed research and discussed briefly in the upcoming section.

2.2.2 Local Smoothing Priors

A novel and robust local smoothing approach which is specifically designed to handle redundant data as well as to remove noise was proposed by Alexa et al. (2001). An implicit surface is approximated for every point in sampled data by using neighboring points. This type of neighborhood approximation methods are also known as *moving least square* (MLS) techniques. Strength of smoothness can be controlled by varying the weighting function θ to reduce surface deformities caused by error-prone depth measurements. The approximation process is a two step process, a local **reference domain** (plane) $h(x)$ for the point x is extracted which minimizes a local weighting sum of the square distance of points p_i to the plane in the first step using:

$$h(x) = \arg \min_{n,d} \sum_i (\langle n, p_i \rangle - d)^2 \theta(\|p_i - x\|_2) \quad (2.2)$$

Where n is approximated normal of p_i and d is distance of p_i from the local plane. In the second step, a local bivariate smooth polynomial approximation function $f(x)$ is estimated via LSQ

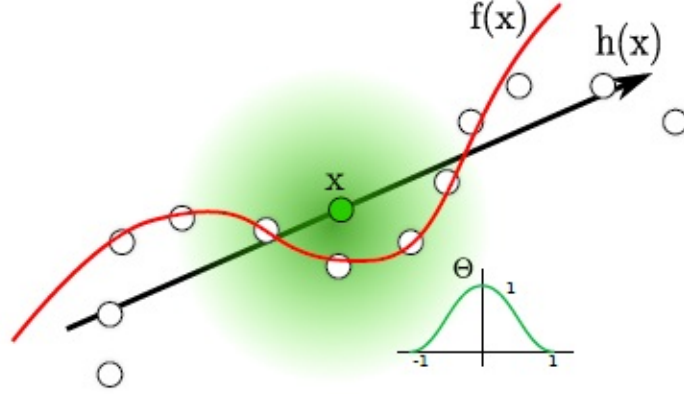


Figure 2.8: Local surface approximation by Alexa et al. (2001).

(Figure 2.8) to the height value $h(p_i)$ for each sample:

$$\arg \min_f \sum_i (f(u_i, v_i) - h(p_i))^2 \theta(\|p_i - x\|_2) \quad (2.3)$$

All available points p_i are re-sampled with the estimated shape and rendered using a variant of point based rendering proposed by Rusinkiewicz and Levoy (2000). This surface approximation approach received much attention due to its handling of noisy and redundant point samples while producing smooth continuous surface representation. Further experimentation by Kolluri (2008) enabled the control of smoothness via point-based blending which employs point normals n_i into a shape function as:

$$f(x) = \frac{\sum_i n_i \cdot (p_i - x) \varphi(\|p_i - x\|_2)}{\sum_i \varphi(\|p_i - x\|_2)} \quad (2.4)$$

where φ is *sharpness* and the weighting function is defined as:

$$\varphi(r) = \frac{1}{r^2 + \epsilon} \quad (2.5)$$

In principal, both $f(x)$ and φ can be controlled by a user-defined parameter ϵ , Figure 2.9 demonstrates effects of varying ϵ to acquire an appropriate smoothness level. Although MLS based techniques are efficient to handle sampling noise and redundancy, however they fail to produce accurate 3D models when the samples are sparse. This problem is addressed by Öztireli et al. (2009) where they extended the polynomial model from Alexa et al. (2001) by

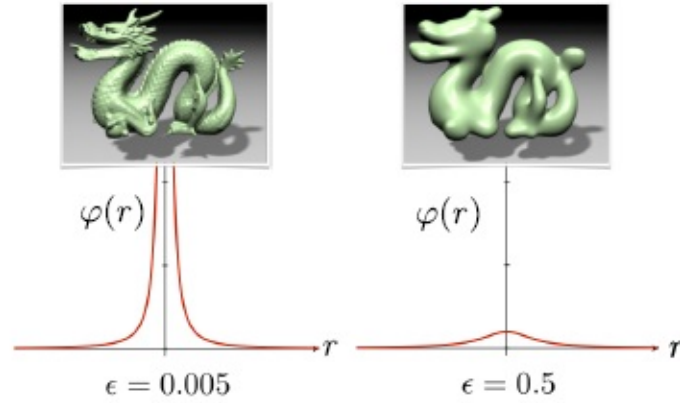


Figure 2.9: Smoothness of point-to-plane blending controlled by ϵ by Kolluri (2008).

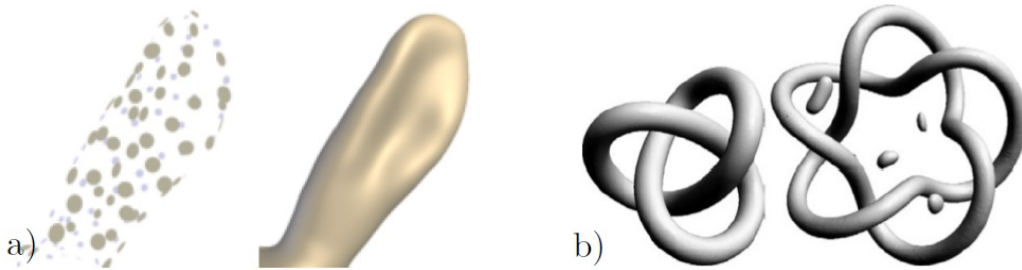


Figure 2.10: a) Sphere fitting from sparse samples Öztireli et al. (2009) b) MLS without and with outliers Ohtake et al. (2005).

fitting spheres of variable radius to local samples.

All aforementioned local approximation approaches tend to accommodate each sample to a consistent surface, this property of least square estimation fails to accommodate strong sampling noise and/or outliers and produce artifacts as shown in Figure 2.10.b. Specialized statistical outliers removal techniques such as Rusu et al. (2008) can be applied on 3D samples to reduce outliers, since removing outliers is an essential part of the research objective these techniques are briefly described in Section 2.5.

2.2.3 Global Smoothing Priors

Global shape approximation techniques exploit the implicit representation of an underlying surface to create a globally consistent shape. Carr et al. (2001) proposed one of the first such global shape approximation methods in which the implicit function $f(x)$ which approximates

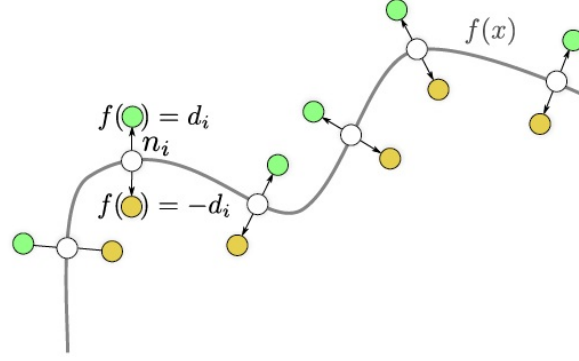


Figure 2.11: The implicit function $f(x)$ approximating surface points.

off-surface points is defined as

$$f(x) = \sum_i \alpha_i \varphi(\|x - c_i\|_2) \quad (2.6)$$

where α_i are weights and c_i are centres for i^{th} Radial Basis Functions (RBF). The technique exploits included normals information n_i to further enhances the approximation of an implicit function as shown in Figure 2.11. An optimal shape function gives zero at the sample i.e. $f(p_i) = 0$ and d_i at the off-surface points $f(p_i + \epsilon_i n_i) = d_i$. A convex LSQ minimization task which is used to calculate α_i is defined as:

$$\alpha = \arg \min_{\alpha} \sum_i f(p_i)^2 + (f(p_i + \epsilon_i n_i) - d_i)^2 + (f(p_i + \epsilon_i n_i) + d_i)^2 \quad (2.7)$$

The smoothing effect for the final representation can be controlled by the polynomial degree of RBF and offset distances d_i . Good extrapolation capabilities combined with achieved smoothness allows global approximation techniques to deal with irregular sampling issues while maintaining details in high density areas. The selection of an appropriate offset distance d_i to deal with noise while maintaining details is however a critical problem.

Hornung and Kobbelt (2006) introduced a simplified discrete variant of the global approximation to address the off-surface distance. The 3D space around samples is divided into a *narrow-band* voxel grid, distance values are calculated from the nearest sample to the center of each voxel and stored in the corresponding voxel (Figure 2.12). Graph-cut techniques from Boykov et al. (2001) are then applied to extract the shape from voxels. This technique is not suitable for large datasets containing millions of 3D points due to high computational

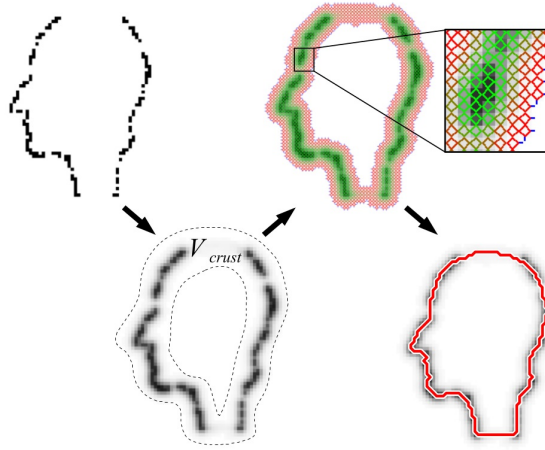


Figure 2.12: *Narrow-band* voxel grid around samples for graph-cut, Hornung and Kobbelt (2006).

complexity.

Various implementations and improvements have been proposed over the years to further extend RBF based global approximation. However the most relevant and notable contribution is proposed by Walder et al. (2006) involving a two-step processing. In the first step, small regions are approximated independently via a global RBF (Figure 2.13). In the second step, a compound RBF which compactly supports local approximations is estimated. Walder further proposed a regression model which forces sample normals n_i to align with shape function f such that:

$$\nabla f(p_i) = n_i \quad (2.8)$$

In terms of a convex optimization task, this constraint emulates the cost term. Since this method enforces a locally defined function to work with a globally smooth RBF, this imposition however leads to over-smoothing. Calakli and Taubin (2011) extended Walder's regression model to work with a discrete form on an octree and proposed a second order minimizer, which led to *Smooth Signed Distance Fields* (SSDF) surfaces. Figure 2.14 shows the extrapolation behaviour of SSDF reconstruction on non-sampled 3D points.

All of the aforementioned global methods are targeted to approximate a consistent surface from given 3D points at the time of execution. In case of incremental updates specially in the real-time 3D reconstruction, these methods fail to accommodate latest updates in input 3D points. This challenging rigid behaviour of global smoothing methods provided main

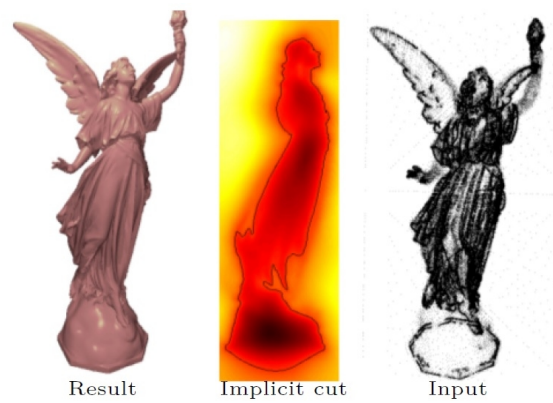


Figure 2.13: A smooth and global implicit shape extracted via radial basis functions, Walder et al. (2006).

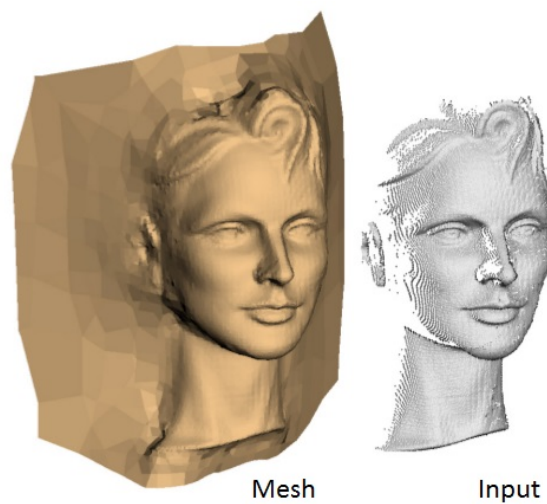


Figure 2.14: Surface reconstruction using SSDF by Calakli and Taubin (2011).

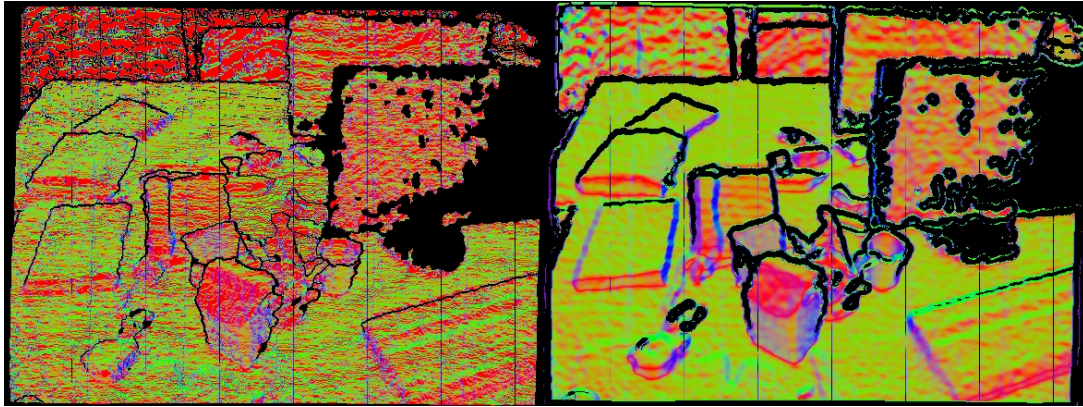


Figure 2.15: Surface normals from raw depth image (left) vs smooth depth image (right). Canelhas et al. (2013)

motivation for the proposed research to explore incremental 3D fusion techniques which are capable of approximating real-time 3D models of the environment. These models are briefly discussed and evaluated in Section 2.4. Family of depth smoothing techniques which are designed to reduce noise in depth images instead of 3D samples are evaluated in the upcoming section.

2.3 Depth Map Smoothing

The process of representing a 3-dimensional object using series of range values starting from camera origin to the surface of the object in the form of a 2D image (also referred to as depth map) is a well established norm in computer graphics and vision community. A measurement error in the form of added noise is accumulated in the depth acquisition process, untreated depth noise produces abrupt and geometric deformities in the 3D reconstruction. To cater this problem, various depth map smoothing algorithms have been investigated, proposed and implemented in the past decade. Since properties of the depth noise are different to that of normal color or gray-scale image noise, applying legacy smoothing filters such as low-pass filter introduce further surface deformities in the reconstructed 3D model.

Newcombe et al. (2011) introduced a modified edge aware bilateral filter to produce discontinuity preserved depth map from raw depth map acquired by the Kinect system in real time. Surface normals are commonly used to visualize depth smoothing effects by applying bilateral filter, Figure 2.15 highlights depth smoothing effects with bilateral filtering. Zhao

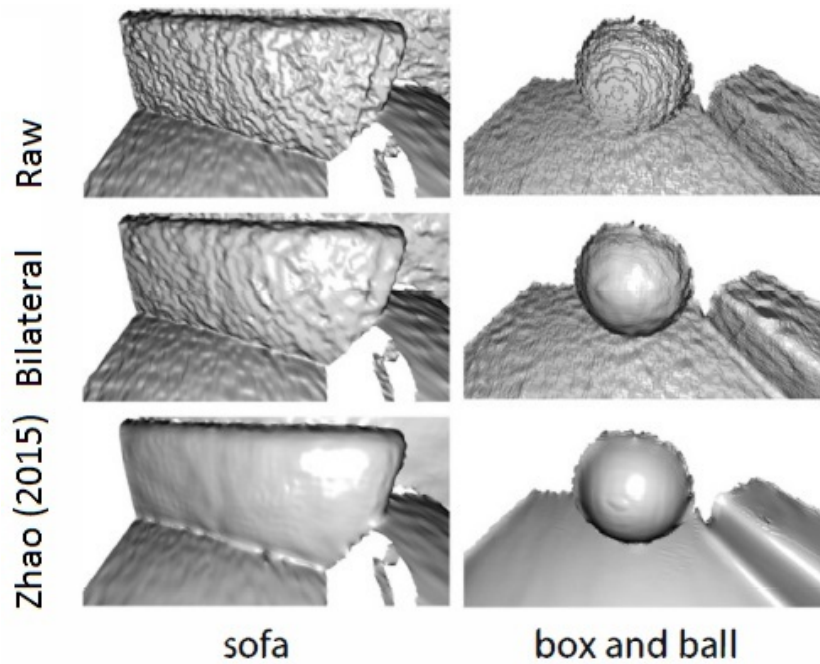


Figure 2.16: 1st row: 3D surface from Raw Kinect depth image, 2nd row: Using bilateral smoothing and 3rd row: Smoothing with Zhao et al. (2013)

et al. (2013) proposed a specialized depth filtering method which employs surface orientation analysis per pixel surface orientation analysis to further enhance the smoothing process. Promising results have been demonstrated by Zhao et al. (2013) as shown in Figure 2.16.

All of the previously mentioned depth smoothing algorithms are designed to handle Kinect-like depth noise, however depth images estimated from stereo images are prone to a higher intensity of depth noise due to estimation mis-match (i.e. sudden change in estimated surfaces) or texture-less or self-similar environment. Ranftl et al. (2012) proposed a stereo model featuring a second-order regularizer which reduces estimation errors and produces smooth depth images. Balzer and Soatto (2013) proposed a similar optimization technique in an iterative fashion to smooth surface deformities in multi-view stereo image based 3D reconstruction. Graber et al. (2015) argued that unconstrained total variation based regularization techniques are somehow prone to produce staircase artifacts (Figure 2.17.b) since they overlook 3-dimensional geometry of the perceived environment.

Edge-aware bilateral smoothing techniques and their variants are robust in nature however lack the capability to handle high-intensity noise. Contrarily, total variation based regularization methods respect geometry of surface at the expense of computational complexity.

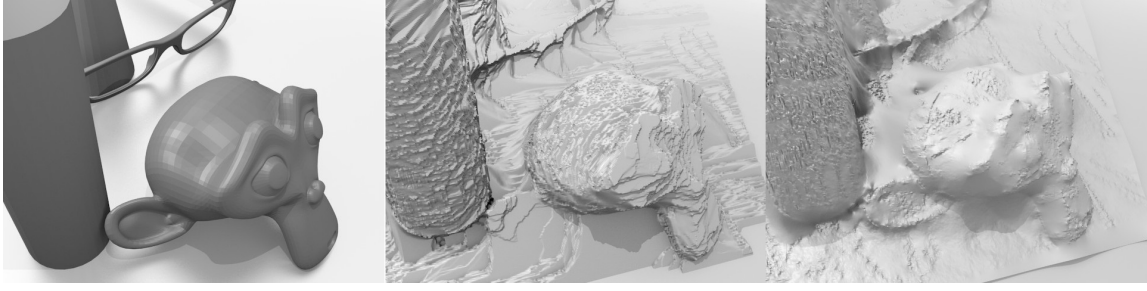


Figure 2.17: a) Ground truth surface, b) Total variation regularization and c) Minimal-surface regularization by Graber et al. (2015).

Therefore, a novel depth smoothing mechanism is needed to handle both kinect-like noise and stereo estimation depth noise while maintaining a low computational complexity profile.

2.4 Incremental 3D Fusion

Curless and Levoy (1996) proposed a novel implicit volumetric method targeted to support the reconstruction of complex models from range images. The potential and simplicity of volumetric method to handle incremental updates in the form of range images motivated various researchers to extend the core concept to utilize modern computational resources such as Newcombe et al. (2011), Whelan et al. (2012) and Kähler et al. (2015) etc. In principal, pre-aligned range images are represented and updated incrementally as weighted signed distance functions stored in a predefined voxel grid. Multiple error-prone observations of the particular region of interest from either single or multiple-views reduces acquisition noise and result in a high quality approximation of 3D object. An underlying volumetric representation method transforms the range image R_i to a signed distance function value $d_i(x)$ from a surface and weights $w_i(x)$. A simple truncation mechanism ensures that values of $d_i(x)$ are bounded within in D_{min} and D_{max} , this truncation plays a vital role in determining the proximity of a particular voxel near suspected surface. The implicit surface (also known as zero-crossing) can be extracted by casting a ray from the sensor position to each voxel and registering a zero-crossing as shown in Figure 2.18.a.

Incremental updates on the volumetric grid are carried by following equations:

$$D_{i+1}(x) = \frac{W_i(x) + D_i(x) + w_{i+1}(x)d_{i+1}(x)}{W_i(x) + w_{i+1}(x)} \quad (2.9)$$

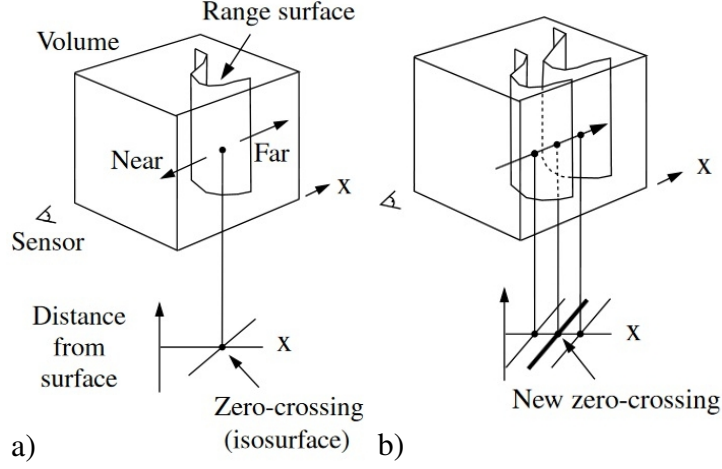


Figure 2.18: a) A range surface along x-axis from sensor position b) two range-surface are integrated to form a new zero crossing.

$$W_{i+1}(x) = W_i(x) + w_{i+1} \quad (2.10)$$

where $D_{i+1}(x)$ and $W_{i+1}(x)$ are cumulative signed distances and weight functions for all valid voxels $x \in \mathbb{R}^3$ after integrating the i th range image as shown in Figure 2.18.b.

The volumetric nature allows this representation scheme to integrate multi-view range images to form a consistent 3D model. This behavior is demonstrated in Figure 2.19 in which two separate cross-sections of volumetric SDF data 2.19.a and 2.19.b are integrated using Equation 2.9 and 2.10. Usually, a *space-carving* procedure is applied to identify potential voxels followed by the iso-surface extraction to render 3D models. Since contents of the voxel grid are updated in an incremental fashion, Curless and Levoy (1996) suggested to employ a fast marching cube algorithm (Lorensen and Cline (1987)) which can be initiated on demand.

Main drawback of using a dense volumetric grid for the incremental 3D reconstruction comes from the excessive use of memory and computational requirements as described in Section 2.1.3. State-of-the-art volumetric reconstruction frameworks utilize multi-threaded architecture of modern CPU and GPU to facilitate large scale environment. Relevant and notable contributions which extends the capability of volumetric method to large-scale environment and real-time modeling are briefly reviewed and their performance is evaluated in Section 2.6.

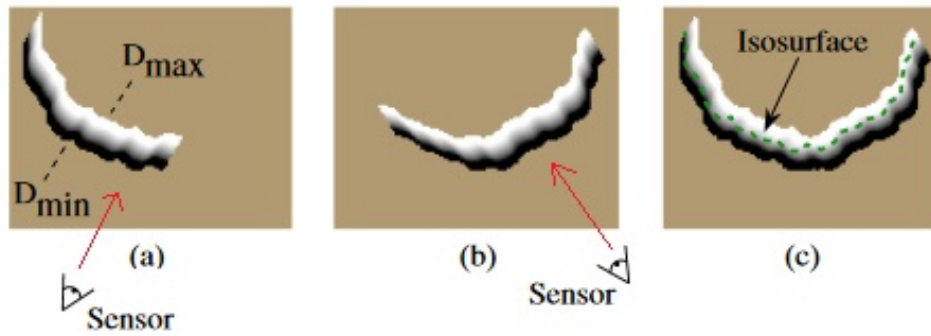


Figure 2.19: a and b) SDF values from multiple views in cross-section of volumetric grid c) integrated SDF values to form compound iso-surface Curless and Levoy (1996).

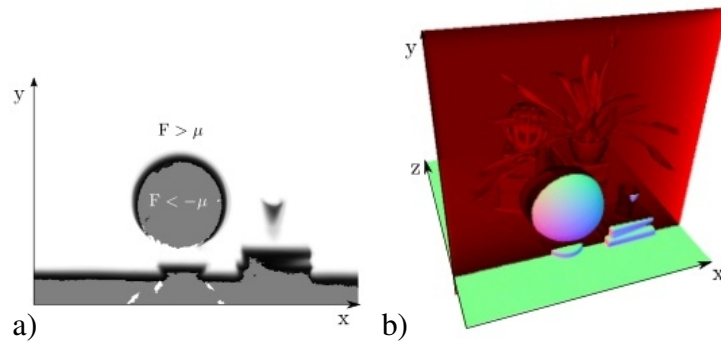


Figure 2.20: a) Slice of SDF volume demonstrating potential truncation mechanism and b) overall 3D volume (Newcombe et al. (2011)).

Newcombe et al. (2011) extended the concept of incremental volumetric 3D fusion with real-time camera pose estimation using Iterative Closest Point (ICP) tracking. This extension enabled low-cost depth scanning devices such as Kinect to reconstruct small scale environment as shown in Figure 2.20 where a slice through the signed distance function F highlights the use of truncation mechanism (i.e. validity criteria for each voxel v : $\mu \leq v \leq -\mu$). The surface of volumetric data is extracted with the help of ray-casting from viewing camera as suggested by Curless and Levoy (1996).

The use of GPU processing and memory enabled KinectFusion to reconstruct a 3D model of the environment in an online fashion. In principal, GPU processing threads are designed to perform simplified and repetitive tasks using massively parallel processing architecture. This mechanism restricted KinectFusion to work in small-scale environments. Newcombe et al. (2011) further suggested to use *frame-to-model* tracking for the pose estimation and

reconstruction, this localization mechanism utilizes available memory in an optimal way while producing globally consistent surfaces as shown in Figure 2.21 where the sensor is rotated around the area of interest.

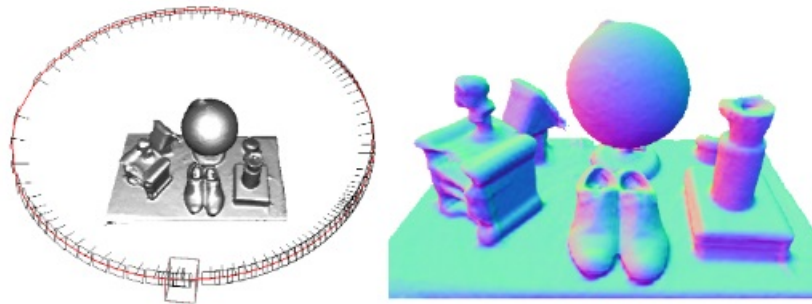


Figure 2.21: Camera tracking information visualized around region of interest (left) and reconstructed 3D model (right) (Newcombe et al. (2011))

Roth and Vona (2012) presented a novel memory efficient approach of a moving TSDF volume from one location to another with respect to the camera moment, this allows an active TSDF volume to be remained in fast acting memory while inactive parts can be moved out of the memory on-demand. This technique is targeted to accommodate sensor movement and reconstruction, however a rigid transformation combined with movement of TSDF volume accumulates localization drift and may result in multiple copies of misaligned surfaces. Figure 2.22 demonstrates the working of moving volume approach where the initial TSDF volume (left) is remapped to align with updated TSDF volume (middle) using a fixed relative transformation.

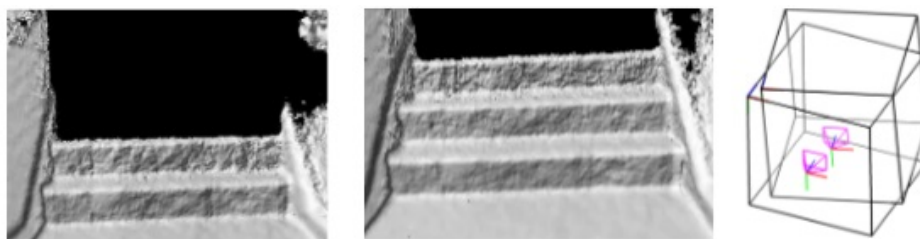


Figure 2.22: Initial TSDF volume (left), updated TSDF volume after integration (middle) and movement tracking information (right) (Roth and Vona (2012))

Whelan et al. (2012) proposed *Kintinous* as an extension of the KinectFusion which uses incremental triangular meshes in addition to volumetric mapping to handle large-scale

reconstruction, Figure 2.23 demonstrates the effectiveness of *Kintinous* to reconstruct relatively large-scale environment consisting of a two story apartment having multiple rooms.

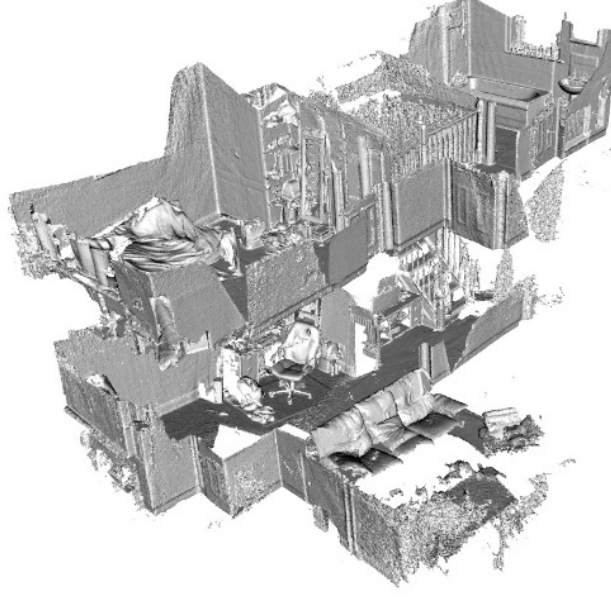


Figure 2.23: Large-scale 3D reconstructed model of apartment from *Kintinous* (Whelan et al. (2012))

Nießner et al. (2013) introduced a novel voxel hashing data structure targeted to achieve real-time management of implicit volumetric surfaces in the forms of voxel blocks from GPU's memory and processing resources. The proposed streaming in/out mechanism for voxel blocks eliminated spatial restrictions from 3D reconstruction while retaining the quality of reconstructed 3D models from degradation. An efficient GPU accelerated *hash table* is used to allocate voxel blocks in the proximity to surface geometry, each voxel block is accessed using an integer world coordinate (x, y, z) . All active world coordinates (x, y, z) are mapped to hash value $H(x, y, z)$ using the *hashing function*:

$$H(x, y, z) = (x.p1 \oplus y.p2 \oplus z.p3) \bmod n \quad (2.11)$$

where $p1, p2$ and $p3$ are large prime numbers and n is hash table size. A strict streaming in/out mechanism which checks each voxel block against the camera frustum is responsible of the data management, this ensures that active voxel blocks remain in the fast acting memory

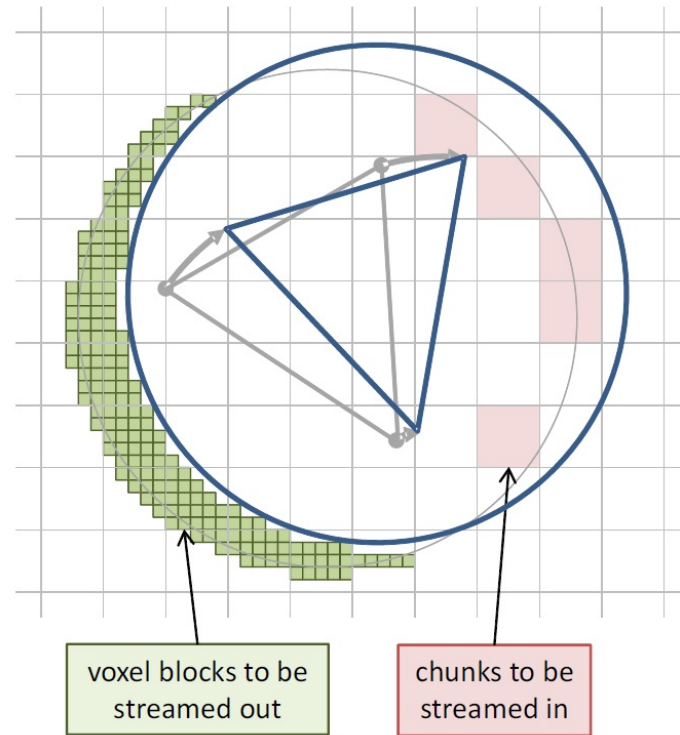


Figure 2.24: Streaming in/out process of voxel blocks from Nießner et al. (2013)

while dormant voxel blocks do not consume memory resources. Figure 2.24 demonstrates this streaming in/out process in which the camera is moving from left to right and respective voxel blocks are flagged accordingly.

The concept of voxel block hashing inspired Kähler et al. (2015) and Steinbruecker et al. (2014) to implement very fast state-of-the-art real-time 3D reconstruction named *InfiniTAM* and *FastFusion* respectively. All aforementioned incremental 3D fusion techniques including state-of-art frameworks rely on either the quality of depth measurements or the amount of samples for a specific surface region from RGB-D cameras such as Kinect and Kinect v2 however dealing with error-prone depth data specially from stereo cameras remains a serious concern. Furthermore, the underlying core principle of weighted volumetric integration using Equations 2.9 and 2.10 remained unchanged over past two decades. Since the original concept of volumetric integration was designed to handle range images with minimal surface noise, the resulting frameworks struggles to accommodate depth images with high depth noise such as from stereo cameras etc. This provided the main motivation for the proposed research to handle depth noise using total variation based filtering in a recursive manner.

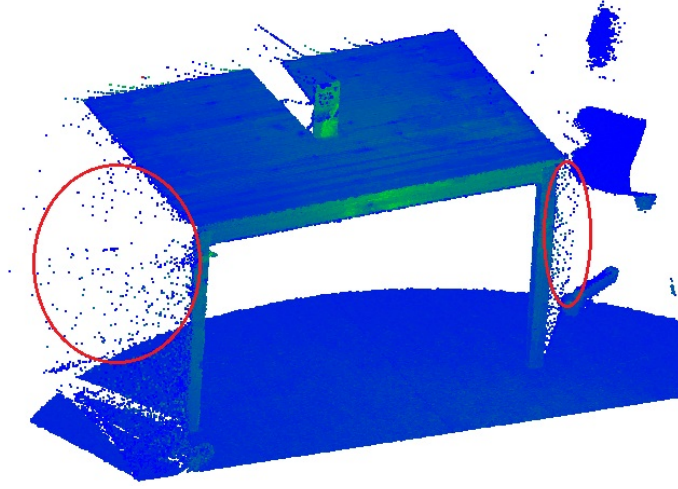


Figure 2.25: Typical point cloud with highlighted depth outliers

2.5 Depth Outliers removal

Incorrect estimations of depths, in general depth outliers, are a common phenomena and almost all 3D scanners are prone to this challenging problem. Laser based 3D scanners such as LiDAR produce depth outliers when the surface on the object of interest exhibits reflective properties. Similarly, active depth sensors which use pattern projection and detection such as Kinect produce incorrect depth measurements when the surface of foreground meets background, similar depth deformities have been identified when the depth image is subjected to over-smoothing by a depth filter. All aforementioned depth outliers pose serious problems for both camera pose estimation and 3D reconstruction. Figure 2.25 shows a typical 3D point cloud having depth outliers at edges of the table where the sensor detected two surfaces having different height profiles.

Depending upon the spatial proximity of the depth outlier with respect to the actual surface, an outlier can be classified as sparse, isolated or non-isolated outliers. Rusu et al. (2008) proposed a simplified Statistical Outliers Removal (SOR) method to target sparse and isolated outliers which calculates the distance of each point along its K neighbours in the first pass. Mean μ and standard deviation σ of accumulated distances are then calculated to determine appropriate distance threshold using following formula:

$$Threshold = \mu + \alpha * \sigma \quad (2.12)$$

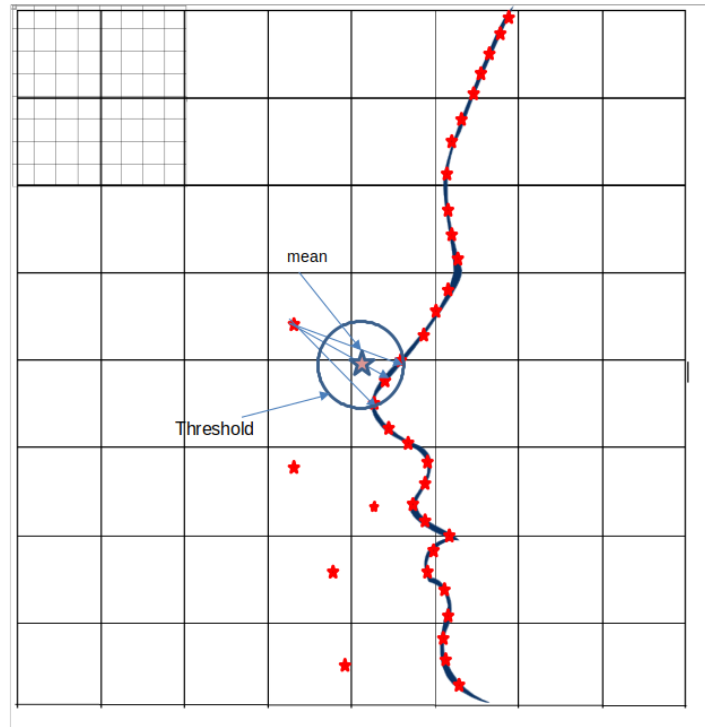


Figure 2.26: Statistical measures to detect outliers

where α is a user defined parameter to control overall distance threshold. This process is illustrated in Figure 2.26. In the second pass, all 3D points are classified as either inliers or outliers depending upon the distance threshold. Although this SOR method is highly effective in removing both sparse and isolated outliers, the first pass of the SOR method uses time consuming memory lookups to access neighbouring 3D points. This time consuming profile of the process poses serious concerns when using this scheme with real-time 3D reconstruction system. For instance, processing a typical 640 x 480 pixel depth image captured from Kinect RGB-D camera using SOR filter can take approximately 650 milliseconds to detect outliers using a SOR filter.

Wang and Feng (2015) proposed a majority voting based algorithm to target all types of outliers. Such voting based algorithm removes outliers with precision at the cost of high computational complexity. Techniques having a computationally expensive profile and lacking real-time processing such as Wang and Feng (2015) and Zhang et al. (2016) are not considered for the quantitative comparison with the proposed research.

2.6 Key Considerations

In order to summarize the literature review and establish a potential baseline, all aforementioned shape representation and Incremental 3D fusion techniques are evaluated separately with four evaluation criteria, which are:

1. **Generality:** Ability of the technique to reconstruct both complex and simple shapes.
2. **Robustness:** Ability to handle strong noise and outliers.
3. **Computation Speed:** Processing time and computational complexity of the technique.
4. **Accuracy:** Overall accuracy of the technique.

Table 2.1 summarizes reviewed shape reconstruction methods in the light of evaluation criteria, the plus sign indicates whether a particular technique is adequately fulfilling a certain criterion. These findings have been derived mainly from the literature, publicly available source code and in certain cases by direct communication with authors. In some cases, computation speed metric is evaluated with the help of the open-source tool *Meshlab* developed by Cignoni et al. (2008) which contains fast implementations of various reviewed algorithms.

All reviewed techniques which do not consider prior information were found incapable of handling either noisy or sparse samples, with the one exception of SDF by Newcombe et al. (2011) which uses a stochastic convergence property to reduce noise at the expense of very high storage requirements.

Techniques which use **Regularity Priors** are designed specifically for the targeted application such as vehicles and building facades, however this lack of generality constraints the application domain.

Both **Local** and **Global** smoothness priors were observed to produce accurate shape approximation and reconstruction, thus a large number of applications can integrate smoothness information to increase the quality of the 3D reconstruction. Local methods were found to produce faster parallel computations, however they are less suitable for high-noise or sparse 3D samples. Here, global methods such as *Poisson* and SSDF by Kazhdan and Hoppe (2013) and Calakli and Taubin (2011) respectively significantly outperformed and produced comparatively

Table 2.1: State-of-the-art 3D shape reconstruction approaches

Technique	1: Generality	2: Robustness	3: Speed	4: Accuracy
No Priors				
α shapes - Edelsbrunner and Mücke (1994)			+++	+
BPA - Bernardini et al. (1999)	+		+++	+
SDF - Newcombe et al. (2011)	+++	+	++	++
Regularity Priors				
Clustering - Pauly et al. (2008)		+++		+
Subspace tension - Berner et al. (2011)	+	+++	+	+
Learning Clusters - Yingze Bao et al. (2013)		+++		+
Local Smoothness Priors				
MLS - Alexa et al. (2001)	++		+++	++
Point blending - Kolluri (2008)	++		+++	++
APSS - Öztireli et al. (2009)	++		+++	++
MPU - Ohtake et al. (2005)	++	+	+++	++
Global Smoothness Priors				
RBF - Carr et al. (2001)	++	+	+	+
Graph Cut - Hornung and Kobbelt (2006)	++	+		++
Fourier - Kazhdan et al. (2005)	++	+	++	++
Wavelet - Manson et al. (2008)	++	+	++	+
Poisson - Kazhdan and Hoppe (2013)	+	++	++	++
SSDF - Calakli and Taubin (2011)	+	++	++	++

accurate 3D models, however the requirement and assumption of closes surfaces reduced the generality of these approaches.

Table 2.2 provides evaluation insight for various state-of-the-art 3D fusion frameworks. Since all presented 3D reconstruction frameworks are derived from volumetric integration by Curless and Levoy (1996), the robustness metric value is more or less the same. However differences in computation speed and accuracy are directly dependent upon external factors such as the camera localization algorithm, scale of reconstruct etc. We also found that **point based 3D fusion** techniques have more generic profile and support dynamics of environment however lack behind in handling high depth noise or sparse samples by stereo sensors.

Table 2.2: Incremental 3D Fusion frameworks

Framework	1: Generality	2: Robustness	3: Speed	4: Accuracy
Volumetric 3D Fusion				
InfiniTAM	++	++	+++	+++
FastFusion	++	++	++	+
KinectFusion		++	++	++
Voxel hashing	+	++		+
MonoFusion	+	++		++
Real-time vol rec	++	+		+
Point Based 3D Fusion				
Point-based Fusion	++	+	+	+
ElasticFusion	++	+	++	+

2.7 Summary

This chapter reviewed both legacy and state-of-the-art shape representation techniques in order to determine a suitable 3D reconstruction candidate with incremental fusion capabilities from error-prone 3D samples. We found that although simplexes representations is designed to utilize modern rendering frameworks, however their inability to handle error-prone 3D sample data makes these techniques unsuitable for real-time incremental reconstruction. Parametric representation methods were found suffering greatly due to their complex computational profile. Surface splatting techniques were found to accommodate dynamic changes in environment, however a large number of depth samples are required to reduce noise affects.

This constraint directly affect camera movements and makes surface splatting infeasible for low-frame rate depth sensors such as IPS (Grießbach et al. (2014)) which captures 10 frames per second. Implicit representations, more specifically volumetric 3D fusion methods produced promising noise handling behaviour, however the stochastic convergence property for noise removal depends directly on the quality of the depth measurements.

Despite gradual noise removal properties of the volumetric 3D fusion, handling of depth noise in 3D samples remains a serious concern and hence specialized image based edge aware depth smoothing schemes were reviewed briefly and found that total variation based filtering based depth map smoothing filters produced promising results. Depth outliers removal techniques were briefly discussed and found that statistical methods which use spatial proximity information effectively to reduce outliers, however high memory access time in underlying mechanisms makes such methods unsuitable for real time.

Chapter 3

Methodology

This chapter presents standard analysis practices and performance benchmark criteria adapted to evaluate the proposed contributions. In addition, technical requirements along with various datasets are briefly described to establish a minimal working environment and to reproduce findings. Initially, the structure of proposed framework is introduced which presents a high level understanding of the data transformation at each process. Secondly, various datasets along with standard quantitative and qualitative measures used for evaluation and validations are outlined. Finally, technical requirements such as the input data type, computational resources, development environment and software required to visualize and evaluate output are briefly described.

3.1 Framework Structure

The proposed framework in this thesis refers to a systematic 3D reconstruction pipeline which employs proposed contributions for high quality models from input depth and color data. The framework is designed to exhibit generic traits in terms of input data, which can be acquired from active RGB-D cameras, passive stereo based depth scanners or 3D laser scanning systems. Usually, laser based 3D scanners produce relatively accurate depth information, however main motivation is to process error-prone depth data captured from a low-cost active or passive sensor, which would enable mobile robots to perceive accurate 3D information in a cost-effective manner.

The overall structure of the framework is illustrated in Figure 3.1 which highlights the data

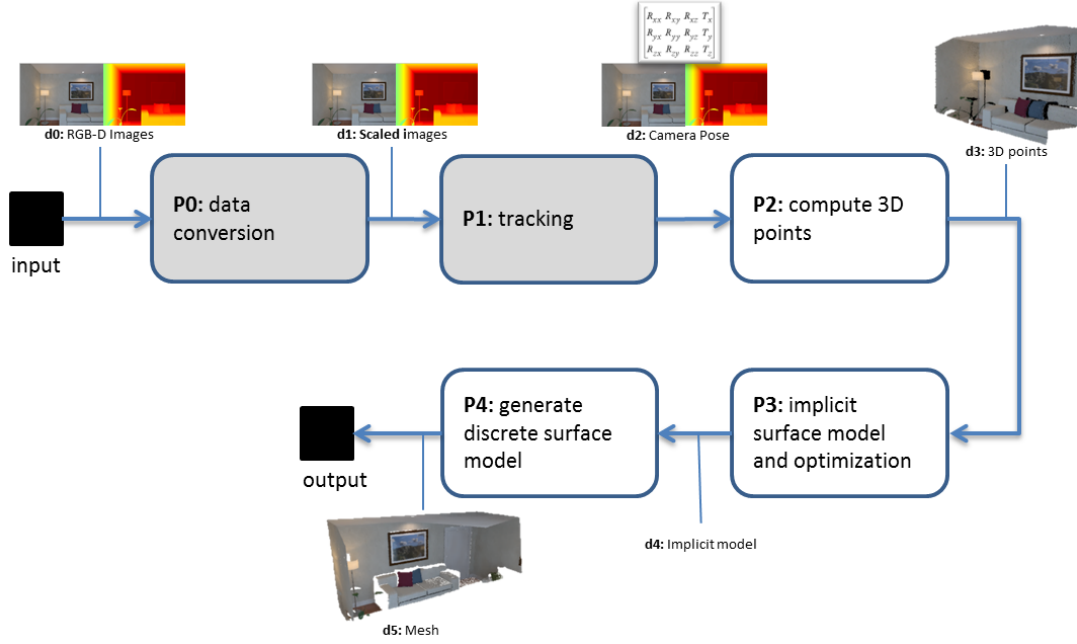


Figure 3.1: Framework applied on RGB-D image stream

transformation by each individual process for RGB-D data. Processes **P0** and **P1** are shown in shaded-gray as they are not part of the presented framework. In processing unit **P0** an appropriate depth conversion is employed to convert depth images in to a standard format. State-of-the-art localization methods such as ORB-SLAM2 (Mur-Artal and Tardós (2015)) or ICPCUDA (Whelan (2018)) can be employed for ego-motion sensor tracking in **P1**. Depending upon the type of 3D information given as input, one or more processes of the framework are not utilized. For example, in case of using the stereo based IPS sensor¹, processes **P0** and **P1** are not applied since IPS inherently performs stereo matching (i.e. depth conversion) and ego-motion localization and resulting depth, color and camera pose are fed directly to **P2**.

Assuming that an active RGB-D sensor such as Kinect RGB-D camera is used as an input, the first process **P0** converts each depth pixel $d_i(row, col)$ from i^{th} time-stamped raw depth image to an appropriate distance using a standard non-linear function (as suggested by OpenKinect.org (2018)):

$$D_i(row, col) = \frac{1}{d_i(row, col) - 0.00307 + 3.3309} \quad (3.1)$$

¹Integrated Positioning System (IPS) is a stereo based depth estimation system designed to assist in 3D navigation and reconstruction.

where $D_i(row, col)$ is a distance of a particular depth sample from the camera origin in millimetres (OpenKinect.org (2018)). The resulting depth image D_i and the pre-registered color image C_i are processed in **P1** which localizes the sensor position in world coordinates with the help of an ego-motion estimation. Process **P2** uses the camera pose information $pose_i$ and the intrinsic matrix K to register each valid depth pixel $D_i(row, col)$ in a world coordinate system. Process **P3** encapsulates the core method for producing an accurate 3D shape reconstruction, which contains three key contributions: 1) implicit shape approximation, 2) robust statistical outlier detection and 3) integration of *a priori* information to reduce noise inherently. Since the framework is designed to work with streams of images, each new input instance invoke **P0** \rightarrow **P3** in an iterative fashion and the implicit model **d4** is updated incrementally. Finally, **P4** uses a standard marching cube algorithm (see Lorensen and Cline (1987)) to process the implicit model and to produce globally consistent 3D models in the form of meshes **d5** as output.

3.2 Validation and Evaluation

The actual implementation of the proposed fusion framework is carried out in C++ programming language which is preferred among others due to highly extendable functionality with the help of publicly available libraries and its real-time processing profile. In order to validate various aspects of the implementation, *unit-tests* have been employed to ensure that low-level optimization tasks such as matrix manipulations and image conversion are correct. The data-based evaluation tests have been employed to ensure that the overall shape reconstruction algorithm is accurate and competitive.

3.2.1 Evaluation Framework

The accuracy of the reconstructed model and the processing time taken by the fusion framework are considered as standard performance metrics for shape reconstruction as suggested by Strecha et al. (2008). The comparison of the processing time among two functionally identical fusion frameworks is a straightforward process, however various development traits such as use of GPU computing or limiting the memory allocation for the abounded reconstruction provides a biased favour to a particular technique. Henceforth, state-of-the-art frameworks along-with

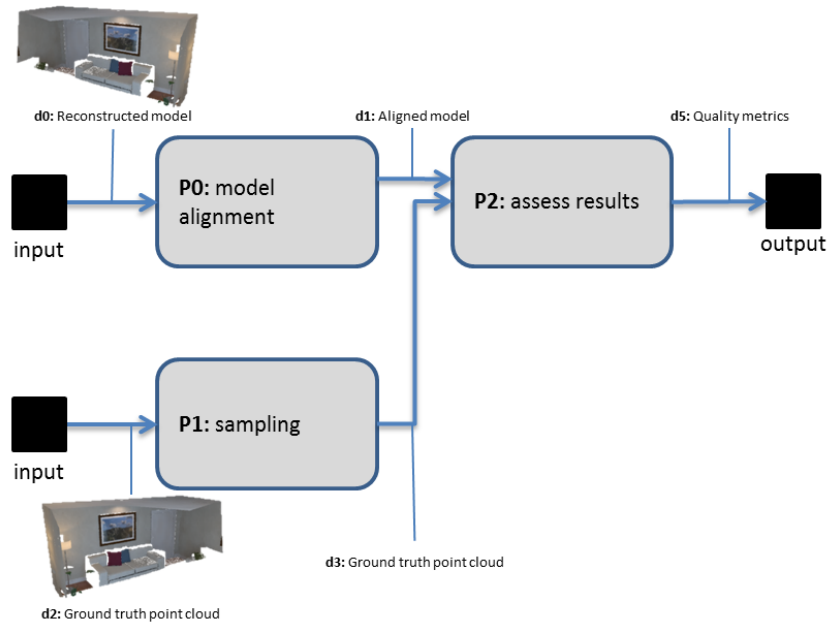


Figure 3.2: The process of acquiring quantitative measures among reconstructed and ground truth 3D models

the proposed framework are subjected to a variety of evaluation aspects such as accuracy, functionality and processing time to establish a comprehensive profile of each technique.

The accuracy of the reconstructed model naturally involves comparing a resulting model against an *a priori* known result, called *ground truth model*. Unfortunately, acquiring ground truth models for large-scale 3D reconstructions is a tedious and careful process involving laser guided depth data collection in the form of a point cloud followed by high-quality mesh generation. Although careful measurements combined with a time-consuming shape approximation process generates a high-quality 3D model, the resulting estimated ground truth model is still expected to contain approximation errors. Ground truth models for large-scale environments are usually not available. Synthetic environments and relative models are therefore preferred since they provide very accurate measures.

The evaluation framework is a process pipe-line which produces quantitative measurements given approximated 3D models and ground truth. The assessment framework is summarized in figure 3.2 and three underlying processes perform the following tasks:

- **P0:** Given a reconstructed 3D model in the form of a 3D mesh, appropriate *Scaling*, *Rotation* and *Translation* operations are applied to make the given model consistent with the ground truth model.

- **P1:** The Ground truth model is sampled with at least 10^6 points to ensure that surface contours are captured regardless of the shape and size of the model in the sampled 3D point cloud **d3**.
- **P2:** Each point in **d3** is registered to a closest polygon in the reconstructed 3D model and a perpendicular distance is recorded. This distance when averaged, provides five quantitative measures (mean, median, standard deviation, min and max distance) and are considered as standard evaluation criteria as suggested by Handa et al. (2014). This assessment is carried in *Cloud-Compare* (Girardeau-Montaut (2015)) which is freely available open-source software.

3.2.2 Performance Metrics

In order to inspect the performance of the proposed fusion framework, a numerical evaluation is employed to collect appropriate performance metrics which allow meaningful information and conclusions to be extracted from the shape approximation process. In principal, the Euclidean distance is used between the reconstructed surface and ground truth sampled data as suggested by Berger et al. (2013). For each sample, an absolute distance from **d3** to the closest polygon from **d1** is computed in process **P2**. This error measurement is used as a primary source to extract more significant statistical measures such as error histograms and cumulative error distributions. A similar approach has been applied to evaluate multi-view stereo reconstruction techniques by Strecha et al. (2008). Following performance metrics are applied:

- **Visual inspection:** Real-time 3D reconstruction results are visualized using the OpenGL renderer. However once the reconstruction is completed and the model is stored in memory, intensive inspection is carried out and appropriate features of the model are captured using *Meshlab*.
- **Absolute surface error:** Absolute error measurements collected from the process **P2** are projected onto the reconstructed model in color coded error maps (also referred as *heat-maps*) to visualize spatial deviations by the reconstruction process. This process is shown in Figure 3.3 where the reconstructed model is compared against sampled ground truth 3D points to calculate basic quantitative measures and absolute surface errors.

Common statistical visualization standard tools (see Tufte (1990)) such as median and variance, error histograms (Figure 3.4.a) and cumulative error distribution plots (Figure 3.4.b) are generated.

- **Mean, median and variance:** After computing the absolute surface errors between **d1** and **d3**, **P2** summarizes the error distribution information in mean and median values. Median Distance Error (MDE) is calculated by sorting all non-zero distances in ascending order and taking the central sample from the sorted list. In some special cases, MDE is selected over mean value as a primary error descriptor, since it is robust to large outliers in a particular spatial location. In cases where relatively large absolute surface occur at one particular position, MDE metric is less likely to reflect drastic changes. Furthermore, mean and variance error metrics are computed to further analyse distribution.
- **Statistical measures:** Probability density functions are considered as a de-facto metrics to evaluate the nature of a random process. Since the computed error-distribution is discrete in nature, histograms are used to represent how well the reconstructed shape is aligned to the ground truth model. The peak of the histogram represents most frequent errors made by the reconstruction process, however the comparison of two histograms is a non-intuitive and relatively difficult process. This is addressed by calculating a cumulative distribution from each histogram which can then be used to analyse in an intuitive manner. Figure 3.4 illustrates the relation between a histogram and a cumulative error distribution. An optimally superior method would rise sharply to the 100% while a low-quality technique will produce a gradually increasing curve.
- **Runtime:** The time taken by an algorithm to accomplish a computational task is usually measured in milliseconds using the CPU clock. This empirical metric is used to evaluate the processing time taken by the implemented algorithm. All experiments have been performed on a desktop computer having following specifications:

- Intel Core i7-4790
- Nvidia Quadro K620²

²Used only to evaluate InfiniTAM by Kähler et al. (2015).

- 8 GB RAM
- Windows 7 (64-bit) and Linux 14.04 operating system.

Using a high performance computer will accordingly enhance the runtime performance metric.

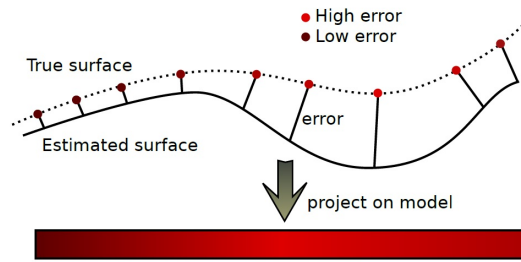


Figure 3.3: Process of calculating error distance between sampled ground truth 3D points and the reconstructed model and resulting an absolute surface errors in a color coded error map.

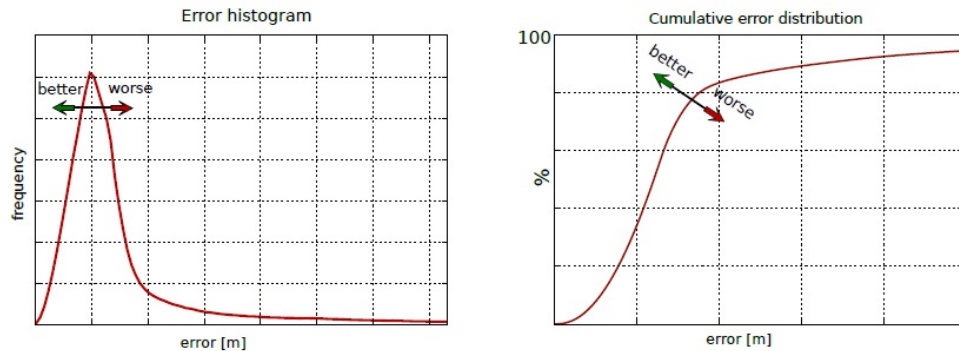


Figure 3.4: Typical error histogram (left) and cumulative error distribution (right).

3.3 Datasets

3.3.1 Synthetic Piecewise Function

Main motivation behind this research is to perform a reliable 3D implicit fusion while handling effects of depth noise. It is therefore beneficial to critically evaluate the performance initially on a synthetic piecewise 2D function. This initial testing phase allows an effortless visualization

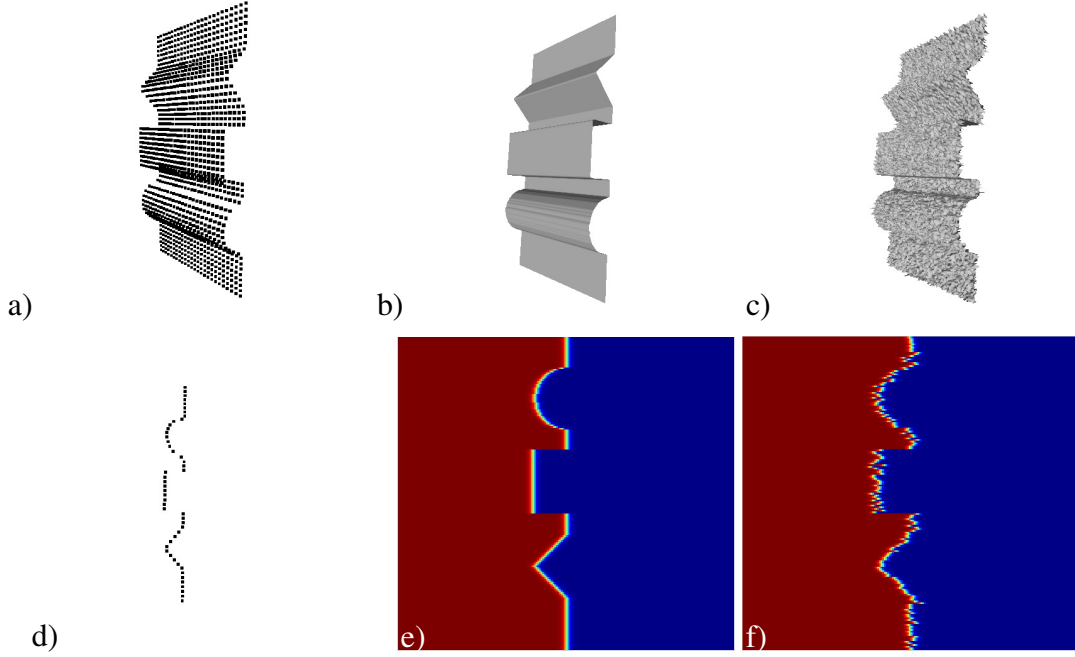


Figure 3.5: Synthetic 3D and 2D function represented as point cloud followed by respective implicit representation with and without added depth noise.

which enables rapid prototyping of the solution development. A replicated version of a 2D piecewise function in the third dimension is used to validate a response of the proposed framework with additive noise in a 3D environment. Figure 3.5 illustrates both 2D and 3D functions (represented with points) with respective volumetric implicit surfaces with and without additive depth noise.

Main benefit of using synthetic functions is that parameters such as additive noise and scale of reconstruction can easily be controlled. The model contains gradual curves, planar areas and sharp edges so that the performance of the proposed framework can be evaluated in detail.

3.3.2 Synthetic 3D Complex Environment

There exist a variety of publicly available synthetic 3D models which can be used to generate synthesized camera movements and characteristics, however the standard RGB-D dataset by ICL-NUIM (Handa et al. (2014)) was selected to obtain un-biased results. This dataset is specifically selected to reflect a broad range of requirements often needed by the 3D fusion framework. Thus, an algorithm performing well on the simulated environment is expected to perform well in a realistic datasets obtained from either active or passive depth sensors, even if

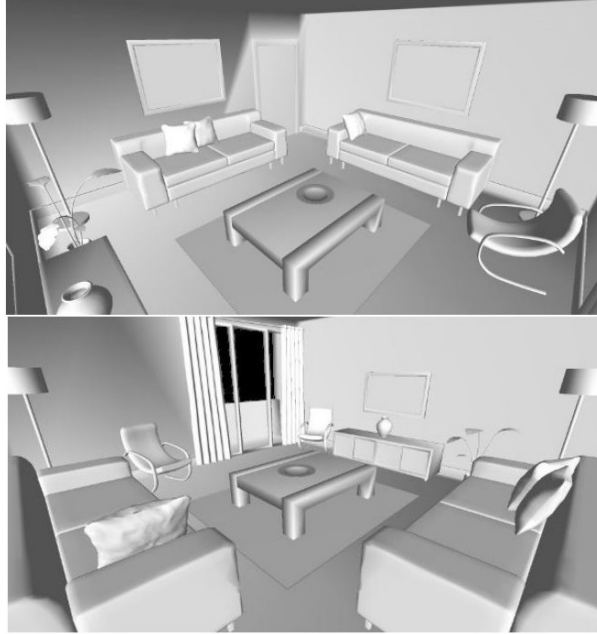


Figure 3.6: The interior of a synthetic living room scene (color removed to highlight geometry)

the ground truth model is not present for quantitative evaluation.

The dataset contains four distinct trajectories of the *living-room* environment simulating repetitive loop closure, fast and slow moving camera movement along with two different sets of depth image streams which simulate noisy depth measurements from a Kinect-like RGB-D camera and clean depth images. The simulated *living-room* environment consists of challenging micro and macro objects having planar, sharp and gradual curvatures. Figure 3.6 shows two rendered views highlighting the challenging complex geometry of the scene. Both clean and depth images which are corrupted with noise were used to evaluate the proposed framework. Results are presented in Section 6.

3.3.3 Realistic 3D Complex Environment

One of the main motivation behind this research is to develop a generic fusion framework with controlled regularization parameters to accommodate variety of depth sensing systems. Therefore, three distinct realistic benchmarking datasets *Comprehensive RGB-D Benchmark for SLAM* (CoRBS) (Wasenmüller et al. (2016)), *KITTI* vision benchmark suite (Geiger et al. (2013)) and *IPS* dataset (Grießbach et al. (2014)) were selected to demonstrate the flexibility of the proposed framework. Unfortunately, ground truth models for *KITTI* and *IPS* datasets

are not available, therefore screenshots of the reconstructed model are provided for qualitative comparison in Section 6. Characteristics of each datasets are:

- **IPS-Dataset:** The dataset has been captured from a passive depth acquisition system which uses stereo cameras to capture time-stamped images. Registered images are processed with the *Semi Global Matching* (SGM) algorithm (Hirschmuller (2005)). Ego-motion information acquired from built-in *Inertial Measurement Unit* (IMU) are fused with a visualSLAM algorithm to produce a globally consistent camera pose information. IPS is capable of producing approx. 10 instances containing depth, color and camera tracking information. Figure 3.7 shows sample depth and color images acquired from the IPS depth sensor system. IPS is capable of transmitting real-time depth sensing via TCP/IP communication for real-time processing, however to facilitate a repetitive evaluation process, trajectories were recorded and processed with the fusion framework Griebbach et al. (2014).

Although IPS provides high-accuracy localization information with the help of multi-sensor fusion and bundle-adjustment, estimated depth measurements suffer from strong estimation noise and outliers caused by reflections, varying illuminations and fast camera movements. Three scenes *mine*, *corridor1* and *corridor2* have been recorded for benchmarking that represent large-scale scenes with challenging environmental conditions. *Corridor1* and *corridor2* scenes demonstrate non-textured surfaces, occlusions and difficult lighting conditions which further contributes in depth noise and outliers, while *mine* scene highlights low illumination conditions with a fast moving camera.

- **CoRBS:** The dataset contains four distinct scenes captured by latest Kinect v2 RGB-D sensor. Unlike state-of-the-art RGB-D datasets which only provide a camera trajectory as the ground truth, CoRBS provides high-quality 3D ground truth models captured by projecting light patterns onto a surface (see Figure 3.8). Camera trajectory information is captured by an external motion capturing system with sub-millimeter precision. The resulting trajectories and ground truth models are aligned to a global coordinate system to further simplify the evaluation process. Thus CoRBS is selected to demonstrate work of proposed framework using novel RGB-D depth sensor system.

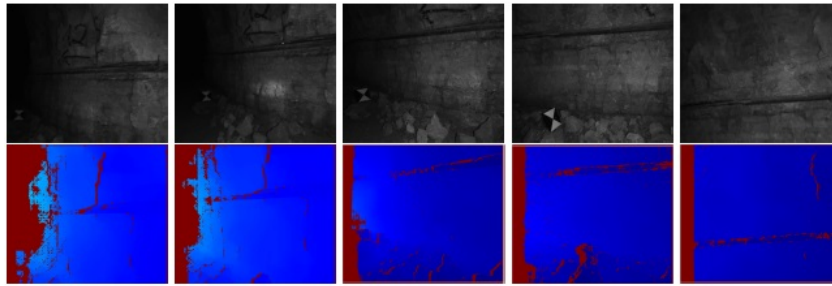


Figure 3.7: Arbitrary sampled instances of registered color and depth images from IPS sensor

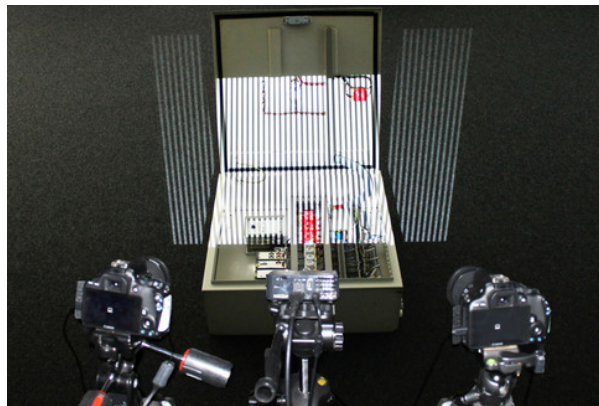


Figure 3.8: Light pattern projection based range image acquisition using two cameras (Wasenmüller et al. (2016)).

- KITTI vision benchmark suit:** The dataset is captured from a standard vehicles mounted with a 360° rotating laser scanner, two pairs of stereo cameras, and IMU module capable of capturing 10 samples per second to facilitate benchmarking series of computer vision research problems such as stereo matching, optical flow, ego-motion estimation and object tracking. Figure 3.9 illustrates the sensing equipment mounted on *AnnieWAY*. Localization information in the form of sensor movement for trajectories are also provided as the ground truth for testing visualSLAM algorithms. Since main focus of the research is to facilitate a 3D shape reconstruction, we employ only stereo images and laser scanner data with estimated camera pose from state-of-the-art visualSLAM algorithm to demonstrate a real-time 3D reconstruction scenario with the proposed fusion framework.



Figure 3.9: AnnieWAY with mounted multi-sensor system set-up.

3.4 3D Sensors

Mobile robotic applications such as automated drones, unmanned vehicles etc are usually constrained to provide limited amount of power to various sensors. Therefore, such mobile robots are generally equipped with low-power consuming sensors such as RGB-D cameras connected to a mobile computing device which transmits the captured depth and color image streams to the high-performance computer. Provided that enough network bandwidth is available to send the image stream, the processing computer can either apply an on-line 3D fusion approach or store the image stream for off-line processing.

Fortunately, autonomous vehicles are capable of providing sufficient power and maneuverability, this allows them to utilize multiple 3D sensors and high performance computing on the go. Although autonomous vehicles are expected to navigate in a variety of lighting and environmental conditions, a multi-sensor set-up ensures that not all sensors are affected by a particular lighting condition.

Active depth sensor systems especially Kinect, ASUS Xtion Pro and Kinect v2 are designed specifically for in-door environments where lighting conditions are either regulated or are kept deterministic, however their efficiency to perceive the environment decrease drastically with certain lighting conditions. Furthermore, pattern based active depth sensors are prone to highly illuminated surfaces, while time-of-flight sensors are prone to introduce surface estimation errors on darker surfaces. Although active depth sensors provide high frame-rates (i.e. approx

30fps), their incapability to perceive distant objects restricts their usability in fast moving outdoor environments.

Passive depth sensors which utilize stereo images and disparity estimations provide long range sensing capability to mobile robots. In principal, the quality of estimated depth's in stereo matching algorithms rely heavily on the base line textures of surfaces. Since depth estimation is a computationally expensive process, special *System-on-Chip* (SoC) or GPU based portable computing nodes are required to facilitate real-time depth estimation.






Laser based 3D scanners are a more suitable choice for autonomous vehicles since they provide high-quality depth measurements without interpretation. However these scanners usually provide sparse depth measurements which are not suitable for 3D reconstruction since each measurement has a different orientation (i.e. an external projection is required to interpret these measurements into a depth map). Furthermore, these laser scanners are high cost and prone to measurement errors in-case two similar sensors are scanning the environment.

Table 3.1 summarizes aforementioned characteristics of depth scanners and suggests a suitable application environment.

3.5 Summary

This chapter presented a research methodology and standard practices used to evaluate and analyze performance aspects of the proposed research. The generalized structure of fusion framework is introduced which subdivides overall processing tasks into a process pipe-line, this modular design allows effortless validation and testing of the data at different stages of the pipeline. Standard quantitative and qualitative evaluation metrics have been introduced. Finally, both synthetic and realistic datasets have been briefly described which highlights applications of the proposed research in a real-time 3D reconstruction environment. The following chapter will introduce theoretical aspects of the proposed research contributions.

Table 3.1: 3D depth sensors and their respective characteristics

Sensing Technology	Sensor	Image	Pros	Cons
Pattern projection	Kinect		<ul style="list-style-type: none"> - Low cost - High frame rate 	<ul style="list-style-type: none"> - Short range - Sensitive to high illumination
	ASUS Xtion Pro			
Time of flight	Kinect v2		<ul style="list-style-type: none"> - Low noise - High frame rate 	<ul style="list-style-type: none"> - Short range - Inability to detect dark color objects
Stereo	IPS		<ul style="list-style-type: none"> - long range - Suitable for outdoors - Produces camera trajectory 	<ul style="list-style-type: none"> - Low frame-rate - Inability to detect dark color objects
Laser	LiDAR		<ul style="list-style-type: none"> - long range - Suitable for outdoors 	<ul style="list-style-type: none"> - Sparse sensing - Very high cost - Heavy - Low frame-rate

Chapter 4

Fundamentals of Volumetric 3D Integration

Chapter 2 identified the effectiveness of an implicit representation for error-prone depth images for an incremental 3D integration and reconstruction. Furthermore, it was also found that employing smoothness prior information can be used to reduce noise artefacts. This chapter will provide in-depth analysis and theoretical background of the underlying volumetric fusion process to identify potential challenges of employing smoothness prior in integration process. Moreover, the core concept of the implicit fusion is analysed with the help of various error-prone synthetic SDF signals. Presented concepts of legacy volumetric integration are expected to serve as theoretical foundation for proposed contributions in Chapter 5.

Initially, implicit representations of volumetric integration are formally introduced and related properties are discussed. Secondly, incremental aspects of the dense 3D fusion are analysed by fusing simulated noisy depth signals. Finally, the rationale for using sparse or semi-dense voxel grids are presented which highlights trade-offs between computational complexity and quality of the reconstructed model.

4.1 Signed Distance Function

The Signed Distance Function (SDF), also referred to as Distance Transform is a basic building block for visualizing and processing volumetric 3D data. In computer graphics community,

SDF is commonly used to accelerate the rendering process for high-quality 3D models (Hart (1996)), however state-of-the-art reconstruction frameworks employ SDF to integrate incremental updates from depth cameras. In principal, to render a SDF voxel-grid, a non-zero crossing of SDF along a viewing ray is considered as implicit surface (also referred to as *iso-surface*).

4.1.1 Definition

To elaborate the aforesaid definition in a formal construct, consider a mapping function

$$D(x) : \mathbb{R}^n \rightarrow \mathbb{R}, \quad (4.1)$$

which transforms n -dimensional space to a scalar value. Since our target application domain is 3D space we assume $n = 3$, however a 2D analogy is employed for illustrative purposes. Assuming an implicit surface of a circle having the radius $r = 5$ in τ units defined by

$$x^2 + y^2 = r^2, \quad (4.2)$$

Assuming that $g(x, y)$ denotes the euclidean distance from the origin, the implicit surface for such set-up can be defined by

$$g(x, y) - r = 0, \quad (4.3)$$

which ensures that each voxel at location $v = [x, y]^t$ contains a signed distance value from nearest implicit surface. In most cases, a linear truncation function

$$\hat{D}(x) = \max(\min(g(v) - r, D_{max}), D_{min}), \quad (4.4)$$

is applied to constrain SDF values in a particular range (i.e. $d_{max} \leq \hat{D}(x) \leq d_{min}$), the range is referred to as *support* in the upcoming text. Controlling the *support* of SDF is important parameter which plays a significant role in fusing SDF volumes. Therefore, a robust truncation function having properties of *generalized logistic function* is defined by

$$\hat{D}(x) = 1 - \left(\frac{2}{1 + e^{kd}} \right), \quad (4.5)$$

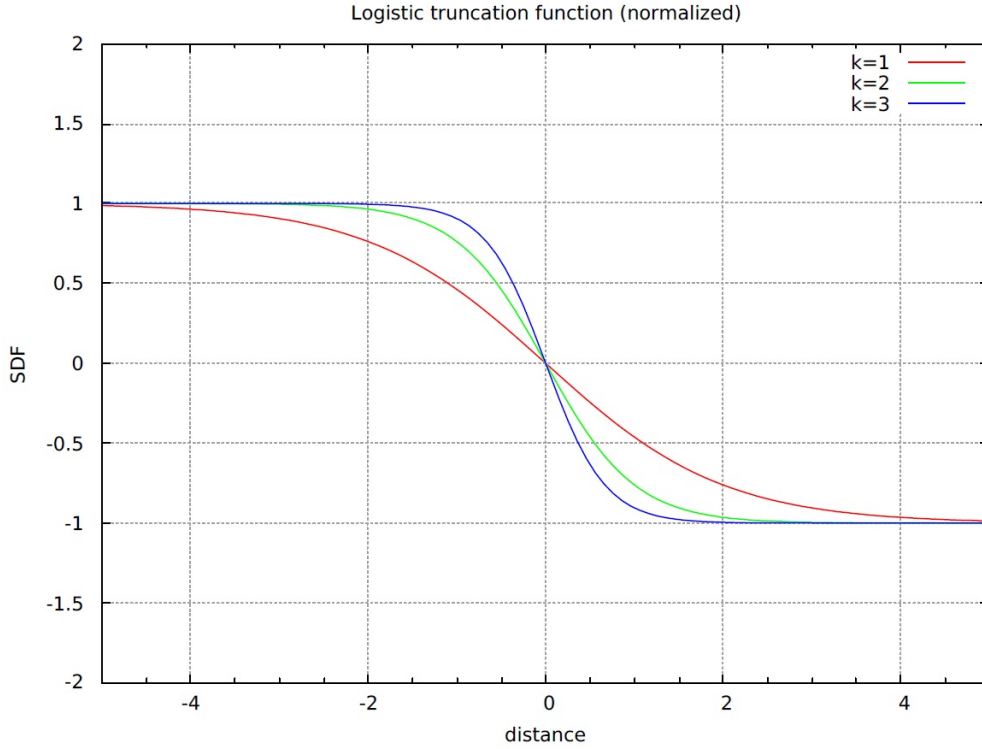


Figure 4.1: Truncated SDF function adapted from logistic function.

where *support* is controlled by varying parameter k and d denotes the depth information from camera origin, effects of varying parameter k are shown in Figure 4.1. To avoid unnecessary complexity, the upcoming text presumes that either a linear or truncated logistic function is employed, however the actual implementation of 3D reconstruction framework uses Equation 4.5 to compute TSDF values. Truncated SDF values which satisfy the range criteria can be used to achieve efficient memory utilization.

Figure 4.2 illustrates aforementioned properties of SDF from a circle having $r = 5$. Since the distance of each location $v = [x, y]^t$ is relative to the edge of circumference, therefore the positive and negative values represent outside and inside respectively while SDF value being equal to zero are on the edge of circle. When dealing with actual sensor data, finding local or global implicit function which satisfies the geometry of an acquired depth image is a computationally expensive task (discussed in Section 2.1.2). Hence an approximation of the SDF is assumed and briefly discussed in upcoming section.

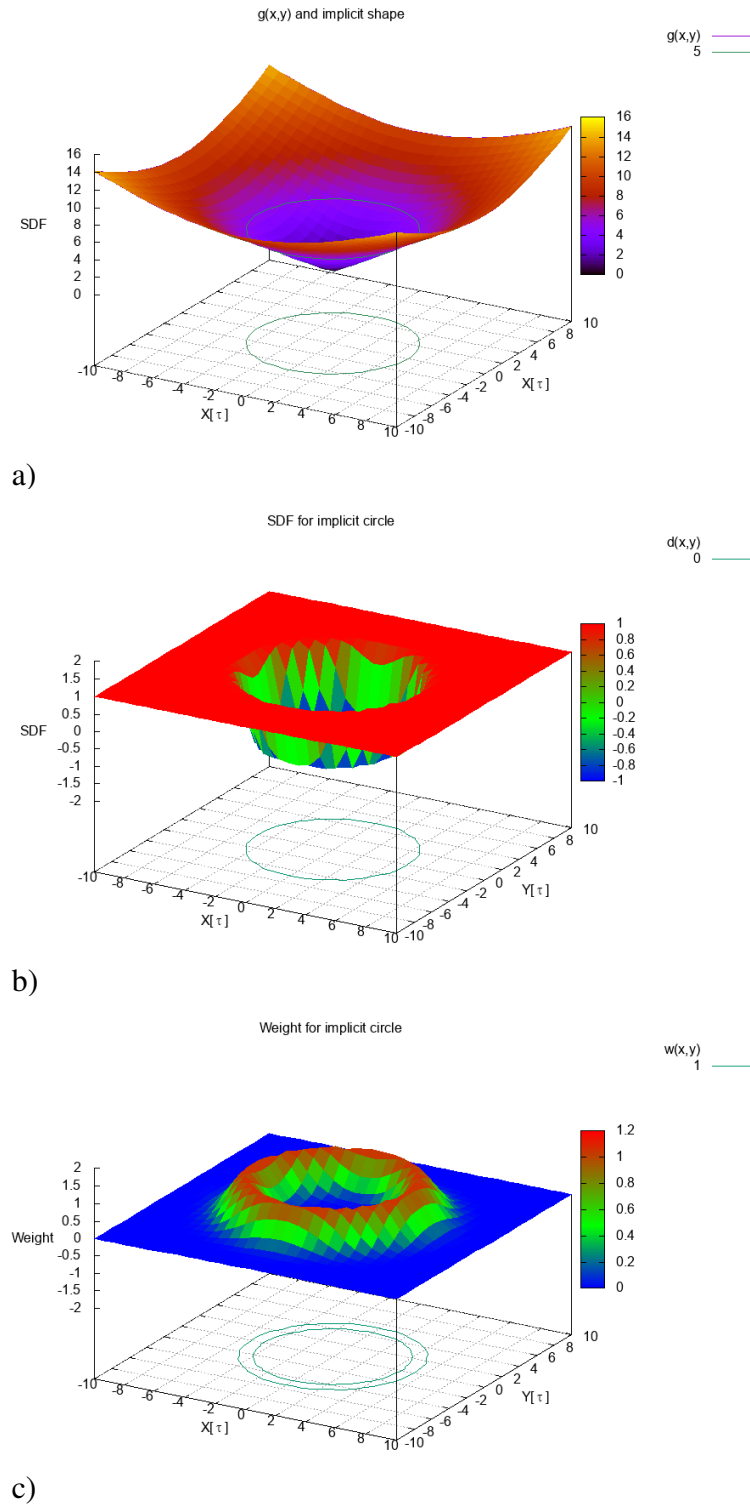


Figure 4.2: a) Plot of $g(x, y)$ from origin and green contour line shows all the points with distance equal to $r = 5$ in τ units from origin , b) Truncated implicit surface $\hat{D}(x, y)$ and c) Respective weighting function $W(x, y)$ with projected green contour lines highlighting the suspected surface.

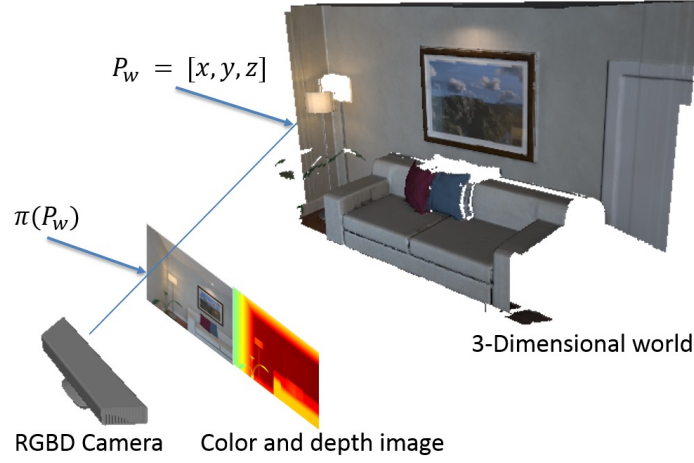


Figure 4.3: Projection of world information onto depth and color cameras.

4.2 SDF from depth images

In order to maintain generalization aspects of the research objective, we assume that streams of depth and color images are captured from 3D scanners. This assumption ensures that the proposed research framework is capable of processing input depth images irrespective of the input source. A standard format for depth and color image stream is suggested by Sturm et al. (2012) in which each image is registered and time-stamped for easy access. This arrangement of storing depth and color image stream is further supported by state-of-the-art visualSLAM algorithms (such as ORB-SLAM2 and RGBD-SLAM), therefore appropriate depth conversion must be employed to convert depth data from IPS or laser sensor.

Considering a set-up in which a pre-localized 3D scanner captures a depth and a color image (denoted by \mathbb{Z} and \mathbb{I} respectively) it is presumed that both intrinsic and extrinsic parameters of the 3D sensor are known where $f = (f_x, f_y)$ and $c = (c_x, c_y)$ are focal lengths and central point respectively. In principal, every physical point in the world coordinate system $P_w = [x_1, x_2, x_3]$ is projected onto \mathbb{Z} and \mathbb{I} using a perspective-projection function $\pi(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ formally defined by

$$\pi(P_w) = \begin{bmatrix} \frac{x_1}{x_3} f_x + c_x \\ \frac{x_2}{x_3} f_y + c_y \end{bmatrix} \quad (4.6)$$

where $\pi(P_w) = [u, v]^t$ and the process is illustrated in Figure 4.3.

In order to represent a depth image as a SDF, a discrete voxel grid of finite size having

spatial resolution τ is initialized. In most cases, τ determines the scale of reconstruction however there exist multi-scale variants (see Steinbruecker et al. (2014)) in which the scale of a particular spatial bounding box depends upon the proximity and the quality of the acquired depth image. For simplicity, we presume τ is a constant parameter selected before the fusion process and all spatial measurements such as the dimensionality of every voxel, focal lengths and coordinate system are converted accordingly.

For computational simplicity it is a common practice to presume that \mathbb{Z} represents a three-dimensional surface. Therefore, every voxel $v = [x_1, x_2, x_3]$ and the corresponding projection information $\mathbb{Z}(\pi(v))$ can be used to determine a signed distance value of each voxel using

$$D(v) = \mathbb{Z}(\pi(v)) - x_3, \quad (4.7)$$

where $D(v)$ denotes the SDF value of voxel v from the presumed surface.

Similar to implicit representation of a circle (see Section 4.1), Equation 4.7 produces positive, zero and negative values depending upon whether the centre of voxel is outside, at or inside the presumed iso-surface. The truncation function is then applied to constrain SDF values in the proximity of the iso-surface. Figure 4.4 shows a cross-section of a three-dimensional voxel-grid to highlight the SDF representation. Since typical 3D scanners can perceive environment information from one view, multiple depth images captured from different spatial locations (usually referred to as multi-view) are used to reconstruct a complete 3D model of the object and/or environment.

4.3 Effects of incremental 3D fusion

Although the core concepts of weighted incremental 3D fusion by Curless and Levoy (1996) are introduced in Section 2.4, this section is intended to analyze the effects of incremental 3D fusion when the framework is provided with noisy depth data. For illustrative purposes, considering a single ray is considered originating from camera centre towards the surface of object at 11τ units away. As stated earlier that 3D scanners are prone to introduce depth noise, it is assumed that the system collects two measurements of the suspected surface with added depth noise of form $\mathcal{N}(\mu, \sigma) = \mathcal{N}(0.0, 1.0)$ without moving the camera and/or object. The resulting error-prone depth measurements represented with TSDF and weighting function

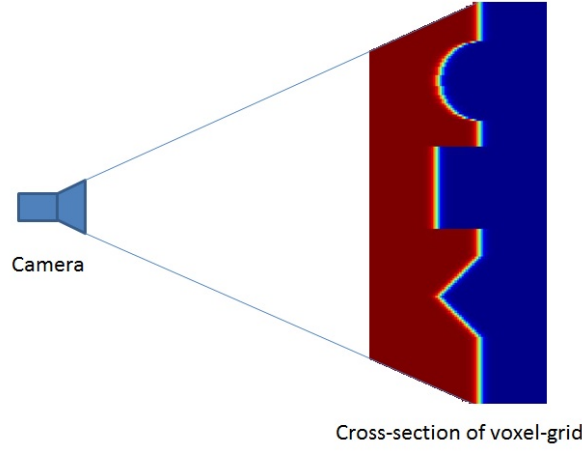


Figure 4.4: Cross-section of a voxel-grid with color coded SDF values.

$w(x)$ are shown in Figure 4.5.a and 4.5.b respectively.

The integration technique proposed by Curless and Levoy (1996) utilizes a weighted addition of SDF values using Equations 2.9 and 2.10. Figure 4.6.a and 4.6.b shows integrated TSDF and weighting function values respectively. In principal, the zero-crossing of the fused implicit SDF signal represents a better approximation of the actual surface than individual noisy depth samples. It is therefore expected that each incremental update of depth information reduces the estimation error between the iso-surface and the actual surface. Unfortunately, this statistical convergence property depends heavily on the number of depth samples and properties of added noise. Determining noise characteristics of every available depth sensor is unfortunately a tedious process, Nguyen et al. (2012) showed that depth noise from Kinect sensor can be estimated with Gaussian distribution. It is therefore presumed that depth measurements are corrupted with standard Gaussian noise. In practice however, properties of noise can be exploited by varying the weighting function of the 3D integration.

4.3.1 Relation of convergence with weights

In order to highlight the concept of convergence in 3D fusion with a varying degree of the depth noise, 200 instances of synthetic piecewise signal from Section 3.3.1 were generated and corrupted with additive Gaussian noise of the form $\mathcal{N}(\mu, \sigma) = \mathcal{N}(0.0, 5.0)$. As stated earlier, the integration process involves fusing multiple instances of SDF and weight values together using Equations 2.9 and 2.10. Once applied in an incremental fashion, the accumulated

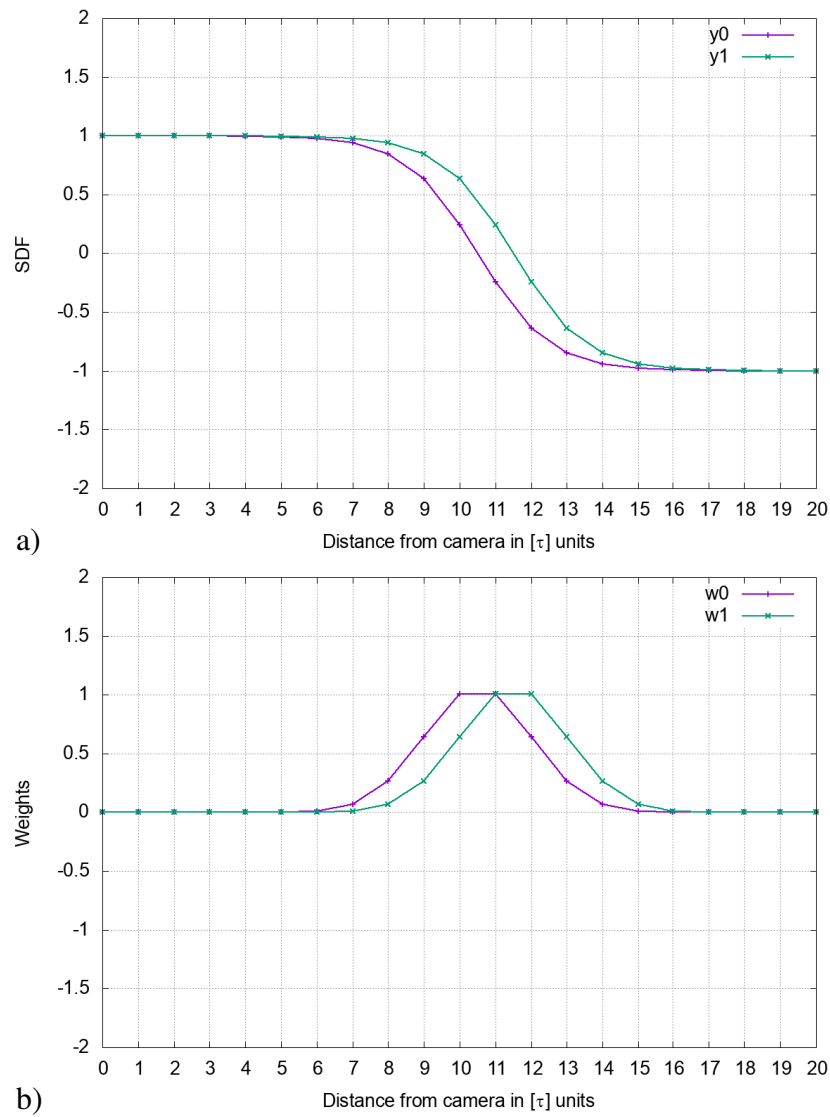


Figure 4.5: a) Error-prone depth measurements represented as one-dimensional TSDF function and b) Respective weight values generated using standard Gaussian function.

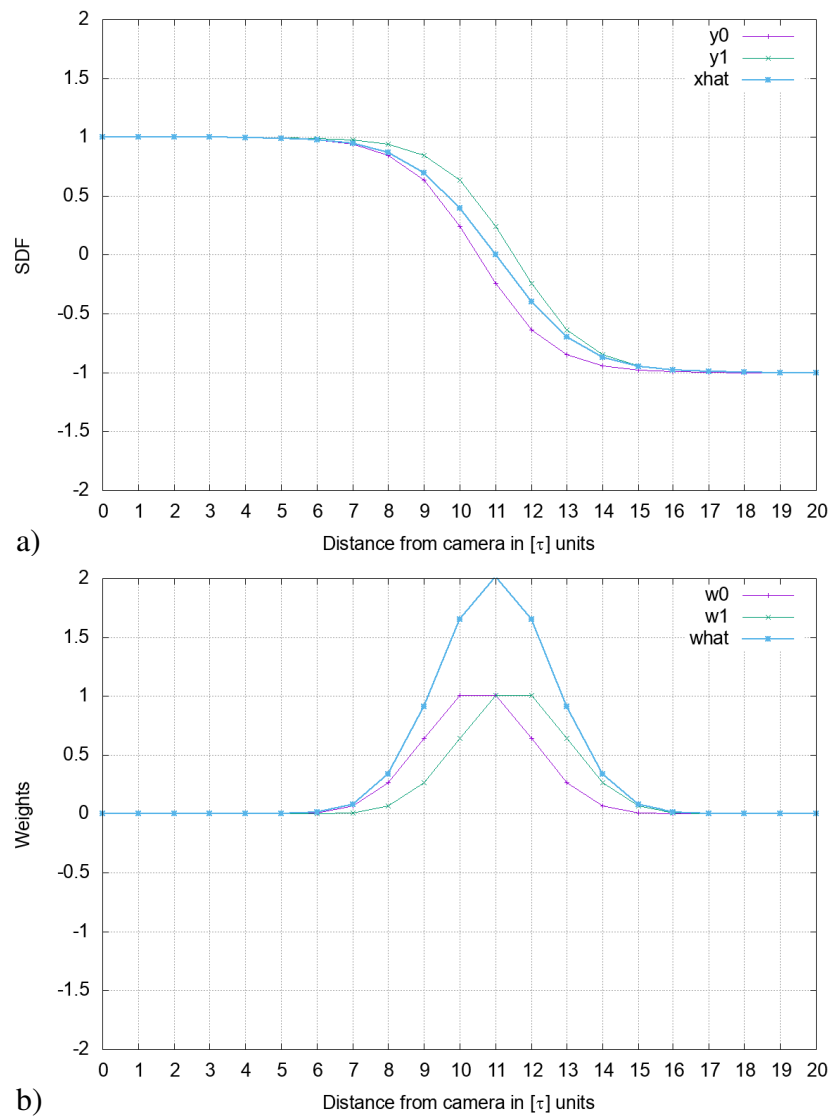


Figure 4.6: a) Fused TSDF function and b) Updated weighting function.

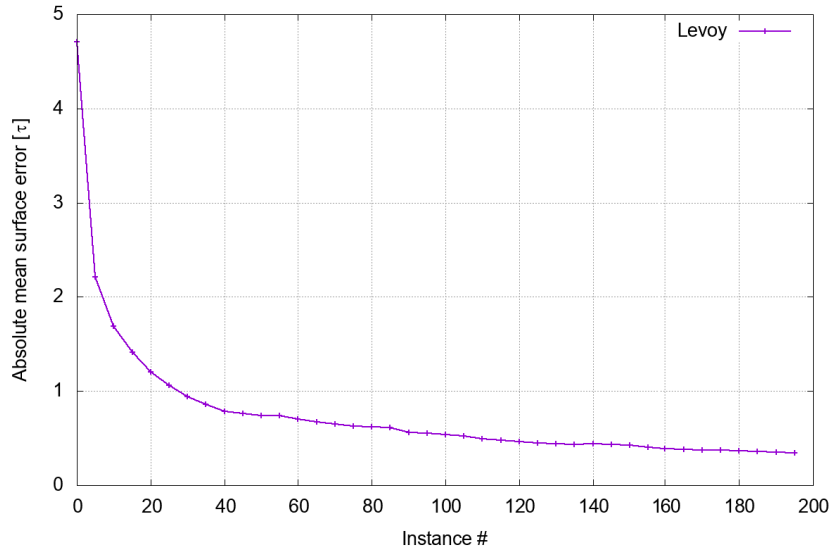


Figure 4.7: Mean absolute surface error convergence with incremental fusion.

weight values are expected to exhibit properties of a *normal* distribution in which the peak of distribution represents estimated implicit surface. Less erroneous samples are expected to produce a sharp peak in weight values and vice-versa.

In principal, the weight values are analogous to the *confidence* of the estimated implicit surface, where each incremental update increases the overall confidence. Quantitative analysis of this integration technique confirms the presence of *convergent* behaviour in terms of minimizing the absolute mean surface error, this phenomenon is shown in Figure 4.7.

Although the relation between weight values and error convergence is loosely proportional, selecting an appropriate weighting function for a specific depth sensor is a tedious and time consuming task which requires expert human interaction. In special cases where the surface of a perceived object is either non-rigid or non-stationary, the *confidence* value generates multiple zero-crossings in SDF values and result in inconsistent surfaces. Furthermore, the weight value for each voxel location is stored as a floating point element occupying *32-bit* of memory space. This memory in-efficient utilization further restricts the application domain of the 3D reconstruction to high-end processing devices.



Figure 4.8: Undesirable holes in reconstructed model from *ICL-LR2* trajectory.

4.4 Semi-dense voxel grid

Running-time analysis of the reconstruction framework is a significant performance metric, many implementations of volumetric fusion such as Izadi et al. (2011) and Steinbruecker et al. (2014) presume a spatial limitation of the observable environment and achieve efficient memory utilization by implementing a dense voxel-grid. Modern techniques utilize a truncation function to identify a proximity of each voxel near the expected iso-surface which allows large-scale reconstructions and efficient utilization of memory resources. Such truncation produces semi-dense spatial voxel locations which are eventually stored in a linear memory with the help of a hash function.

Processing a semi-dense voxel grid with SDF is a robust extension of traditional fusion, however limiting the number of voxels which satisfy the proximity criteria restricts the application domain and affect the quality of reconstructed models. In the case of voxel-block implementation, the resulting 3D model is prone to contain undesirable holes due to close proximity of the estimated iso-surface and the alignment of the particular voxel-block. These holes in the reconstructed model are indications of difficulties the rendering system has with finding the zero-crossing within each voxel-block from the viewing angle, Figure 4.8 illustrates this phenomenon in which casted rays cannot detect the iso-surface by checking zero-crossing in SDF values.

Another category of semi-dense voxel grid implementation uses the position of camera and 3D samples to generate a list of voxel coordinates which satisfy *along-the-ray* criteria.

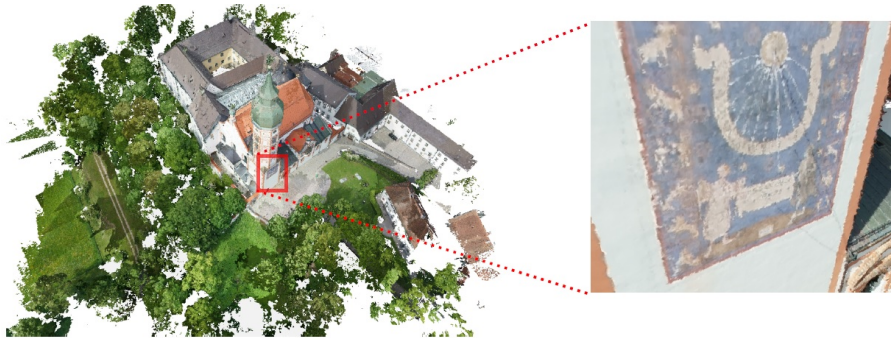


Figure 4.9: Large-scale 3D reconstruction using hashed voxel grid.

The selected voxels are then updated with computed signed distance values, Funk and Börner (2016) demonstrated the effectiveness of using hashed voxels to store large-scale 3D models in a real-time scenario. Figure 4.9 highlights the amount of details captured using such hashed voxel-grid.

In principal, computer programs and algorithms are expected to exhibit memory-processing trade-off relations. Pre-allocated and bounded dense reconstruction algorithms are usually faster in terms of processing due to pre-defined memory accesses while modern semi-dense representation allows boundless reconstruction at the expense of extra calculations for each memory access. Similarly, the execution time of implementation depends greatly on the type of representation. Based on aforementioned characteristics, voxel-block based volumetric representation is preferred over dense and *along-the-ray* for the actual implementation of the proposed framework since it allows boundless reconstruction while allowing fewer hashed address calculation.

4.5 Summary

This chapter provided in-depth theoretical background of underlying implicit representation and related causes and effects. The concept of fusing SDF values to reduce noise effects is formally introduced and the rationale behind the *convergence* of the estimated implicit surface is presented. Furthermore, properties of semi-dense implementations of voxel-grids such as voxel-block and *along-the-ray* are discussed to highlight performance and quality trade-offs. Provided analytical discussion and core concepts are utilized in developing the core-principle of the proposed research and are discussed in detail in upcoming chapter.

Chapter 5

Concept and Design

This chapter presents theoretical insights and rationales behind proposed research contributions which serve as solutions to the overall research question. In Chapter 2, it was discussed that an efficient 3D reconstruction framework should be able to integrate incremental depth updates to an existing representation while exhibiting a robust profile to handle depth noise and outliers. Since these characteristics are fully aligned with the overall research objectives, this chapter is divided into three sections to focus each characteristic individually while keeping the underlying rationale distilled.

Initially, a novel least square estimation based alternative to the traditional weighted integration method is introduced and a recursive form is derived which highlights the flexibility of the proposed scheme to utilize the quality of depth measurements in an optimal manner. Secondly, the regularization information is integrated into a least square estimator to handle erroneous depth measurements by applying the total variation denoising in a recursive manner. This regularization aspect is shown to produce smoother surfaces using comparatively less input sample data. Furthermore, a robust outliers removal technique is introduced which targets isolated and sparse outliers in real-time. Finally, the overall design of the 3D reconstruction framework is presented which utilizes the proposed contributions to achieve high-quality 3D models from series of depth and color images.

5.1 Recursive least squares as 3D fusion approach

Curless and Levoy (1996) argued that the optimality of weighted integration and the resulting

iso-surface is equivalent to a least square minimizer system. Although the actual intent of the provided proof is purely conceptual, their equivalence relation is an important analogy. Based on this proof, an approximate solution to least square minimizer based integration systems can be implemented and employed to perform the 3D fusion. In principal, such system is expected to show similar noise reduction and depth integration characteristics as a weighted SDF fusion. Furthermore, capabilities of a least square estimator are highly expendable in terms of working principal such as weighted least squares, linear, non-linear and regularized least squares. These characteristics provided the needed motivation to develop and implement a novel least square based integration system.

In order to describe the problem of depth fusion as a least square estimator, the observable environment is represented as a semi-dense voxel grid as proposed by Rajput et al. (2016) in which a fixed number of voxel locations (referred to as *support*) around a 3D sample are accessed and their SDF values are represented as a standard vector notation. These vectorized implicit values (written compactly as SDF-signal in upcoming text) are used as input and output of the linear least square estimator represented by equation

$$Y = \Phi \hat{x} + \nu \quad (5.1)$$

where linear system coefficients Φ are used to estimate \hat{x} from Y and resulting ν is the approximation error. Considering a typical scenario where the number of signals n used for an estimation is greater than *support* (denoted as m), such system is expected to produce an approximate solution which satisfies all versions of $y_i \in Y$. The aforementioned set of n input signals, estimated output and system coefficients can be arranged in a matrix notation to simplify the mathematical representation, and can be written as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix} \hat{x}_n + \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_n \end{bmatrix} \quad (5.2)$$

where \hat{x}_n is the SDF signal approximated by integrating n instances of noisy SDF signals, and ν_n denotes the estimation error which is expected to monotonically decrease with incremental updates. Such representation of a least squares estimation is computationally expensive since every new $(n + 1)^{th}$ instance of y and ϕ are concatenated with existing data and consequently computation time for underlying mathematical operation grows exponentially. Therefore, a recursive least square solution is usually employed in practical applications. The mathematical derivation is described in Section 5.1.1.

In practical applications, the true state of system x is expected to remain unknown, since every attempt to measure x will further increase the overall estimation error. It is therefore assumed that importance of ν is insignificant and removal of this term does not affect the overall system design. Based on the aforementioned characteristics, Equation 5.1 can be reduced to a minimization problem defined by

$$\min \|Y - \Phi\hat{x}\|^2$$

Such least squares estimator is expected to produce similar convergent behaviour as a traditional weighted fusion, however the true potential of such representation is demonstrated in Sections 5.2 where the regularization parameter is introduced to extend the capability of a least square system to handle depth noise inherently.

5.1.1 Weighted least squares and standard derivation of ML-Estimate

In order to derive¹ a recursive form of a least square estimator from Equation 5.1, the difference between estimated values \hat{x} and noisy measurements can be written as

$$\epsilon = Y - \Phi\hat{x} \quad (5.3)$$

A cost function $J(\hat{x})$ which tries to find the value of \hat{x} through a minimization process can be written as:

$$\begin{aligned} J(\hat{x}) &= \epsilon^T \epsilon \\ &= (Y - \Phi\hat{x})^T (Y - \Phi\hat{x}) \\ &= Y^T Y - \hat{x}^T \Phi Y - Y^T \Phi \hat{x} + \hat{x}^T \Phi^T \Phi \hat{x} \end{aligned} \quad (5.4)$$

¹Mathematical derivation of recursive estimator is adapted from Simon (2006) and modified to accommodate depth estimation.

Partial derivative of J with respect to \hat{x} is employed to achieve the necessary vanishing condition of minimization, that is,

$$\frac{\partial J}{\partial \hat{x}} = -2Y^T\Phi + 2\hat{x}^T\Phi^T\Phi = 0$$

Solving for \hat{x} ,

$$\hat{x} = (\Phi^T\Phi)^{-1}\Phi^TY \quad (5.5)$$

where Φ and Y are augmented matrices and their values can be used from Equation 5.2.

A typical least square estimator applies equal weights to every accumulated measurement, this weighting mechanism is somehow flawed since it presumes a linear relation between the measuring depth to the accumulated error. In order to integrate a weighting mechanism in the least square estimation, a weight value calculated from a respective noise model (e.g. see Equation 3.1 for Kinect depth sensing) is employed with each measurement $y_i : 1 \leq i \leq n$. Typically, a covariance matrix R containing $\sigma_i^2 : 1 \leq i \leq n$ for each measurement is used to reflect the weighting aspect, that is,

$$R = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

Equation 5.4 which minimizes the sum of squared differences weighted with weight matrix R can be written as

$$J(\hat{x}) = \epsilon^T R^{-1} \epsilon = \frac{\epsilon_1^2}{\sigma_1^2} + \frac{\epsilon_2^2}{\sigma_2^2} + \cdots + \frac{\epsilon_n^2}{\sigma_n^2}$$

J can be expanded as follows:

$$\begin{aligned} J(\hat{x}) &= \epsilon^T \epsilon \\ &= (Y - \Phi\hat{x})^T R^{-1} (Y - \Phi\hat{x}) \\ &= Y^T R^{-1} Y - \hat{x}^T \Phi R^{-1} Y - Y^T R^{-1} \Phi \hat{x} + \hat{x}^T \Phi^T R^{-1} \Phi \hat{x} \end{aligned} \quad (5.6)$$

Similarly, minimizing J with respect to \hat{x} and solving for \hat{x} yields,

$$\begin{aligned}\frac{\partial J}{\partial \hat{x}} &= -2Y^T R^{-1} \Phi + 2\hat{x}^T \Phi^T R^{-1} \Phi = 0 \\ \hat{x} &= (\Phi^T R^{-1} \Phi)^{-1} \Phi^T R^{-1} Y\end{aligned}\tag{5.7}$$

The solution of Equation 5.7 exist only when the matrix R is non-singular, i.e. every measurement y_i is corrupted with some degree of noise for the estimation technique to work.

As stated earlier, augmenting Y , Φ and calculating the inverse with each incremental update is computationally expensive task. Therefore a recursive update algorithm can be formulated which can utilize the existing system estimate \hat{x}_{k-1} to compute \hat{x}_k without tedious matrix augmentation and inversion. Such typical linear recursive estimator can be written as,

$$\begin{aligned}y_k &= \Phi_k x + \nu_k \\ \hat{x}_k &= \hat{x}_{k-1} + K_k (y_k - \phi_k \hat{x}_{k-1})\end{aligned}\tag{5.8}$$

where ϕ_k is a $m \times m$ system coefficient matrix for instance k where m is the *support* of SDF signal. The correction term $(y_k - \phi_k \hat{x}_{k-1})$ (i.e. transition of \hat{x}_k from previous estimate \hat{x}_{k-1}) is controlled by the *estimator gain matrix* denoted by K_k having $m \times m$ dimensions. Therefore, the current estimation error is

$$\begin{aligned}\epsilon_k &= x - \hat{x}_k \\ &= x - \hat{x}_{k-1} - K_k (y_k - \phi_k \hat{x}_{k-1}) \\ &= \epsilon_{k-1} - K_k (\phi_k x + \nu_k - \phi_k \hat{x}_{k-1}) \\ &= \epsilon_{k-1} - K_k \phi_k (x - \hat{x}_{k-1}) - K_k \nu_k \\ &= (I - K_k \phi_k) \epsilon_{k-1} - K_k \nu_k\end{aligned}\tag{5.9}$$

where I is the $m \times m$ identity matrix. The mean of this error can be written as,

$$E(\epsilon_k) = E(I - K_k \phi_k) E(\epsilon_{k-1}) - K_k E(\nu_k)$$

A typical least square estimator is expected to exhibit *unbiased* behavior towards each measurement. It is therefore assumed that on average, an estimated \hat{x}_k and the true value of x are roughly equal. In principal, an optimal value of gain matrix K_k is expected to reduce

the aggregated variance of the estimated error, therefore a cost function for such optimality criterion can be written as:

$$\begin{aligned}
 J_k &= E(\|x - \hat{x}_k\|^2) \\
 &= E(\epsilon_k^T \epsilon_k) \\
 &= E(\text{tr}(\epsilon_k \epsilon_k^T)) \\
 &= \text{tr}(P_k)
 \end{aligned}$$

where the trace operator (tr) is applied to the aggregated variance $P_k = E(\epsilon_k \epsilon_k^T)$. The value of the estimation error ϵ_k from Equation 5.9 can be employed to obtain P_k as follows:

$$\begin{aligned}
 P_k &= E(((I - K_k \phi_k) \epsilon_{k-1} - K_k \nu_k)((I - K_k \phi_k) \epsilon_{k-1} - K_k \nu_k)^T) \\
 &= (I - K_k \phi_k) E(\epsilon_{k-1} \epsilon_{k-1}^T) (I - K_k \phi_k)^T - K_k E(\nu_k \epsilon_{k-1}^T) (I - K_k \phi_k)^T \\
 &\quad - (I - K_k \phi_k) E(\epsilon_{k-1} \nu_k^T) K_k^T + K_k E(\nu_k \nu_k^T) K_k^T
 \end{aligned}$$

The estimation error computed at time $k - 1$ is independent of the measurement y_k and respective noise ν_k at time k , which implies that

$$\begin{aligned}
 E(\nu_k \epsilon_{k-1}^T) &= E(\nu_k) E(\epsilon_{k-1}^T) = 0 \\
 E(\epsilon_{k-1} \nu_k^T) &= E(\epsilon_{k-1}) E(\nu_k^T) = 0
 \end{aligned}$$

By using the weight matrix R_k and the implied estimation-error relation, the expression for P_k becomes

$$P_k = (I - K_k \phi_k) P_{k-1} (I - K_k \phi_k)^T + K_k R_k K_k^T \quad (5.10)$$

It is worth mentioning that there exists a strong correlation between the estimation cost J and the covariance matrix P_k . Since an optimal value of the gain matrix K_k is expected to minimize the cost function, such minimization can be obtained by differentiating the cost function with

respect to K_k which can be written and simplified² as follows:

$$\begin{aligned} \left(\frac{\partial J_k}{\partial K_k} \right)^T &= \left(\frac{\partial(\text{tr}(P_k))}{\partial t} \right)^T \\ \left(\frac{\partial J_k}{\partial K_k} \right)^T &= 2(I - K_k \phi_k) P_{k-1} (-\phi_k^T) + 2K_k R_k \end{aligned}$$

The optimal value of the gain matrix can be obtained by setting the partial derivative to zero and solving for K_k

$$K_k = P_{k-1} \phi_k^T (\phi_k P_{k-1} \phi_k^T + R_k)^{-1} \quad (5.11)$$

let $S_k = \phi_k P_{k-1} \phi_k^T + R_k$ for simplicity, so K_k becomes

$$K_k = P_{k-1} \phi_k^T S_k^{-1} \quad (5.12)$$

The substitution of simplified K_k into Equation 5.10 followed by the expansion as follows:

$$\begin{aligned} P_k &= (I - P_{k-1} \phi_k^T S_k^{-1} \phi_k) P_{k-1} (I - P_{k-1} \phi_k^T S_k^{-1} \phi_k)^T + P_{k-1} \phi_k^T S_k^{-1} R_k S_k^{-1} \phi_k P_{k-1} \\ &= P_{k-1} - P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} - P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} + \\ &\quad P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} + P_{k-1} \phi_k^T S_k^{-1} \underline{R_k} S_k^{-1} \phi_k P_{k-1} - 1 \\ &\text{merging the underlined terms into } S_k \\ &= P_{k-1} - P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} - P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} + P_{k-1} \phi_k^T S_k^{-1} S_k S_k^{-1} \phi_k P_{k-1} \quad (5.13) \\ &= P_{k-1} - 2P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} + P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} \\ &= P_{k-1} - P_{k-1} \phi_k^T S_k^{-1} \phi_k P_{k-1} \\ &= P_{k-1} - K_k \phi_k P_{k-1} \text{ by 5.12} \\ &= (I - K_k \phi_k) P_{k-1} \end{aligned}$$

Since K_k and P_k are inter-related and their values are computed in a recursive manner from P_{k-1} and H_{k-1} respectively, the overall least square system is expected to reduce estimation costs over time. This *convergent* behaviour of the system is depicted in Figure 5.1 where noisy synthetic SDF signals are integrated in recursive fashion and compared against the ground truth model. It is worth mentioning that the derived system has inherent similarities with Kalman

²using the matrix manipulation property that $\frac{\partial}{\partial t}(ABA^T) = 2AB$ when B is symmetric

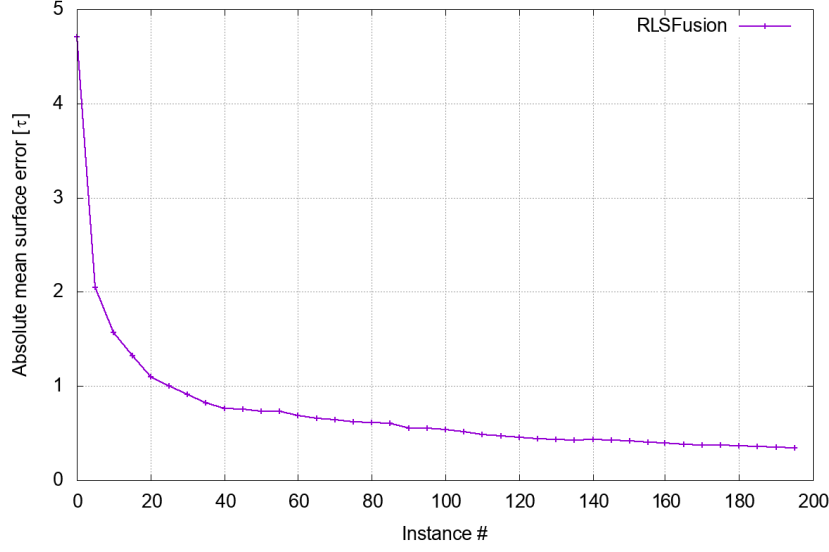


Figure 5.1: Mean absolute surface error convergence with incremental fusion.

filter, however it was observed during experimental evaluation that extending the capabilities of standard kalman filter is not a trivial task. It was therefore decided to use the derived version instead of standard kalman filter for further evaluation and development. For the sake of compactness, this recursive least square 3D fusion is referred to as *RLSFusion* in upcoming text.

5.1.2 Depth fusion with recursive 3D fusion

Considering the scenario presented in Section 4.3 where two depth measurements of suspected surface at 11τ units with added depth noise of form $\mathcal{N}(\mu, \sigma) = \mathcal{N}(0.0, 1.0\tau)$ are captured and represented with TSDF signals (y_0 and y_1). The resulting error-prone depth measurements represented with TSDF are shown in Figure 5.2. The recursive solution to the least square problem from Equation 5.8 can be used in incremental fashion to integrate y_0 and y_1 to estimate \hat{x}_1 . Assuming that $n = 7$ denotes *support* of SDF-signal, then the covariance matrix P_0 is initialized as identity matrix of order $n \times n$ and x_0 . Initially, the system presumes that the provided input instance y_0 reflects the nature of the estimated signal accurately, therefore x_0 is set to y_0 . Afterwards, for each new SDF-signal y_k the system calculates the estimation gain matrix and the covariance matrix applying Equations 5.11 and 5.13 respectively. It is therefore expected that each incremental update y_k contributes to the estimation of x_k however the impact of all contribution is decreasing monotonically as the belief of the system grows

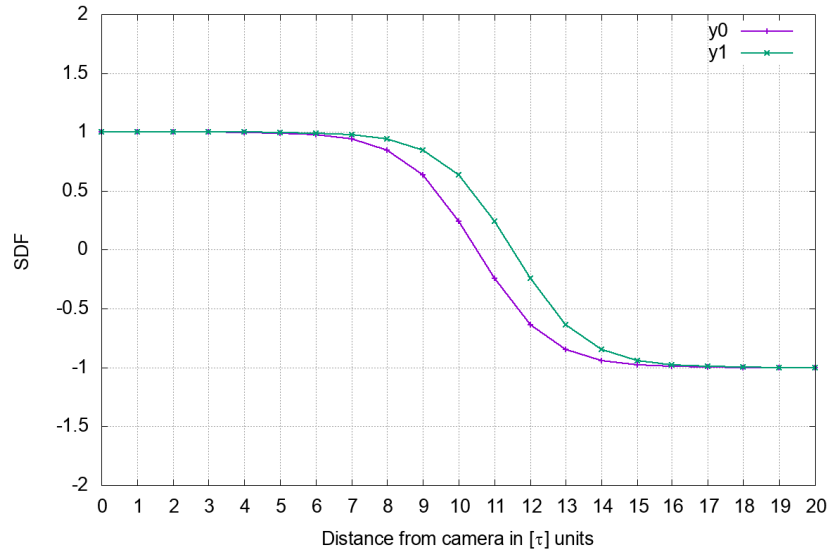


Figure 5.2: Error-prone depth measurements (y_0 and y_1) represented as one-dimensional TSDF function.

with each update. Figure 5.3 shows the zero crossing of the resulting x_k between y_0 and y_1 .

In order to obtain an un-biased quantitative evaluation of RLSFusion against traditional weighted fusion, both methods were provided with 200 instances of a synthetic signal with additive Gaussian noise of form $\mathcal{N}(\mu, \sigma) = \mathcal{N}(0.0, 5.0\tau)$. Figure 5.4 shows the behavior of the mean absolute surface estimation error converging to the sub-pixel accuracy in both cases.

5.1.3 Properties of RLSFusion

True potential of employing RLSFusion as a substitute to traditional weighted SDF fusion comes from the fact that the process of SDF signal fusion can be controlled with external parameters without modifying the volumetric representation or weights. Unlike traditional fusion method in which incremental weight values are responsible of defining the *belief* of a suspected surface, RLSFusion utilizes underlying estimator gain values which can be modified or reset on-demand. This control of estimator gain values allows the system to accommodate a sensor noise model, depth noise and localization errors in a convenient way. Following properties of RLSFusion have been identified using an extensive evaluation:

- Low memory footprint:

In order to highlight the memory footprint of RLSFusion, considering a synthetic

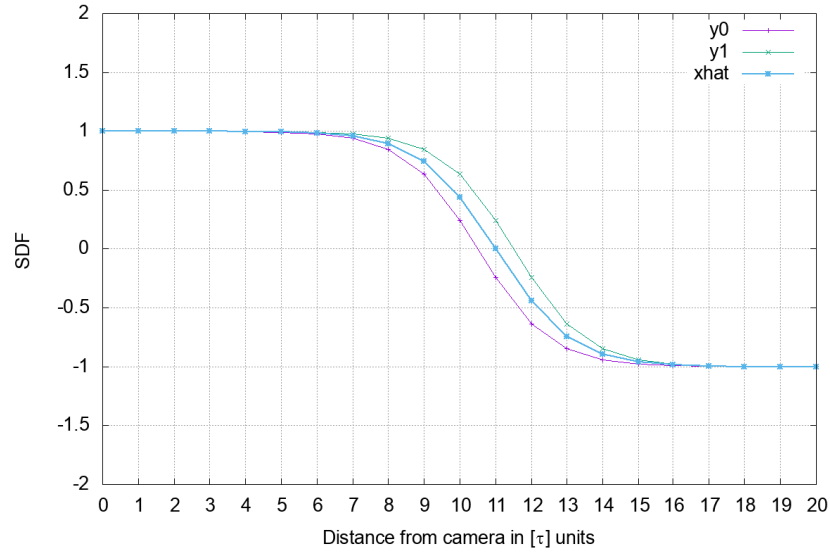


Figure 5.3: Fused TSDF function with the help of Equation 5.8.

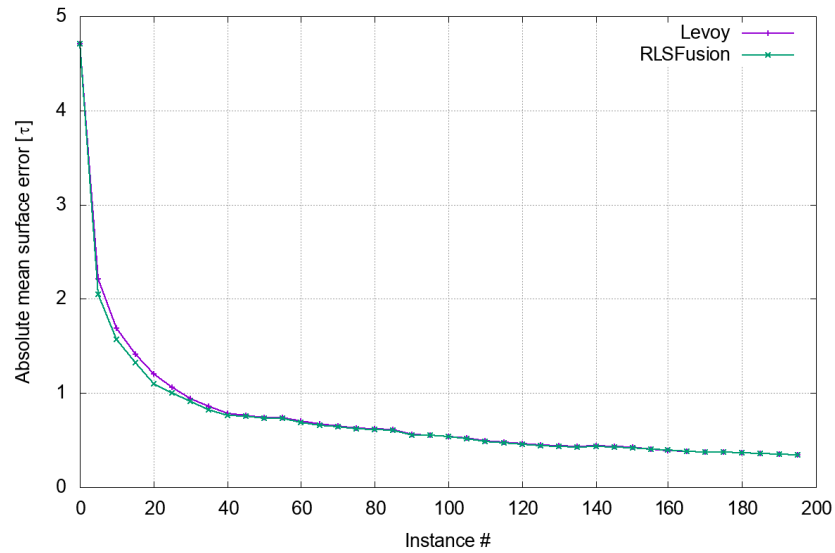


Figure 5.4: Comparison of the convergence between RLSFusion and traditional 3D fusion.

piecewise 3D and 2D signals from Section 3.3.1 are to be represented in an empty voxel grid. Since RLSFusion uses underlying system variables such as estimator gain matrix K_k and covariance matrix P_k to control the estimation process instead of weight values, the values of these matrices can be stored in simple memory array. Since these system variables are updated with respect to the number of time an instance is updated, therefore a single copy of these variables is sufficient. Therefore, the need of storing all the weight values is unnecessary. This representation of system allows RLSFusion to utilize memory in efficient manner. Table 5.1 presents a comparative overview of memory utilization by representing a 2D and 3D synthetic signal with traditional dense, sparse and RLSFusion.

Table 5.1: Comparison of memory consumption (in bytes) among dense, sparse vs RLSFusion.

	Dense	Sparse	RLSFusion
2D Signal	320000	12800	5608
3D Signal	64000000	2560000	1120008

- Controllable gain:

RLSFusion provides a flexible mechanism to control the behavior of fusion with the help of externally provided weights. Implementation of the proposed RLSFusion uses a weighting mechanism to control and manipulate the amplitude of the estimator gain to accommodate less noisy depth measurements. To highlight this capability, a scenario is considered in which k instances of depth measurements are fused using weighted integration. This implies that the impact of each incremental update decreases over time regardless of the quality of measurement. This problem is handled efficiently by RLSFusion with the help of forcing the system to accommodate updates, Figure 5.5 shows the behavior in which the system is provided with 10 less noisy depth signals at $k = 100$. It can be observed that the behavior of the traditional integration method is inflexible (since weights of the system become more rigid overtime) while RLSFusion adapts quickly and produces a less overall error due to this adaptation.

Traditional least square estimators use this weighting mechanism to employ *exponential forgetting* in which a system can be programmed to focus the estimated state towards recent updates while ignoring older measurements. Such utilization of weights can be used to accommodate dynamics of the environment in a real-time volumetric

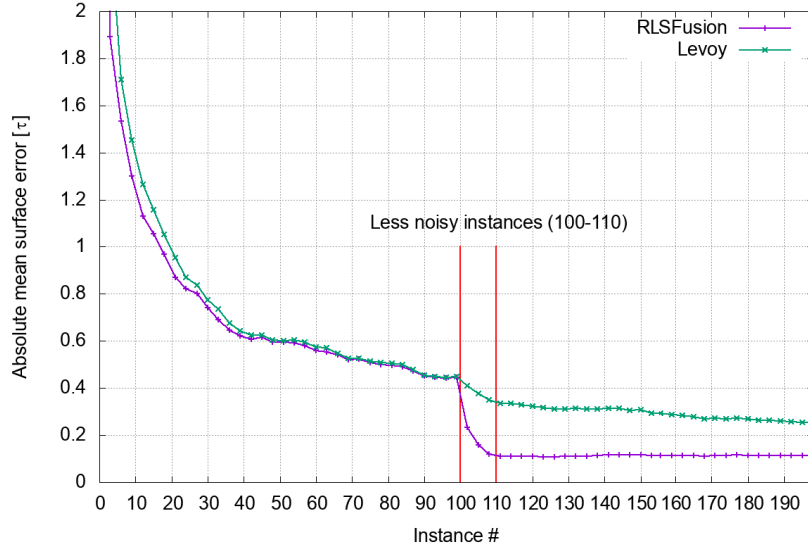


Figure 5.5: Capability of RLSFusion to accommodate less noisy measurements.

reconstruction. Since the main focus of this research is to reduce errors in measurements, this dynamic implementation is left intentionally as a future research direction.

Although proposed RLSFusion shows attractive improvements over traditional methods, dealing with error-prone depth measurements remained untouched. The upcoming section will introduce a novel mechanism to integrate smoothness priors as a regularization parameter in a recursive least square estimator.

5.2 Regularized Recursive Fusion

Section 2.3 discussed the possibility of using external depth smoothing image filters to reduce depth noise. Graber et al. (2015) argued that unconstrained depth smoothing such as applying bilateral filtering can degrade depth images by producing stair-case effects since the filter does not respect the 3D geometry. However, the regularized depth image regularization technique by Graber et al. (2015) suffers from high computational complexity, therefore employing such a technique in incremental integration system is infeasible. Calakli and Taubin (2011) proposed to enforce a regularization constraint which forces implicit values of each voxel to follow a smooth overall surface. As a result, reconstructed surfaces from regularized SDF produce smoother surfaces. Since SSDF is a post-processing step, it presumes that existing measurements are final and all 3D information is properly represented in an octree.

In principal, capabilities of a least squares estimation based 3D fusion system (such as RLSFusion) can be extended to handle error-prone depth samples or implicit SDF signals with the introduction of a regularization constraint. Such regularized system having the properties of a least square system can be expressed as a minimization problem defined by:

$$\|\Phi\hat{x} - Y\|^2 + \lambda\|g(x)\|^2 \quad (5.14)$$

where $g(x)$ is a penalization function, \hat{x} is an estimated SDF signal from augmented Y (similar to RLSFusion) and λ is the regularization parameter which controls the effects of smoothing.

In principal, selecting the penalization function $g(x)$ as a second order finite difference function allows the system to utilize implicit values from neighbouring elements to obtain smoother estimations. Similar to RLSFusion, the system is represented in the matrix/vector notation to utilize modern CPU architectures. Therefore, Equation 5.14 can be re-written as

$$\|\Phi\hat{x} - Y\|^2 + \lambda\|Dx + C\|^2 \quad (5.15)$$

where D and C matrices are designed to facilitate finite differences. The actual derivation of D and C matrices is discussed in Section A.1.

Theoretically, this regularized least squares estimator is expected to handle depth noise inherently since each element of Y is penalized to maintain a low *total-variation* profile. The upcoming section derives a recursive formulation of the aforementioned regularized least square estimator (written compactly as *RFusion*).

5.2.1 Derivation of regularized least squared 3D fusion

In order to derive a recursive form of the regularized least square estimator from Equation 5.15, a cost function $J(\hat{x})$ which transforms the problem in a least square notion can be written as

$$J(\hat{x}) = \min(\|\Phi\hat{x} - Y\|^2 + \lambda\|Dx + C\|^2) \quad (5.16)$$

The partial derivative of $J(\hat{x})$ is employed to achieve necessary the minimization condition as:

$$\begin{aligned}
 J(\hat{x}) &= 2\Phi^T(\Phi\hat{x} - Y) + 2\lambda D^T(D\hat{x} + C) = 0 \\
 0 &= 2\Phi^T\Phi\hat{x} - 2\Phi^TY + 2\lambda D^TD\hat{x} + 2\lambda D^TC \\
 0 &= (\Phi^T\Phi + \lambda D^TD)\hat{x} + \lambda D^TC - \Phi^TY \\
 0 &= (\Phi^T\Phi + \lambda D^TD)\hat{x} + \Phi^T\left(\frac{\lambda D^TC}{\Phi^T} - Y\right) \\
 \hat{x} &= (\Phi^T\Phi + \lambda D^TD)^{-1} \left(\Phi^T \left(Y - \frac{\lambda D^TC}{\Phi^T} \right) \right)
 \end{aligned} \tag{5.17}$$

Let $\hat{Y} = \left(Y - \frac{\lambda D^TC}{\Phi^T} \right)$ for simplicity, then the regularized least square estimator can be written as:

$$\hat{x} = (\Phi^T\Phi + \lambda D^TD)^{-1} \Phi^T \hat{Y} \tag{5.18}$$

where the SDF signal \hat{x} is estimated from noisy depth measurements and $y_i : i = 0 \leq i \leq k$ is augmented in matrix form Y .

As mentioned earlier, augmentation of matrices Φ_{k-1} and Y_{k-1} for each incremental update results in computationally expensive calculations. Assuming a recursive successive relation among incremental updates, then Φ and \hat{Y} can be written as follows:

$$\Phi_k = \begin{bmatrix} \Phi_{k-1} \\ \phi \end{bmatrix} \quad \hat{Y}_k = \begin{bmatrix} \hat{Y}_{k-1} \\ \hat{y} \end{bmatrix} \quad \text{and } D_k = \begin{bmatrix} D_{k-1} \\ d \end{bmatrix} \tag{5.19}$$

Equation 5.18 for k^{th} instance can be written as

$$\hat{x}_k = (\Phi_k^T\Phi_k + \lambda D_k^TD_k)^{-1} \phi_k^T \hat{Y}_k$$

Let $P_k = (\Phi_k^T\Phi_k + \lambda D_k^TD_k)^{-1}$ for the sake of simplicity, then using the incremental updates

property P_k can be simplified as

$$\begin{aligned}
P_k &= \left(\begin{bmatrix} \Phi_{k-1} \\ \phi \end{bmatrix} \begin{bmatrix} \Phi_{k-1} & \phi \end{bmatrix} + \lambda \begin{bmatrix} D_{k-1} \\ d \end{bmatrix} \begin{bmatrix} D_{k-1} & d \end{bmatrix} \right)^{-1} \\
P_k &= (\Phi_k^T \Phi + \phi^t \phi + \lambda D_k^T D + \lambda d_k^T d)^{-1} \\
P_k &= ((\Phi_k^T \Phi + \lambda D_k^T D) + (\phi^t \phi + \lambda d_k^T d))^{-1} \\
P_k &= (P_{k-1}^{-1} + (\phi^t \phi + \lambda d_k^T d))^{-1} \\
\boxed{P_k^{-1} &= P_{k-1}^{-1} + (\phi^t \phi + \lambda d_k^T d)} \\
P_k &= \left(P_{k-1}^{-1} + \begin{bmatrix} \phi^T \phi & I \end{bmatrix} \begin{bmatrix} I \\ \lambda d^T d \end{bmatrix} \right)^{-1}
\end{aligned} \tag{5.20}$$

For simplicity assuming $B = \begin{bmatrix} \phi^T \phi & I \end{bmatrix}$ and $C = \begin{bmatrix} I \\ \lambda d^T d \end{bmatrix}$ we get

$$P_k = \left(P_{k-1}^{-1} + BC \right)^{-1}$$

Using matrix inversion lemma

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}$$

$$P_k = P_{k-1} - P_{k-1}B(I + CP_{k-1}B)^{-1}CP_{k-1} \tag{5.21}$$

Equation 5.2.1 with substitution of P_k can be written as follows

$$\begin{aligned}
\hat{x}_k &= P_k \phi_k^T \hat{Y}_k \\
P_k^{-1} \hat{x}_k &= \phi_k^T \hat{Y}_k
\end{aligned} \tag{5.22}$$

By using the assumption of incremental updates from Equation 5.19, the estimator can be written as

$$\hat{x}_k = P_k \left(\Phi_{k-1}^T \hat{Y}_{k-1} + \phi^T \hat{y}_k \right) \tag{5.23}$$

Similarly for $(k - 1)^{th}$ instance

$$P_{k-1}^{-1} \hat{x}_{k-1} = \phi_{k-1}^T \hat{Y}_{k-1}$$

Using the value of $P_{k-1}^{-1} \hat{x}_{k-1}$ in Equation 5.23 we get

$$\begin{aligned} \hat{x}_k &= P_k \left(\phi_{k-1}^T \hat{Y}_{k-1} + \phi^T \hat{y}_k \right) \\ \hat{x}_k &= P_k \left(P_{k-1} \hat{x}_{k-1} + \phi^T \hat{y}_k \right) \end{aligned} \quad (5.24)$$

Substituting the value of P_k from equation 5.20 we get

$$\begin{aligned} \hat{x}_k &= \left[(P_k^{-1} - (\phi^T \phi + \lambda d^T d)) \hat{x}_k + \phi^T \hat{y}_k \right] \\ &= \left(P_k P_k^{-1} - P_k (\phi^T \phi + \lambda d^T d) \right) \hat{x}_{k-1} + P_k \phi^T \hat{y}_k \\ &= P_k P_k^{-1} \hat{x}_{k-1} - P_k (\phi^T \phi + \lambda d^T d) \hat{x}_{k-1} + P_k \phi^T \hat{y}_k \\ &= \hat{x}_{k-1} - P_k \left(\phi^T \phi + \lambda d^T d \right) \hat{x}_{k-1} + P_k \phi^T \hat{y}_k \\ \hat{x}_k &= \hat{x}_{k-1} + P_k \left[\phi \hat{y}_k - (\phi^T \phi + \lambda d^T d) \hat{x}_{k-1} \right] \end{aligned} \quad (5.25)$$

By using the actual value of \hat{y}_k the final update equation of RFusion becomes

$$\hat{x}_k = \hat{x}_{k-1} + P_k \left[\phi \left(y_k - \frac{\lambda d^T c}{\phi^T} \right) - (\phi^T \phi + \lambda d^T d) \hat{x}_{k-1} \right]$$

In principal, RFusion uses Equations 5.2.1 and 5.21 to update the system estimate and calculate the gain respectively. The value of λ in Equation 5.2.1 which controls the amount of regularization can be selected as a constant at the time of execution. Since *prior* smoothing information is integrated in the overall system design, the system inherently reduces noise artifacts and provides a faster convergence of the absolute surface error compared to both traditional weighted fusion and RLSFusion. This fast convergence effect is shown in 5.6 where the absolute surface error produced by RFusion reached to sub pixel accuracy after fusing 10 instances while both traditional and RLSFusion reached same accuracy after 22 instances. The rationale behind faster convergence in the case of noisy data is elaborated in the upcoming section.

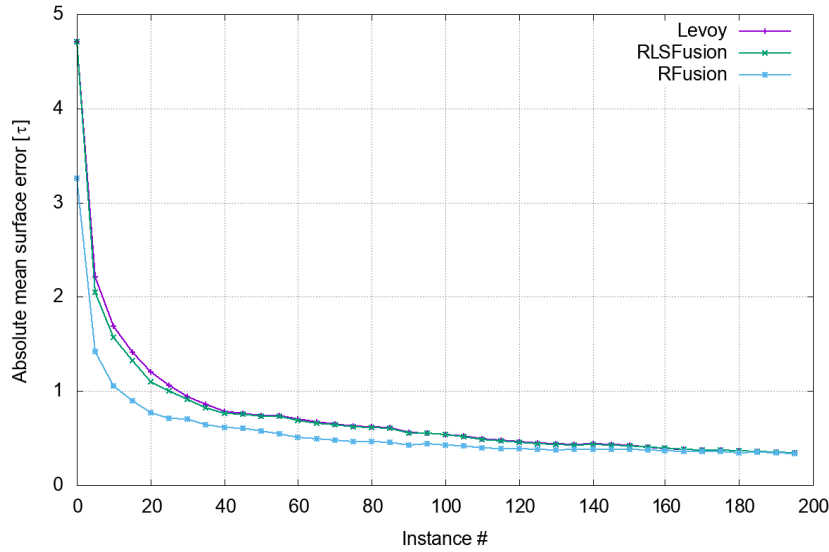


Figure 5.6: Mean absolute surface error convergence with incremental fusion.

5.2.2 Faster Convergence with regularized fusion

Regularized aspect of RFusion takes advantage of neighboring SDF values and introduces a scalar quantity which reduces overall difference among neighbouring voxel values. This addition of counter weight is analogous to using a total variation denoising mechanism on implicit values. Figure 5.6 demonstrates that the proposed RFusion achieves faster convergence to sub-pixel accuracy, however both traditional and RLSFusion catch up with absolute surface error eventually. In practical applications, either the sensing equipment is moved across the environment or the object is moved in-front of the depth sensor. It is therefore unlikely to capture sufficient depth images for a traditional fusion approach to estimate the surface by convergence at the same accuracy. Furthermore, traditional visualSLAM algorithms such as ORBSLAM2 expects sufficient sensor movement in terms of rotation and translation to reduce the localization error. This inverse relationship severely affects the overall 3D reconstruction process. Therefore in such case, RFusion can produce high-quality 3D models with the help of regularized integration while traditional incremental approaches struggles with this situation.

In order to highlight the regularization aspect of RFusion while isolating the effects from incremental fusion, a set-up similar to Section 4.3 is presumed and two depth measurements y_0 and y_1 are recorded and represented as SDF signals. To highlight the potential of RFusion,

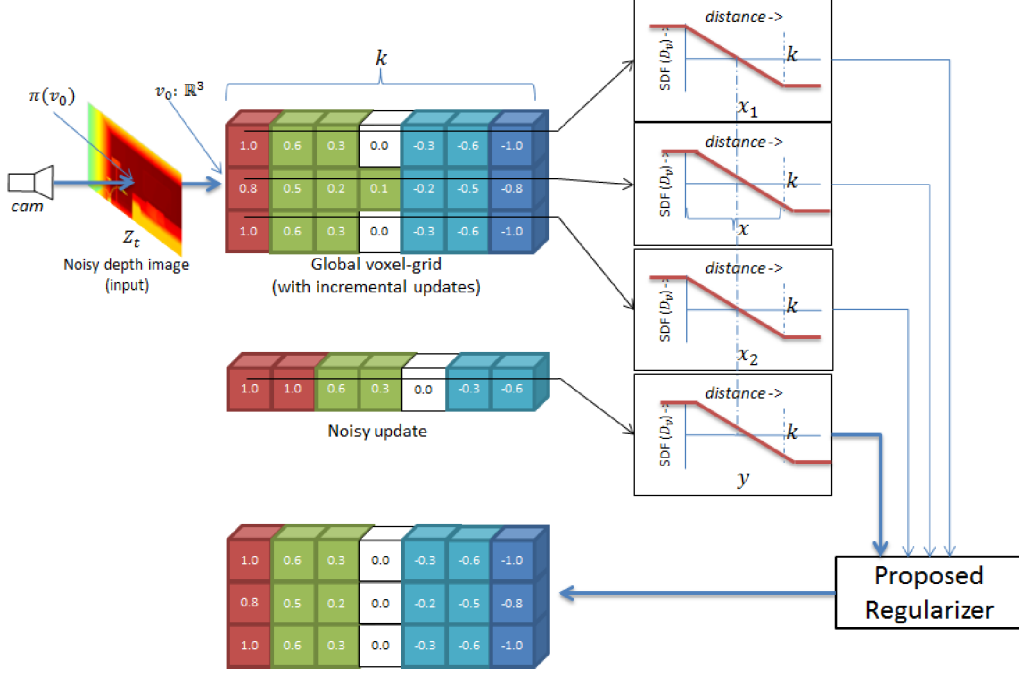


Figure 5.7: Illustration of volumetric regularization and integration process using color coded voxel values.

a challenging scenario is presumed in which both depth measurements are far-sighted³ which implies that the estimated implicit surface from traditional integration methods does not reduce the overall estimation error. Figure 5.8.a shows the erroneous depth measurements and the acquired iso-surface from traditional weighted fusion.

Figure 5.7 illustrates the intuition behind the proposed regularized integration using a noisy depth image. Consider a situation in which the volumetric representation (denoted with x) is updated with a noisy depth update (represented as vectorized SDF signal y). The proposed system extracts neighboring voxel values and arrange them in a vectorized form (denoted by x_1 and x_2) followed by applying the proposed regularization constraint to achieve overall smooth volumetric representation.

In such challenging scenario, RFusion utilizes the underlying *total variation denoising* method on SDF values to reduce implicit surface deformities. Figure 5.8.b shows that estimated iso-surface from RFusion is influenced (more specifically, *regularized*) with neighboring implicit values. In principal, the influence of the regularization parameter λ is supposed to

³RFusion is capable of handling various types of noise efficiently, far-sighted measurements are selected purely to demonstrate the effectiveness.

decrease with incremental updates. This can be achieved by linking the value of λ with the *gain* of the least square estimator. In such case, when the value of λ is equal to 0, the system behaves similar to RLSFusion and Equation 5.2.1 (containing the recursive version of the proposed regularized system) is approximately equivalent to Equation 5.8.

To highlight the effectiveness in a practical scenario, 10 depth images with successive timestamps from ICL-NUIM's *living-room* dataset Handa et al. (2014) were selected and processed with RFusion and traditional volumetric fusion. Figure 5.9 shows the qualitative comparison between standard 3D fusion in Curless and Levoy (1996) and the proposed regularized fusion. It is evident from visual inspection of Figure 5.9 that proposed regularized fusion is reducing noise effects in an efficient manner.

5.3 Outliers removal using spatial information

Section 2.5 introduced the concept of depth outliers and classified them into sparse, isolated and non-isolated categories. The proposed regularized volumetric 3D fusion method handles the effects of non-isolated depth outliers, however dealing with sparse and isolated depth outliers remain a challenging research problem. Traditional approaches which are designed to eliminate these outliers are not suitable for real-time applications due to cumbersome memory access. In this section, a novel outliers removal technique *SORF* is proposed which eliminates sparse and isolated outliers on the basis of their spatial proximity with respect to expected surface.

The process of outliers detection and removal involves three linear passes on provided 3D points (denoted by P_i where $0 < i < n$). In a first pass, an empty pre-aligned sparse grid G_{local} is initialized and all points are registered into small bounding boxes (referred to as *cubes*). At the time of registration, a counter value associated with each cube is incremented. Since the grid is sparse and preferably implemented with hashed memory access, it is possible to obtain the list of active cubes C_k where $0 < k < m$ (where m is the number of all cubes).

In a second pass, each cube in C_k is assessed and labeled as either *active* or *potential* depending upon the *counter* value. This assessment depends greatly on the spatial dimensions of the cube and scale of representation, therefore it is presumed that a parameter *thresh* is selected to an appropriate value. In empirical analysis, it was observed that a typical cube representing a 8cm x 8cm x 8cm spatial bounding box should contain at least 30 samples.

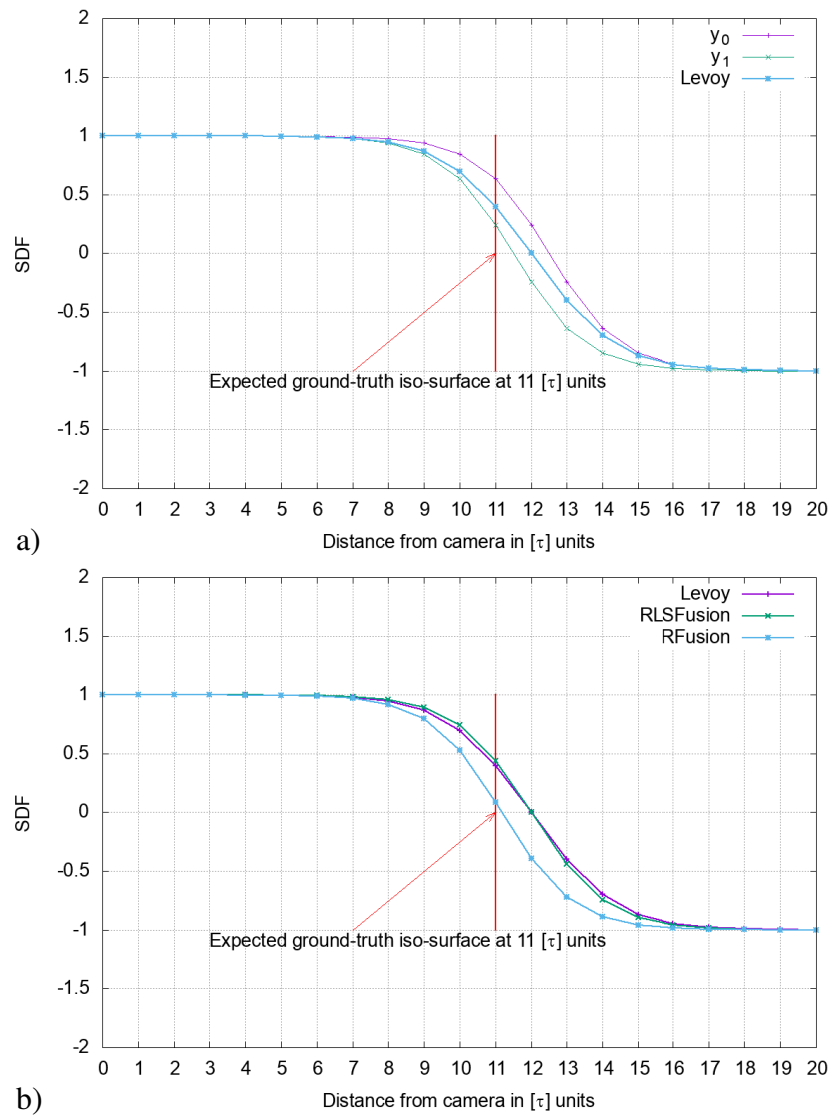


Figure 5.8: a) Erroneous depth measurements represented with SDF signals and b) Estimated SDF signal from traditional incremental methods and RFusion.

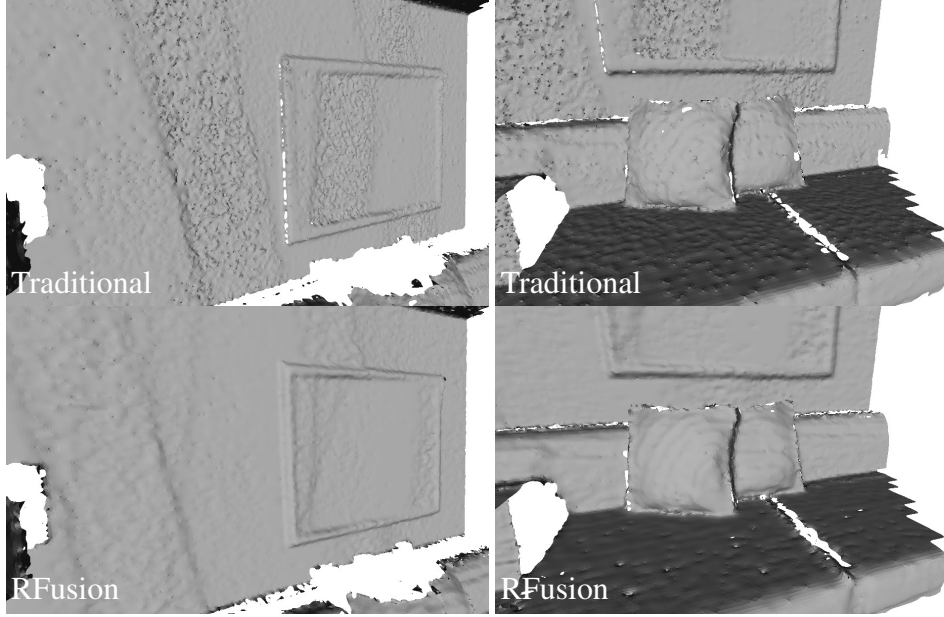


Figure 5.9: Comparison of traditional fusion (upper row) and proposed regularized fusion (bottom row) after fusing 10 depth images

In practice, the value of *thresh* is directly correlated with the sensing capabilities of the 3D sensor. Provided the value of *thresh* is selected appropriately, this pass identifies isolated and sparse depth outliers efficiently, however mis-labeling of cubes can occur due to corners or mis-alignment.

Therefore in a third pass, every potential cube $c \in C_k$ is tested on semantic basis (i.e. sufficient connectivity with active cubes) and the labels are either upgraded to active cubes in the case of validity criteria or dropped the entry from C_k altogether. Finally, all 3D points from finalized active cubes list are arranged in a memory array for the volumetric integration.

In order to demonstrate the working principle of SORF in pictorial form, consider a synthetic surface and corresponding 3D points with added outliers as shown in Figure 5.10.a. An empty local grid G_{local} having similar scale and transformation characteristics as the global volumetric grid G_{global} is initialized. This equivalence relation between both grids allows the proposed framework to initialize, access and modify each particular voxel-block without performing unnecessary conversions. In the first pass, each 3D point is registered and counted in respective cubes followed by labeling the cubes as either *active* or *potential* (shown by green or yellow blocks, respectively in Figure 5.10.b). It can be noticed that although most of the cubes are labeled correctly since they satisfy the counting threshold, cubes containing outliers

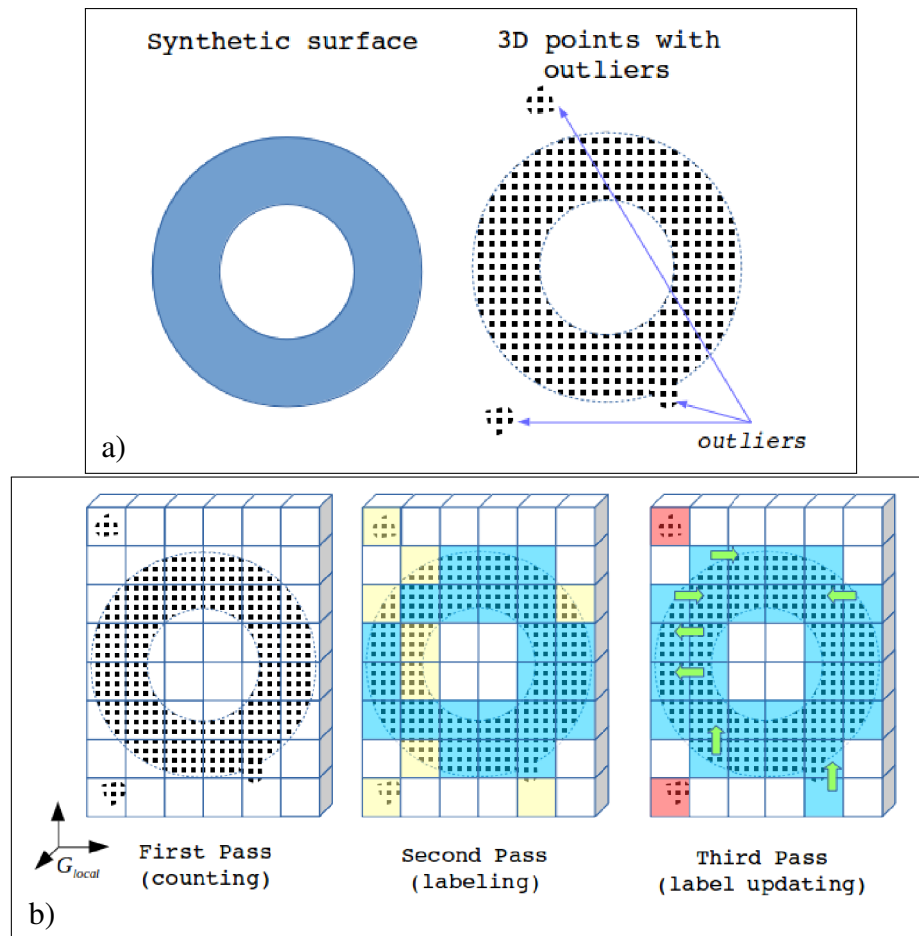


Figure 5.10: a) A synthetic surface and corresponding 3D points with additive outliers and b) Illustrated SORF passes.

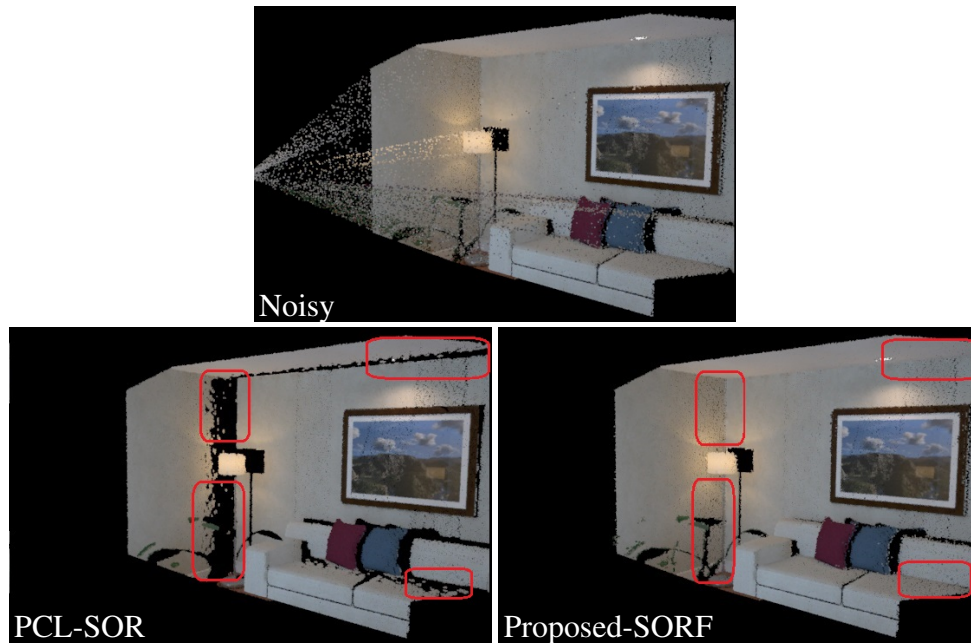


Figure 5.11: Comparison of outliers removal using proposed-SORF vs PCL-SOR

as well as which contain fewer 3D points due to misalignments are labeled as potential. In the final pass, each *potential* cube is tested for spatial connectivity with *active* cubes (this relation is shown with green arrow from potential cube towards active cube), this spatial connectivity ensures that isolated cubes are identified and removed from G_{local} . Finally, the list of active cubes coordinates is sent to the next processing stage where volumetric 3D integration combined with temporal depth updates removes the effects of non-isolated outliers.

A noisy depth image from ICL-NUIM's *living-room* dataset Handa et al. (2014) is selected and processed using statistical outliers removal from PCL Rusu et al. (2008) and the proposed SORF to compare processing time and effectiveness against outliers. It was found that the proposed SORF took 15ms to process a standard 640x480 depth image on commodity computer while the same image took around 600ms when processed with statistical outliers removal from PCL. This processing speed-up is due to linear nature of the proposed outliers removal and efficient use of spatial information. Furthermore, it is evident from the visual inspection of Figure 5.11 that PCL-SOR mis-treated vital 3D points and removed them. This phenomena can be observed in the highlighted regions where two or more surfaces are joining together. It is therefore expected that using the proposed-SORF in real-time applications will remove undesirable outliers *on-the-go*. This removal of outliers also affects the execution time

of an overall reconstruction pipeline in a positive direction.

It is worth mentioning that effectiveness of SORF can easily be visualized in qualitative evaluation (as shown in Figure 5.11), however quantitative evaluation is a non-trivial and tedious problem involving manual histogram equalization of absolute surface errors.

5.4 3D reconstruction framework

In order to integrate the proposed research contributions in the form of a 3D reconstruction pipeline, a modular design is preferred over traditional closed system in which components are strongly interlinked. This modular design enabled rapid prototype development and testing of incremental algorithmic updates without modifying the complete design of framework. Processing elements (or modules) are designed to utilize multi-threading aspects of modern CPU architectures to maximize the processing efficiency. For the sake of compactness, the upcoming text refers to the proposed reconstruction framework as *SmoothFusion*. In order to facilitate the working of SmoothFusion in both the off-line and on-line depth sensing and reconstruction scenarios, two variants have been developed to facilitate each problem scenario.

Both implemented variants share the core concepts of depth noise removal capabilities provided by RFusion and outliers removal by SORF. Figure 5.12 shows the block diagrams of all implemented variants of SmoothFusion.

In the on-line reconstruction scenario where live depth information is acquired with a simple depth sensor such as Kinect and Kinect v2, the *loader* module registers a time stamped depth and color image followed by sharing these images with the *Localization* module which tracks camera ego-motion with the help of state-of-the-art visualSLAM algorithm *ORB_SLAM2* developed by Mur-Artal and Tardós (2015). Since IPS is a sophisticated depth sensing and navigation system, the need of applying the *Localization* module for ego-motion tracking is redundant, therefore captured depth and color image streams can be used directly in upcoming processing modules. However in an off-line reconstruction scenario, localization information is acquired by applying *ORB_SLAM2* and the resulting trajectory along with depth and color images are stored on a secondary storage device in the standard format as suggested by Sturm et al. (2012).

Once the sensor is localized in the world coordinate system, the localization information

combined with depth and color image is considered to be a single processing entity referred to as *input-instance*. *Pre-processor* modules apply appropriate depth scaling and transform depth images into series of 3D points in the world coordinate system with the help of sensor pose information. Depth outliers are removed from acquired 3D points by applying proposed *Spatial Outliers Removal Filter*. Since the working principle of SORF involves the creation of axis-aligned voxel-grid containing *cubes*, the list of active cubes can be utilized in the *Fusion* module to create spatial voxel-blocks.

The *fusion* module uses a hashing function to determine the memory occupancy of each suspected voxel-block. In principal, inactive or temporally older voxel-blocks are swapped out from fast acting memory to save resources. Therefore, all active voxel-blocks are loaded and processed with a regularized least square estimator in a recursive fashion. In principle, each block is updated at a time, which eliminates the need of storing multiple copies of system variables such as the *gain-matrix* for each voxel-block. A dedicated *data structure* is designed to facilitate the storage of such information in an efficient manner.

The *renderer* module applies projective ray-casting to determine iso-surfaces within each voxel-block and resulting zero crossings are stored as an array of *vertex* containing spatial and associated color information. It was observed in the empirical evaluation phase that such vertex based representation allows real-time visualization. High-quality meshes can be generated using a standard marching cube algorithm from implicit representation. Since the mesh extraction step is computationally extensive, postponing this step until all the depth images are integrated produces hassle-free processing. Furthermore, implicit representation enables robotic applications to determine the surface of object by using signed distance information (as suggested in Section 2.1.3).

Figure 5.13 illustrates flexibility of SmoothFusion to handle various sensor types without changing the underlying pipeline. Consider a scenario in which depth images are encoded with non-linear depth encoding to facilitate multiple degrees of precision depending upon the distance of perceived object from the IPS sensor. Although this atypical encoding strategy is highly effective for encoding depth values from stereo based depth sensor, however state-of-the-art 3D reconstruction framework does not natively support such encoding. Contrarily, the *loader* module of SmoothFusion can be simply programmed to accept such atypical depth encoding as shown in Figure 5.14.a. Furthermore, the need of using external visualSLAM

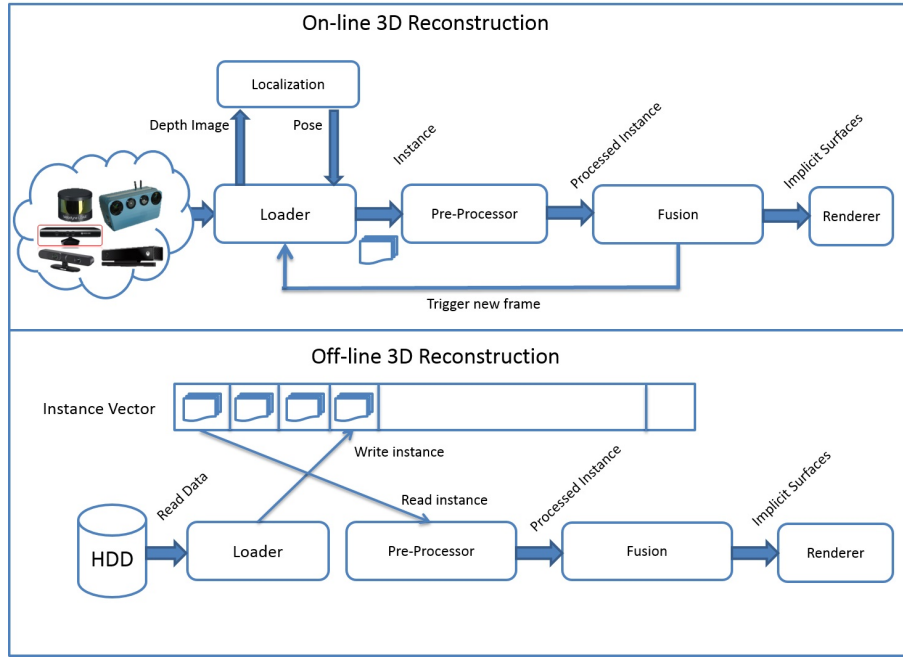


Figure 5.12: Block diagram of SmoothFusion with online and offline processing scenarios.

algorithm is not required since IPS sensor is capable of producing high-quality sensor pose by fusing IMU measurements with visual ego-motion.

Similar problematic scenario can be observed in the case of stereo camera system in which the depth image for each instance is not available. Unfortunately, current state-of-the-art 3D reconstruction frameworks do not facilitate direct color image pair captured from stereo camera system. Contrarily, the modular design aspect of SmoothFusion can be exploited in this scenario in which the *loader* module can easily be programmed to perform stereo matching and depth estimation on-the-go. Such robust profile of SmoothFusion is illustrated in Figure 5.14.b

5.5 Summary

This chapter presented a detailed introduction and analysis of proposed research contributions and evaluated the workings of each contribution against traditional weighted integration. Proposed contributions and their implementation in the form of 3D fusion and reconstruction are shown to efficiently handle sequences of depth and color images. The faster convergence of absolute surface errors produced by *RFusion* in error-prone depth samples is highlighted. In

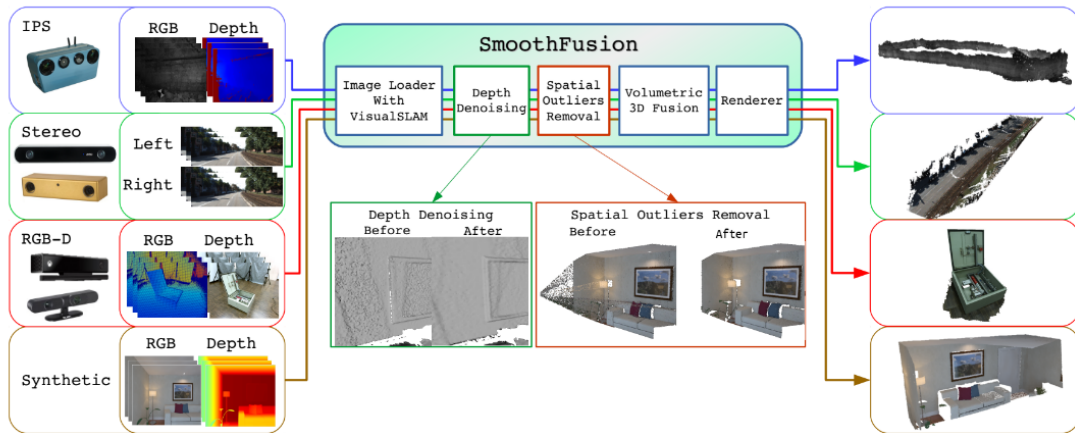


Figure 5.13: Modular design of SmoothFusion to handle multiple sensors and their respective 3D reconstructed models.

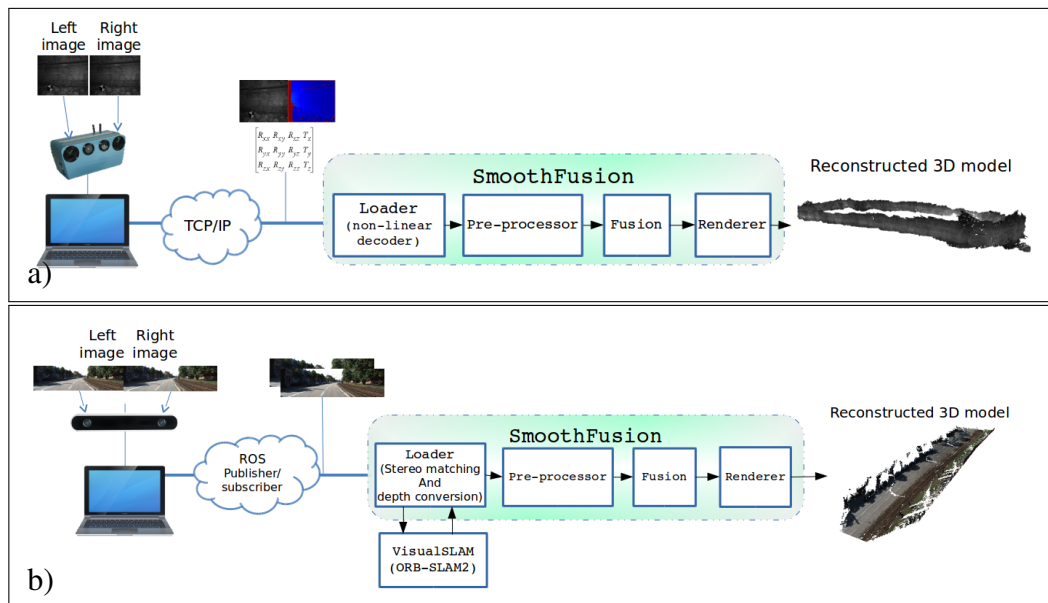


Figure 5.14: a) SmoothFusion with IPS module to use provided sensor pose b) SmoothFusion with stereo matching module combined with ORB-SLAM2 for real-time processing.

the upcoming chapter, the proposed reconstruction framework is evaluated against state-of-the-art methods with actual color and depth image sequences.

Chapter 6

Evaluation

This chapter provides a comparative evaluation of the proposed 3D reconstruction framework (*SmoothFusion*) against *InfiniTAM* and *FastFusion* which are considered to be state-of-the-art volumetric 3D modeling techniques (see Section 2.4). Section 4.3.1 presented the relation between the quality of the reconstructed 3D model and the number of acquired samples. In practice, restricting the movement and/or velocity of a sensor to a particular degree can seriously affect the performance of mobile robots. Therefore, this provided comparative analysis is intended to highlight the ability of the 3D reconstruction method to handle a high degree of depth noise and fast sensor movements. Furthermore, to achieve diversity in this empirical evaluation process while avoiding unnecessary repetitive results, one trajectory is selected from each dataset and acquired assessment results are presented in the form of figures.

The critical evaluation consists of three distinct elements: quantitative assessment (Section 6.1) which employs quality metrics introduced in Chapter 3, qualitative assessment (Section 6.2) which highlights the visual appearance of reconstructed 3D model and running-time analysis (Section 6.3) which emphasizes the applicability of the 3D reconstruction in real-time applications. The chapter is concluded in Section 6.4 where the findings are summarized to highlight the applicability of the proposed framework.

6.1 Quantitative evaluation

Since the ground truth 3D models and sensor trajectories are available for both ICL-NUIM and CoRBS datasets, it is possible to compute the deviation of reconstructed model against the

ground truth model by applying the quality metric introduced in Chapter 3.

The histograms in Figure 6.1.e and 6.3.g show the normalized error distribution from the reference ground truth model. The density function basically describes the relation between registered samples and distances from reference shapes. In principal, smaller overall distances produces a sharp peak which is located closer with respect to zero in the horizontal axes.

Figure 6.1.a-c demonstrate the registered absolute surface error for each sample represented with pseudo color coded heat map. It can be observed that lack of any outliers detection scheme in FastFusion produced undesirable samples which affect the visual appearance of the reconstructed model. If un-treated, such deformities directly influence the perception of mobile robotic applications. The error histogram presented in Figure 6.1.e shows that SmoothFusion produced smaller overall distances, this is achieved by employing regularized 3D fusion on erroneous depth samples. The cumulative error distribution plot in Figure 6.1.f reveals that approximately 90 percent of the registered samples resides within the range of 0.5cm when SmoothFusion is employed. Contrastively, the absolute surface registered in samples from FastFusion and InfiniTAM achieve 90 percent deviation mark at 0.8 cm and 1.0 cm respectively in Figure 6.1.f. Finally, Figure 6.1.g illustrates the respective median errors which summarize overall error classification into a single quantifiable value which shows that processing dataset with SmoothFusion produced comparatively lower surface errors. Similar observations can be recorded from 6.2 in which 3D reconstructed model from *ICL2* trajectory is presented.

CoRBS dataset trajectories are captured with a Kinect v2 depth sensor which utilizes time-of-flight depth sensing. The accumulated error in depth samples is therefore comparatively low. Precise depth information combined with short sensing distances allows high-quality depth images which produce realistic 3D models. Error histogram and cumulative error distribution plots in Figure 6.3.g and 6.3.h respectively show that all reconstruction methods performed adequately in terms of quantitative assessment.

Table 6.1 summarizes the quantitative evaluation performed on 3D models generated from all trajectories in *ICL-NUIM* RGBD dataset. In order to maintain an unbiased evaluation, the effects of outliers have also been excluded from all quantitative evaluation. It can be observed that the error metrics produced by SmoothFusion are comparatively lower for all four noisy sequences. This noise resistant property of SmoothFusion can play a significant role in processing highly erroneous depth information such as produced by IPS.

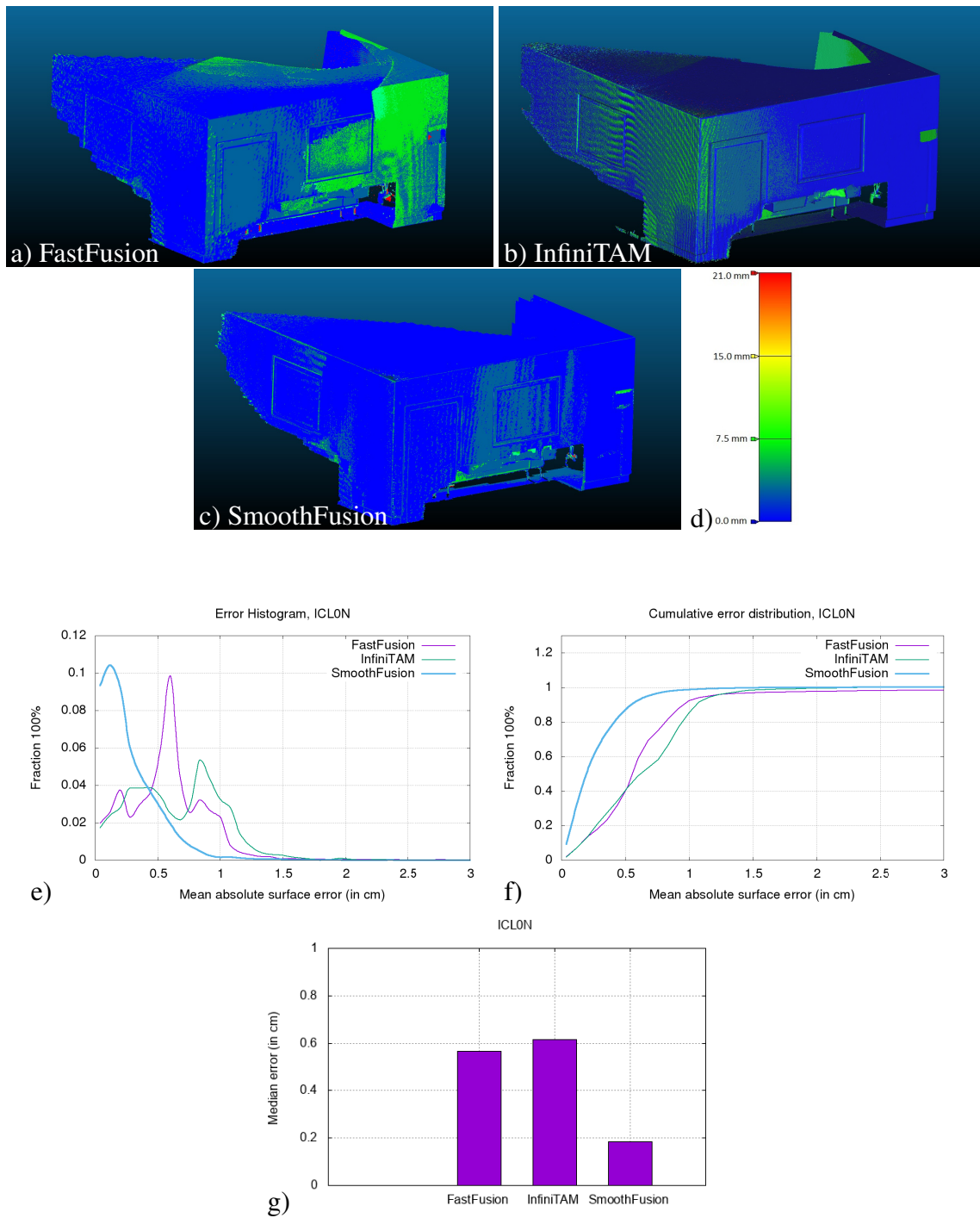


Figure 6.1: 3D reconstruction of noisy depth images from *ICL0* trajectory a-c) Pseudo color coded 3D samples representing absolute surface error from ground truth for the three methods InfiniTAM, FastFusion and SmoothFusion d) Color scale representing absolute surface error in a-c, e) Error histogram, f) Cumulative error distribution and g) Median error comparison.

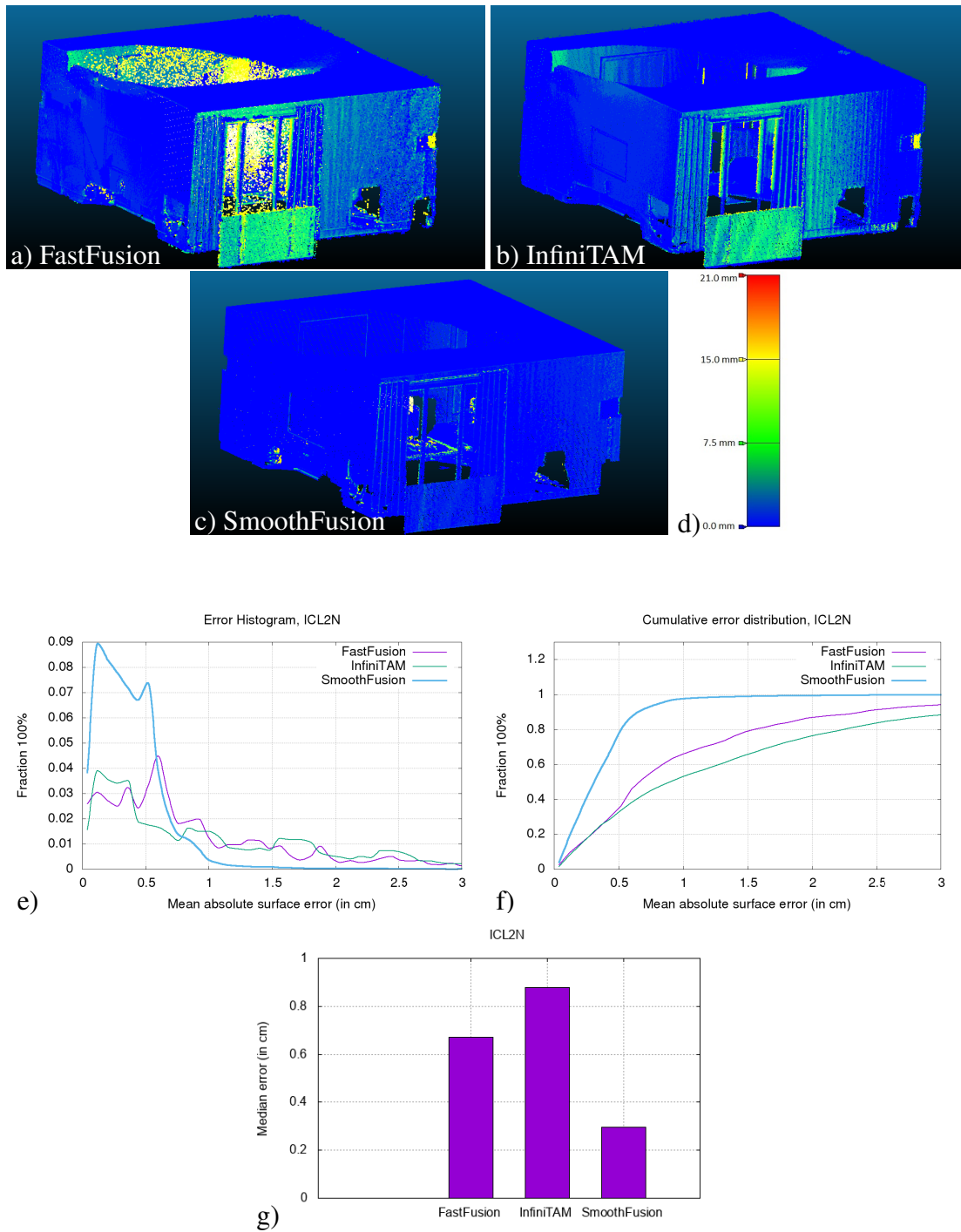


Figure 6.2: 3D reconstruction of noisy depth images from *ICL2* trajectory a-c) Pseudo color coded 3D samples representing absolute surface error from ground truth for the three methods InfiniTAM, FastFusion and SmoothFusion d) Color scale representing absolute surface error in a-c, e) Error histogram, f) Cumulative error distribution and g) Median error comparison.

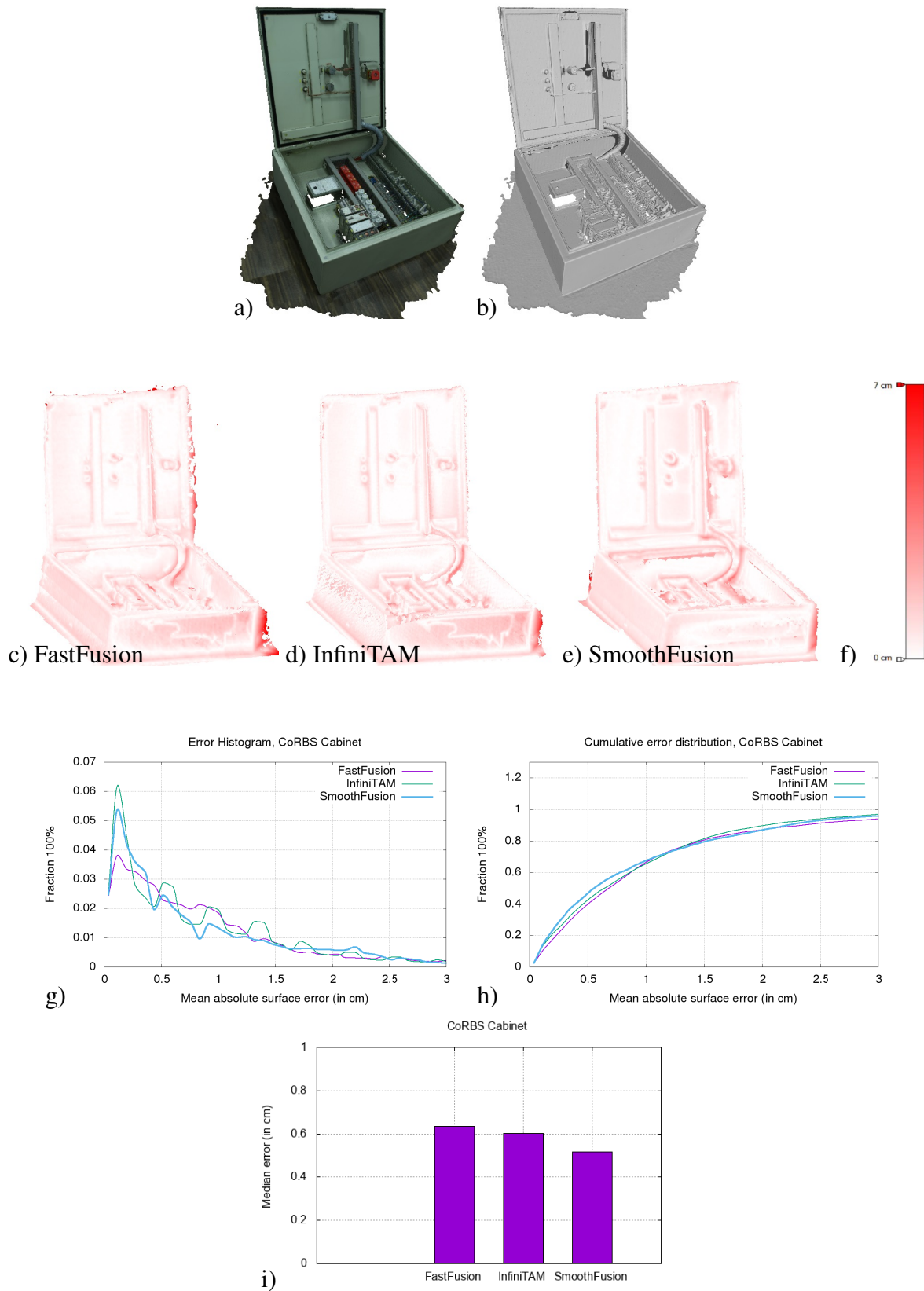


Figure 6.3: a) Textured ground-truth 3D model, b) color removed to highlight geometry, f) color scale representing absolute surface error in c,d,e g) error histogram, h) cumulative error distribution and i) median error comparison.

Table 6.1: Mean absolute surface error (in mm) for *living-room* dataset trajectories

Trajectory \ Method	InfiniTAM	FastFusion	SmoothFusion
Clean depth images			
LR0	2.067	2.7	2.13
LR1	1.117	1.652	1.844
LR2	11.651	10.61	1.944
LR3	1.766	1.922	1.981
Noisy depth images			
LR0	5.307	6.696	2.586
LR1	6.541	5.98	2.799
LR2	13.56	10.655	3.617
LR3	4.831	5.525	2.677

6.1.1 Outliers removal and memory utilization

In the empirical evaluation phase, it was observed that integration of outliers detection and removal mechanism (i.e. *SORF*) allowed the overall reconstruction framework to efficiently use of memory and computational resources. To demonstrate this behavior, 400 instances of left and right images from *KITTI-06* trajectory along with camera position were processed with and without using the *SORF* within the proposed reconstruction framework. The utilization of memory for both variants were recorded after processing each successive image-pair, these results are illustrated in Figure 6.4 where it is evident that the use of *SORF* conserves memory. Evidently, a marginal computational speed-up which is caused due to the removal of outliers was also recorded. Similar findings were recorded for *Mine* and *ICL2* trajectories and are presented in Figure 6.5 and Figure 6.6 respectively.

Although the quantifiable assessment provided insights on the ability of 3D reconstruction frameworks to handle depth noise, in practical applications however, the reference model is usually not presented. Therefore, visual appearance and smooth surfaces are given priority over quantifiable measures, such assessment is provided in the upcoming section.

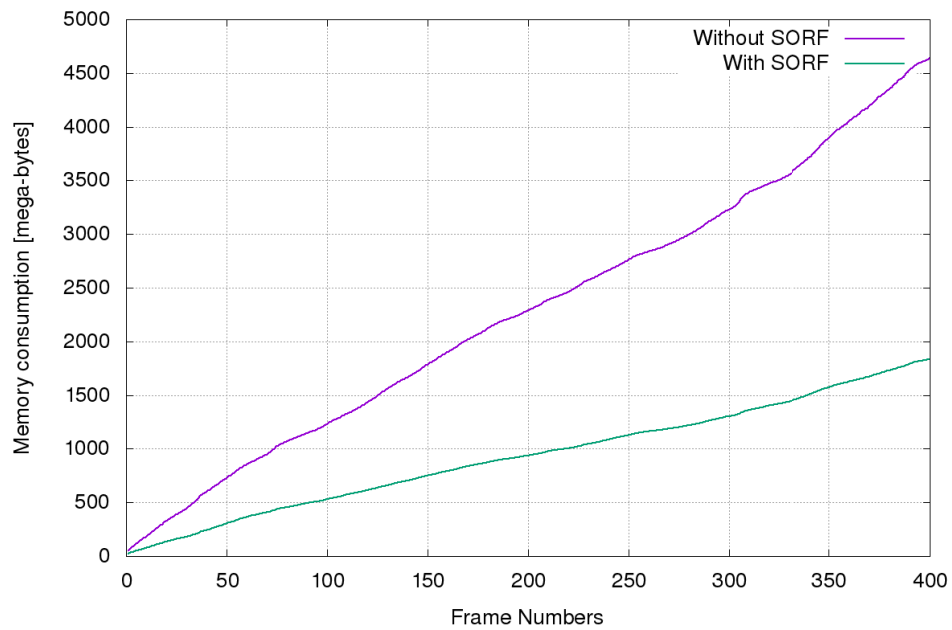


Figure 6.4: Per-frame memory consumption of the reconstruction framework for *KITTI-06* trajectory.

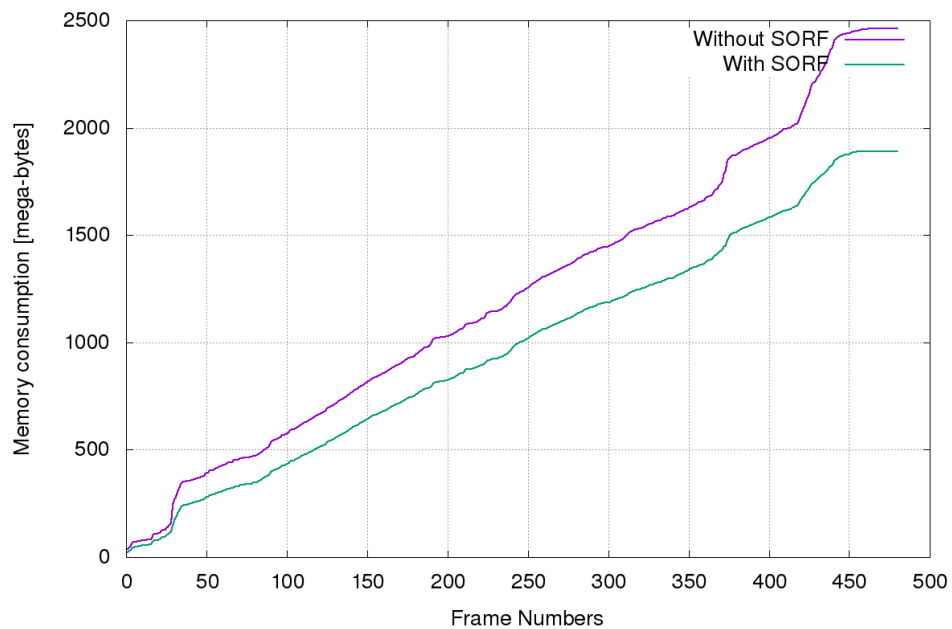


Figure 6.5: Per-frame memory consumption of the reconstruction framework for *mine* trajectory.

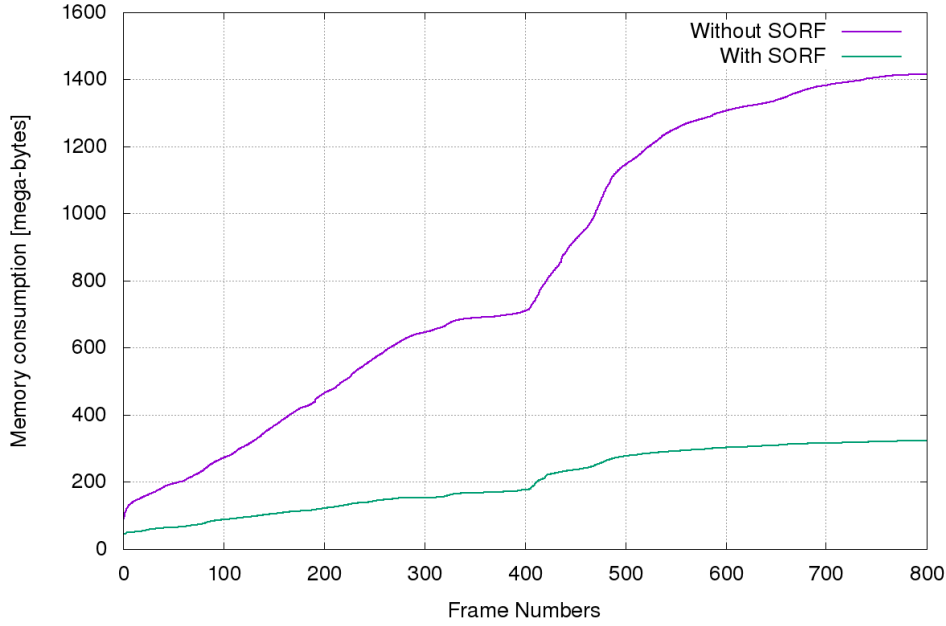


Figure 6.6: Per-frame memory consumption of the reconstruction framework for *ICL-2* trajectory.

6.2 Qualitative evaluation

In order to evaluate the quality of reconstructed 3D models using visual inspection, screenshots from identical viewing parameters are presented in this section to compare the visual aspects of 3D models. Following scenarios have been identified and used to compare the performance of the reconstruction framework:

1. **High variance noise:** Highly erroneous depth samples are prone to corrupt the implicit representation, this correlation between depth information and generated surface is explained in Section 6.1 where un-filtered noisy depth images from *ICL living-room* datasets influenced the mean absolute surface error in reconstructed shapes.
2. **Low sampling density surfaces:** The *convergence* property discussed in Chapter 4 presumes that sufficient depth samples are provided for a particular surface area to reduce the effects of depth noise. In case of large-scale reconstruction applications, gathering enough samples to utilize the convergence property becomes a bottleneck. In empirical evaluation of trajectories captured from IPS sensor, it was found that on average, each *voxel-block* is updated 2-3 times during the reconstruction phase. In such scenarios,

un-regularized implicit surfaces are prone to produce holes and/or undesirable surface deformities.

3. **Low curvature surfaces:** These surface segments are presented to establish a baseline quality performance among reconstruction frameworks.

In the upcoming text, aforementioned scenarios are referred with a corresponding number enclosed in circle. For example, Figure 6.8.a with the highlighted area ① shows a surface segment where the effects of high variance noise in 3D integration are highlighted. It can be observed that the reconstructed surface produced by SmoothFusion contains smooth surfaces even though the depth information is acquired from stereo based depth estimation. Similar results of noisy depth sample to surface can be observed in Figure 6.13 where SmoothFusion was able to produce a smoother surface while retaining the fine details in the reconstructed model. Contrarily, both InfiniTAM and FastFusion were unable to exploit the convergence property to reduce noise effects.

Figure 6.9 shows effects of a fast moving sensor which result in the situation where voxel-blocks suffer from low sampling density and produce noisy surfaces (area ②). It can be observed that while InfiniTAM struggles to render the surface from low confidence voxel-blocks, SmoothFusion applies regularization to achieve planar surfaces for the floor.

Planar and low curvature surfaces in Figure 6.10 are highlighted with ③ to emphasize that all reconstruction techniques were able to produced smooth or planar surfaces where the environment and/or sensing scenario is trivial. This visual comparison is provided to establish a baseline that the proposed regularized fusion is capable of producing highly detailed surfaces when accurate depth samples are provided. In such cases the value of λ can be initialized to a very small value or zero.

In order to highlight the generic nature of the proposed contribution, the regularization mechanism is implemented as a standalone image-based depth noise removal filter (referred to as Total Variation Regularization (TVR) in upcoming text) and was integrated with both InfiniTAM and FastFusion. It was observed that addition of TVR module enhanced the capabilities of both frameworks to handle depth noise in an efficient manner. Findings of such modified InfiniTAM and FastFusion are presented in Figure 6.7 where it can be observed that addition of TVR module produced smoother 3D surfaces compared to the original variants (see Rajput et al. (2018)).

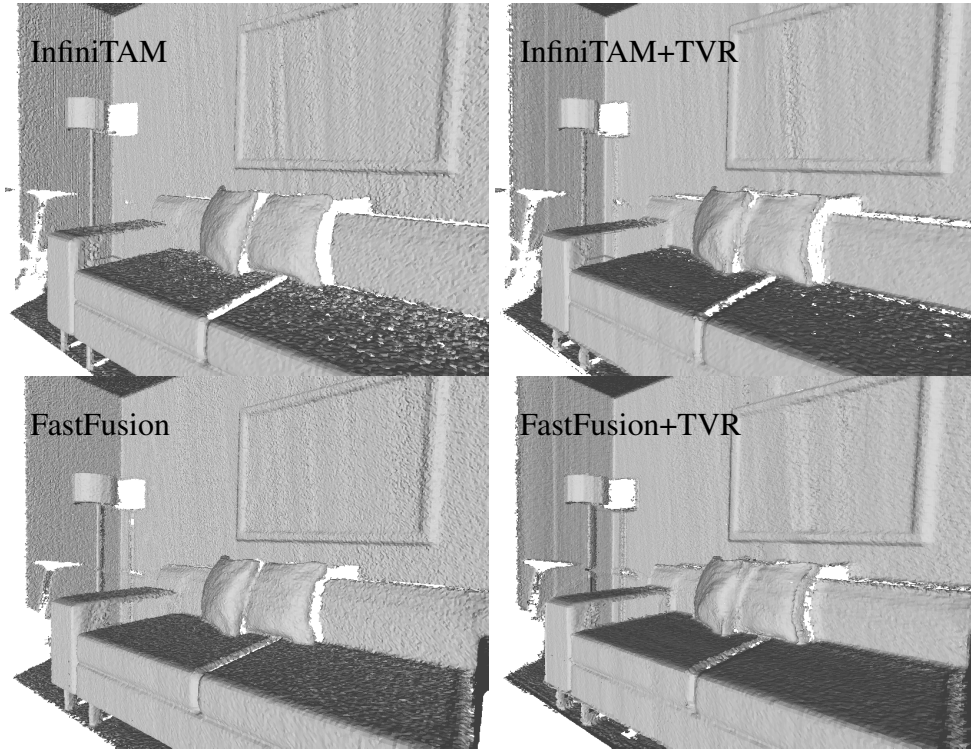


Figure 6.7: Effects of employing proposed-TVSR smoothing with InfiniTAM (upper row) and FastFusion (bottom row).

An experimental implementation of the proposed regularization framework which employs weighted integration of depth images acquired from stereo depth estimation and interpolated depth image from laser range data. Since existing state-of-the-art frameworks does not allow multi-sensor fusion, therefore a reconstructed model from an experimental multi-sensor integration system is compared against depth images acquired from stereo camera system. Figure 6.12 shows a significant improvement in terms of the quality of the reconstructed shapes.

6.3 Running time analysis

In order to assess the performance of the reconstruction framework in terms of execution time, a specialized variant of SmoothFusion is implemented which employs denoising on depth images directly and uses either *RLSFusion* or traditional weighted fusion as a 3D integration mechanism. Such implementation utilizes multi-threaded support of modern CPUs at maximum capability, actual implementation details of this hybrid design can be found in Rajput et al. (2018).

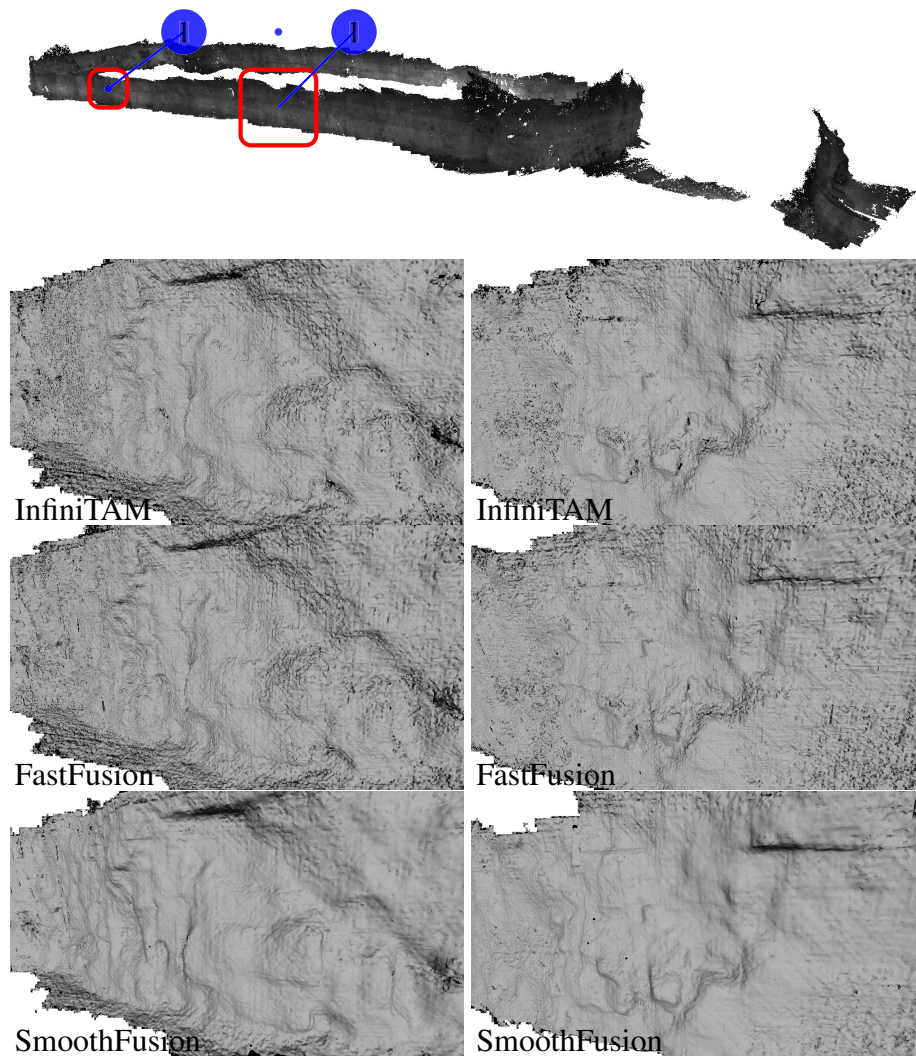


Figure 6.8: Reconstructed models from *mine* dataset captured with IPS

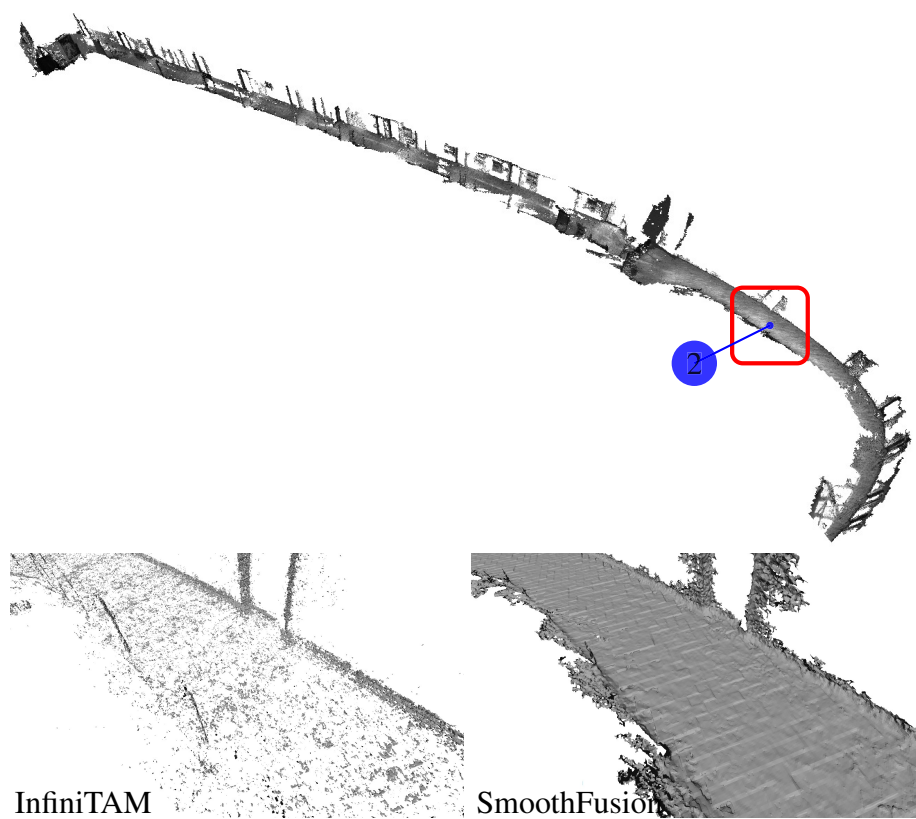


Figure 6.9: Reconstructed models from *corridor* dataset captured with IPS

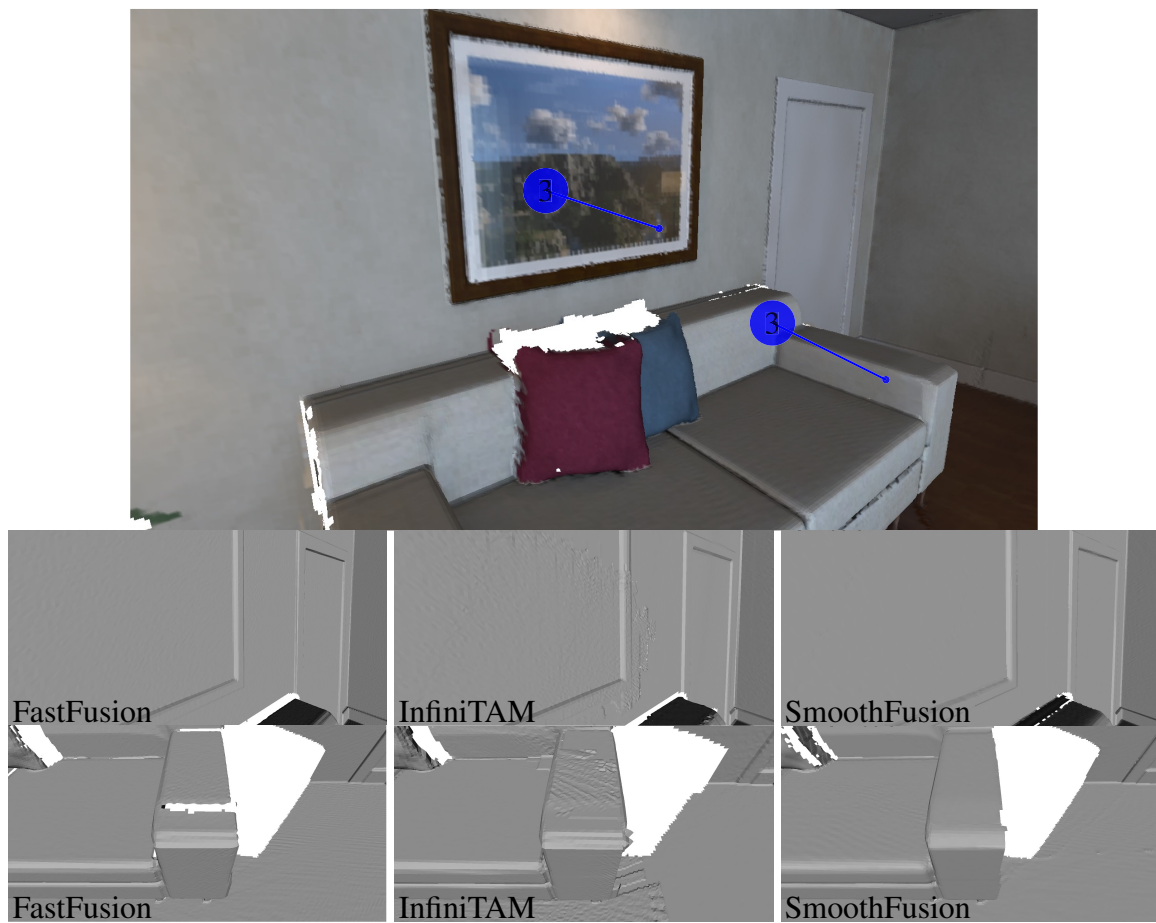


Figure 6.10: Reconstructed model from *LRO* trajectory with clean depth images

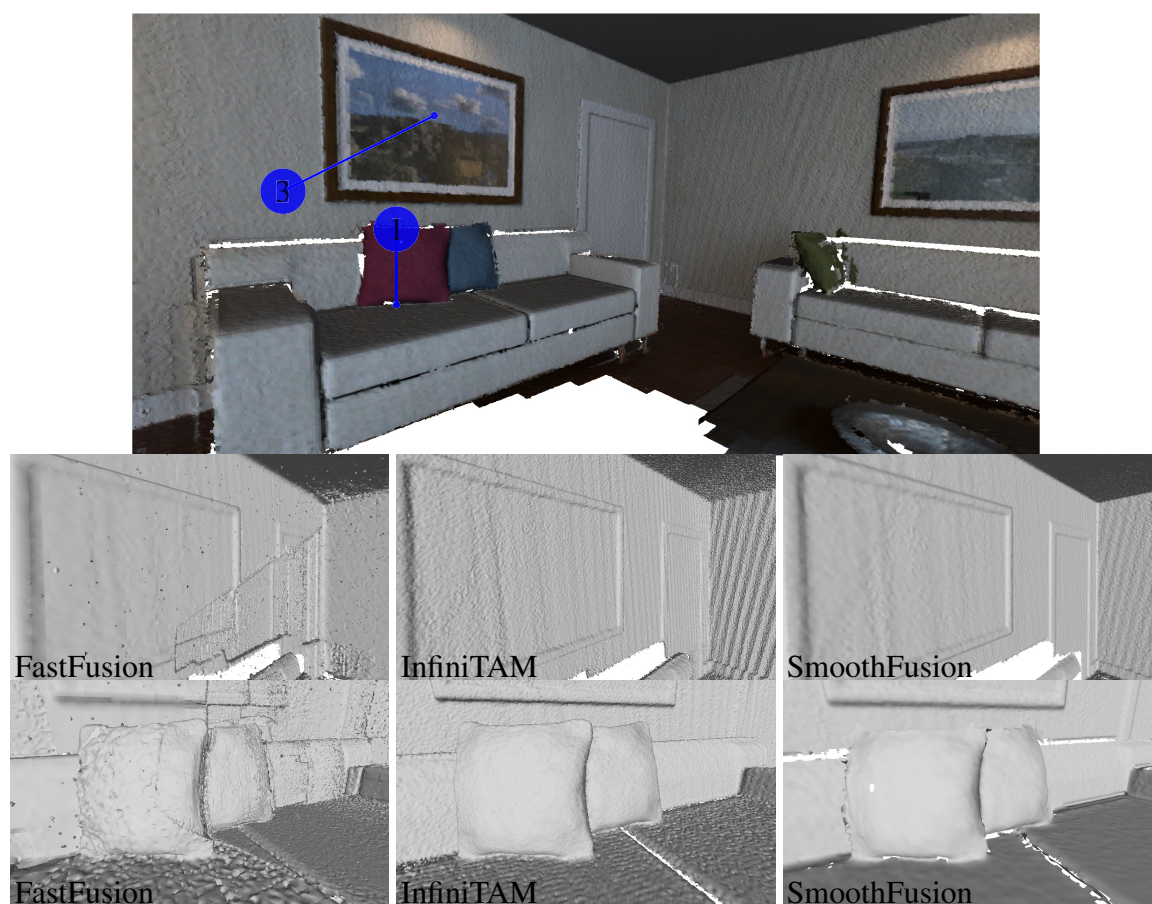


Figure 6.11: Reconstructed model from *LR2* trajectory with noisy depth images

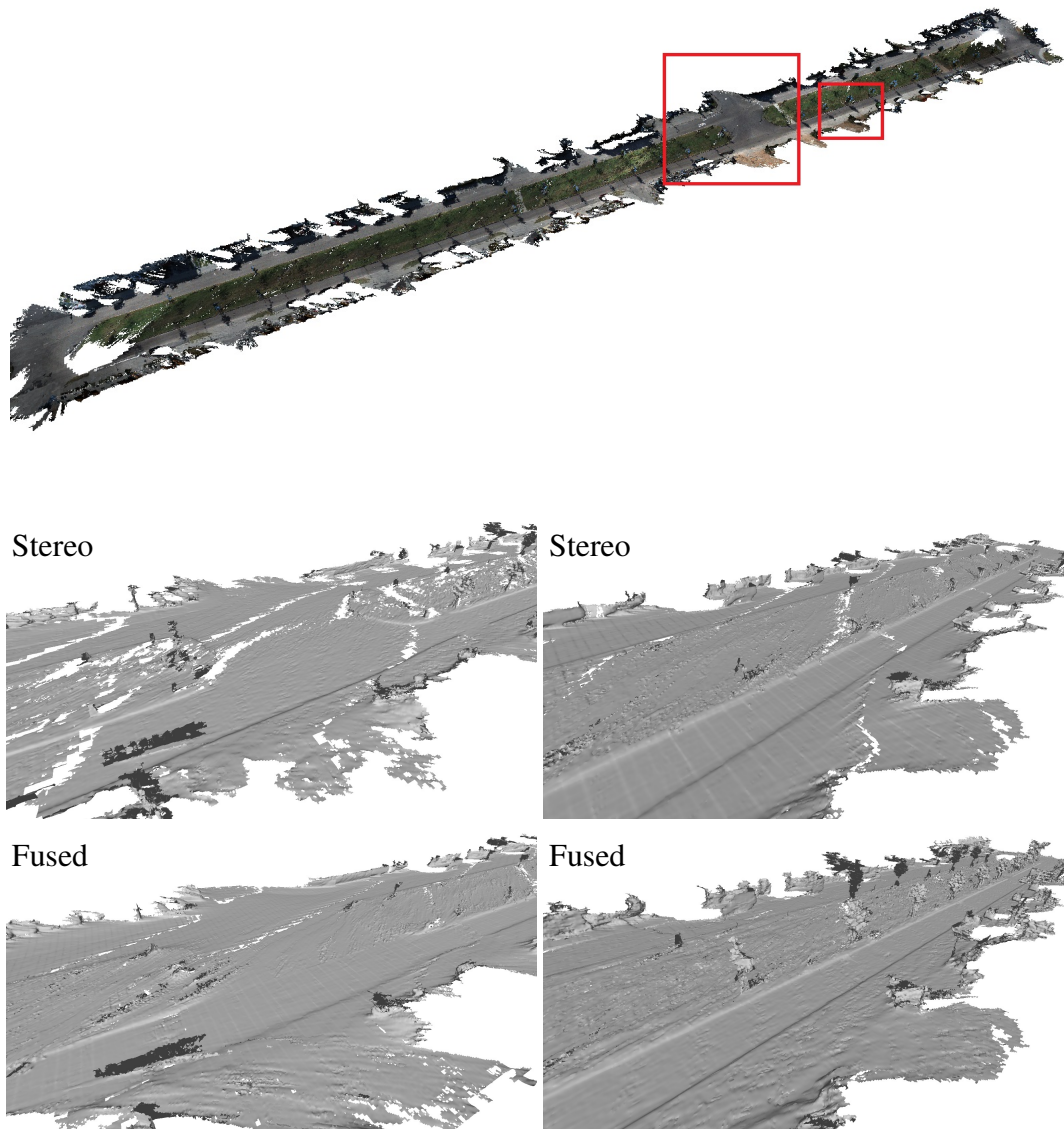


Figure 6.12: Reconstructed model from kitti dataset sequence 06 with two close-up screenshots.

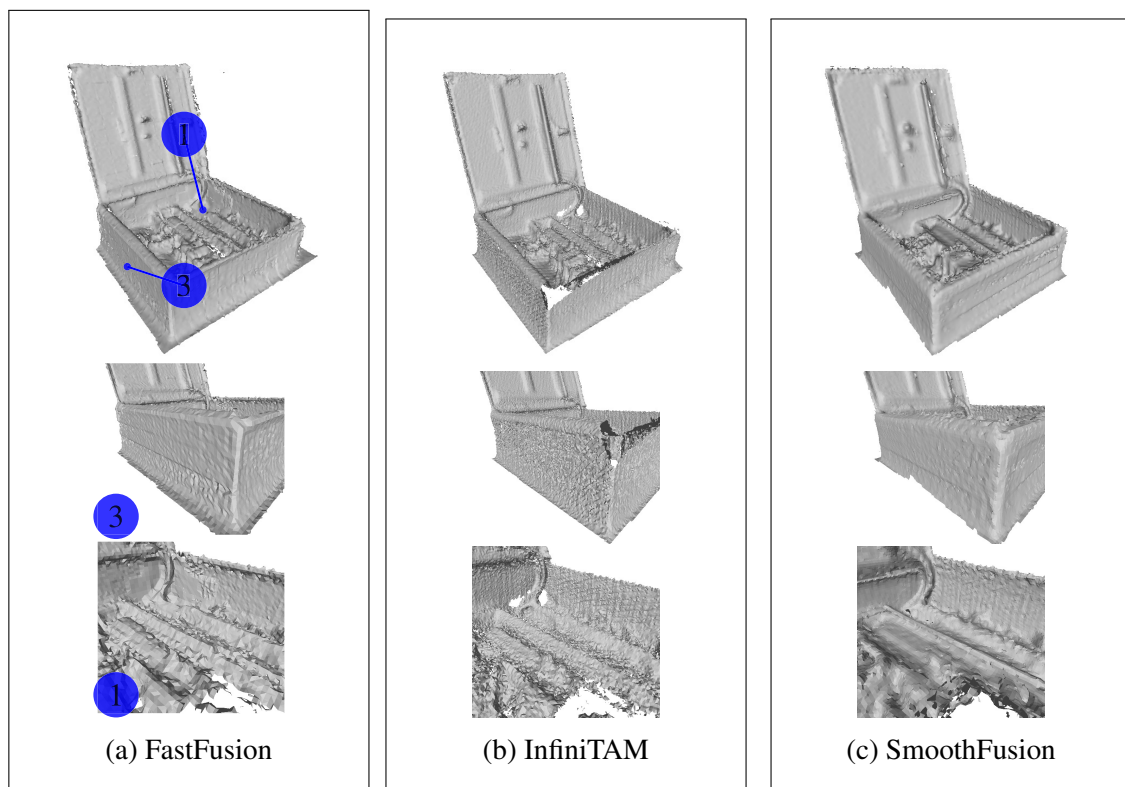


Figure 6.13: Reconstructed 3D model from *electric cabinet* trajectory from CoRBS dataset.

A timing subroutine which determines the execution time of integrating a single depth image into the volumetric integration in terms of CPU-cycles is employed to estimate execution time with milliseconds accuracy. Since source-code for both InfiniTAM and FastFusion is publicly available, an identical code for timing subroutine is utilized.

Figure 6.14.a and 6.14.b show the plot of time taken by each reconstruction method using *mine* and *ICL-LRI* trajectory to integrate a single instance of a depth and a color image¹. It was found that the processing time of FastFusion strongly correlates with the scale of reconstruction. Similarly, the application domain of InfiniTAM is limited since it is designed to utilize the processing capabilities of GPUs. On the contrary, it is evident from Figure 6.14 that the processing time of SmoothFusion is unaffected with the scale of reconstruction. Furthermore, a CPU based implementation allows mobile robot devices to utilize capabilities of SmoothFusion in real-time scenarios.

6.4 Summary

In this chapter, the performance of the proposed SmoothFusion is evaluated in terms of quantitative and qualitative performance metrics to justify the claim of employing regularization aspects of least square systems to reduce sample noise. Quantitative assessment compared the performance of reconstructed models in terms of low-level statistical quality measures such as median and mean. The assessment is then summarized in a high-level metric where error histograms and cumulative error distributions are employed to represent underlying statistical data. In cases where reference 3D models for comparison were not available, visual aspects of reconstructed models are compared and detailed screenshots are presented. Finally, a single frame execution time analysis is performed to highlight the real-time property of the proposed framework on mobile robotic applications.

¹InfiniTAM does not employ color information

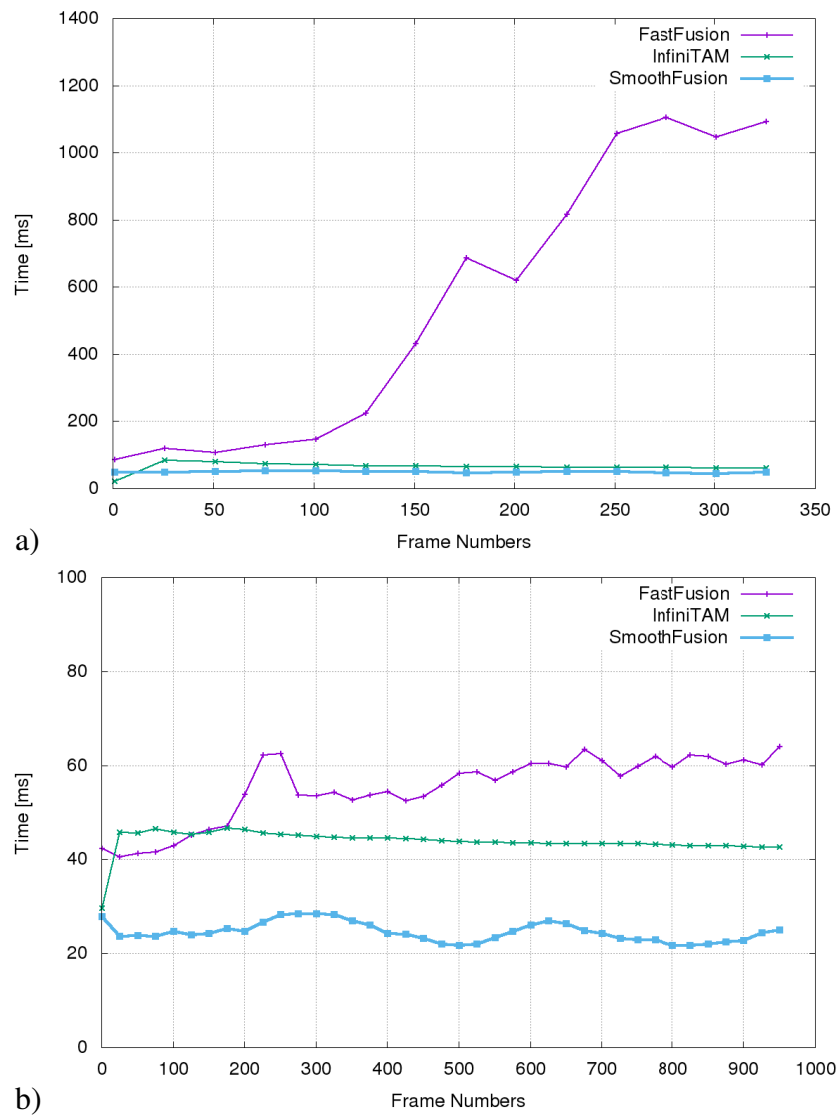


Figure 6.14: Per-frame processing time in large scale environment for *mine* dataset (a) and small scale synthetic environment *LRI* (b) (lower is better)

Chapter 7

Conclusion and Future work

Existing 3D fusion schemes which integrate depth information acquired from 3D sensors in an incremental fashion utilize traditional weighted fusion which was introduced in 1996 by Curless and Levoy (1996). Initially, the technique is proposed to integrate range images acquired from accurate depth sensors. However the general concept of weighted integration is still present as-is in state-of-the-art reconstruction frameworks. Since there is a variety of depth sensors available, there exists a strong correlation between accuracy of perceived depth's and the cost of the sensing unit. Therefore, utilizing a integration system which was originally proposed for range images on low accuracy depth images is prone to either produce undesirable surface deformities in reconstructed models or restricts the sensor movement to achieve multiple depth samples to estimate the depth information.

In Chapter 2, fundamental problems of error-prone 3D samples and their effects in reconstructed models are introduced and the rationale behind the use of *a priori* information is provided to produce smooth and life-like surfaces in reconstructed 3D models. It was concluded that employing smoothness constraints at the time of approximation can reduce noise effects, however existing 3D shape estimation methods capable of employing *a priori* smoothness information on 3D samples does not allow incremental updates to the underlying reconstructed model. Contrarily, existing incremental 3D volumetric fusion techniques are not designed to support the concept of prior smoothing.

This thesis presents a novel 3D fusion framework specifically tailored to address the fundamental question of 3D shape reconstruction from error-prone depth information by the integration of *prior* smoothing constraints. In principal, it is intuitive to presume that contours

of reconstructed surfaces follow planar properties. This assumption is analogous to the fact that on smaller intervals the derivative of a continuously changing function can be approximated with a straight line.

It is worth mentioning that although the proposed system is capable of producing high quality 3D models, however the effects of localization can still effect the reconstructed models. In order to handle the erroneous effects caused by localization error, an on-line version of the proposed framework is created which creates two copies of global voxel grid (referred as primary and secondary grids). The secondary voxel grid (which is not shown to user) is updated once bundle adjustment is done using previously saved depth and color images. Once the updation process is finished on secondary grid, the primary grid is swapped with secondary grid followed by rendering stage. However, it is postulated that an another representation method (such as point-based representation) can also be used to make this process more robust.

Chapter 4 presented the in-depth analysis of traditional weighted integration methods and highlighted potential areas of improvement. Chapter 5 provided novel contributions specifically designed to handle erroneous 3D samples and depth outliers with the help of a novel outliers removal filter and regularized 3D fusion system respectively. The proposed research contributions are implemented in the form of a reconstruction framework and its comprehensive evaluation is performed in terms of a quantitative and qualitative assessment, subsets of findings are provided in Chapter 6.

The proposed 3D reconstruction framework in this thesis makes three original scientific contributions to the computer vision and robotics field:

1. The most significant contribution is the novel least square estimation based 3D integration system capable of employing regularization as a smoothing constraint to handle erroneous depth information.
2. A novel recursive formulation of a regularized 3D fusion estimator which approximates second order differences among neighbouring implicit voxels to reduce total variation and produce a smoother reconstructed model.
3. A robust spatial outliers removal filter (SORF) having a linear complexity ($O(n)$) capable of removing 3D outliers in real-time.

Robotic applications which involve spatial perception and understanding can utilize aforesaid contributions to reconstruct high-quality 3D shapes. Furthermore, generic aspects of the proposed framework combined with a robust computational profile allows further flexibility in selection of the depth sensor. It is therefore expected that the presented research will provide a positive addition in low-cost robotic applications.

7.1 Future Directions

The generic nature and a robust computational profile of the proposed 3D reconstruction framework allow the usability in numerous active robotic applications. The following sections outline some suggestions for integrating this research to cutting edge research in computer vision and robotics.

7.1.1 Adaptive depth denoising

In an empirical evaluation, it was observed that although applying *prior* smoothing constraints in the form of λ produced fruitful outcomes and controlling the λ parameter with gain can produce significant improvements. These temporal updates of a regularization parameter do not accommodate non-linear depth noise. Therefore, an implementation of RLSFusion which accepts a noise function as input parameter instead of a single value at the time of execution will allow the framework to handle depth noise efficiently and accelerate the *convergence* of the absolute surface error.

7.1.2 Automated scene understating

Since the underlying representation of the proposed framework is in the form of semi-dense voxel blocks with implicit values, computer vision and robotic applications specially those which employ Signed Distance Function as representation can benefit from the smooth implicit representation. Notably, 2D and 3D visualSLAM algorithms (Fossel et al. (2015) and Canelhas et al. (2013)) which utilize SDF representation to assist localization estimation can use regularized implicit surface to enhance the accuracy of localization. As shown in Chapter 6 that a new outliers removal filter combined with total variation denoising are capable of producing smooth implicit surfaces. Such continuous representation can further enhance the

localization process.

Furthermore, smooth 3D surfaces acquired from the regularized fusion framework has the potential in assisting the perception phase of miniature mobile robots in an autonomous scenario. Accurate surface boundaries of obstacles acquired from the proposed framework can be used to calculate accurate distances between a particular object and the robot.

7.1.3 Efficient data structure for large environments

The proposed framework is designed to handle large scale environments with the help of hashed voxel-blocks in which temporally non-active blocks are swapped out from fast acting memory to accommodate latest updates. However such swapping can become a bottleneck in scenarios in which the sensor is mounted on a robotic vehicle and velocity and/or trajectory of the robot cause repetitive memory swapping. Therefore, an efficient data structure or representation technique is required which can reduce memory bandwidth and storage. Steinbruecker et al. (2014) implemented the concept of *incremental meshing* in which voxel blocks are represented as polygonal meshes to reduce the memory foot-print, however such conversion is inherently computationally expensive and degrades the real-time performance of 3D fusion approaches.

Bibliography

- Agoston, M. K. and Agoston, M. K. (2005). *Computer graphics and geometric modeling*, volume 1. Springer.
- Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. (2001). Point set surfaces. In *Proceedings of the Conference on Visualization '01*, VIS '01, pages 21–28, Washington, DC, USA. IEEE Computer Society.
- Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. (2003). Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1):3–15.
- Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., and Maierhofer, S. (2013). O-snap: Optimization-based snapping for modeling architecture. *ACM Transactions on Graphics (TOG)*, 32(1):6.
- Avron, H., Sharf, A., Greif, C., and Cohen-Or, D. (2010). L1-sparse reconstruction of sharp point set surfaces. *ACM Transactions on Graphics (TOG)*, 29(5):135.
- Balzer, J. and Soatto, S. (2013). Second-order shape optimization for geometric inverse problems in vision. *arXiv preprint arXiv:1311.2626*.
- Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., and Silva, C. T. (2013). A benchmark for surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(2):20.
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359.

- Berner, A., Wand, M., Mitra, N. J., Mewes, D., and Seidel, H.-P. (2011). Shape analysis with subspace symmetries. In *Computer Graphics Forum*, volume 30, pages 277–286. Wiley Online Library.
- Bodenmueller, T. (2009). *Streaming surface reconstruction from real time 3D measurements*. PhD thesis, Technische Universität München.
- Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239.
- Bridson, R., Marino, S., and Fedkiw, R. (2003). Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 28–36. Eurographics Association.
- Calakli, F. and Taubin, G. (2011). Ssd: Smooth signed distance surface reconstruction. In *Computer Graphics Forum*, volume 30, pages 1993–2002. Wiley Online Library.
- Canelhas, D. R., Stoyanov, T., and Lilienthal, A. J. (2013). Sdf tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3671–3676. IEEE.
- Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM.
- Cazals, F. and Giesen, J. (2004). *Delaunay triangulation based surface reconstruction: ideas and algorithms*. PhD thesis, INRIA.
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. (2008). MeshLab: an Open-Source Mesh Processing Tool. In Scarano, V., Chiara, R. D., and Erra, U., editors, *Eurographics Italian Chapter Conference*. The Eurographics Association.
- Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96*, pages 303–312, New York, NY, USA. ACM.

- Edelsbrunner, H. and Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72.
- Floater, M. S. and Hormann, K. (2005). Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*, pages 157–186. Springer.
- Fossel, J.-D., Tuyls, K., and Sturm, J. (2015). 2d-sdf-slam: A signed distance function based slam frontend for laser scanners. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1949–1955. IEEE.
- Funk, E. and Börner, A. (2016). Infinite, sparse 3d modelling volumes. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics*, pages 593–605. Springer.
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.
- Girardeau-Montaut, D. (2015). Cloud compare, 3d point cloud and mesh processing software. *Open Source Project*.
- Graber, G., Balzer, J., Soatto, S., and Pock, T. (2015). Efficient minimal-surface regularization of perspective depth maps in variational stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–520.
- Grießbach, D., Baumbach, D., and Zuev, S. (2014). Stereo-vision-aided inertial navigation for unknown indoor and outdoor environments. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 709–716. IEEE.
- Guendelman, E., Bridson, R., and Fedkiw, R. (2003). Nonconvex rigid bodies with stacking. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 871–878. ACM.
- Handa, A., Whelan, T., McDonald, J., and Davison, A. J. (2014). A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1524–1531. IEEE.
- Hart, J. C. (1996). Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545.

- Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE.
- Hoff III, K. E., Keyser, J., Lin, M., Manocha, D., and Culver, T. (1999). Fast computation of generalized voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 277–286. ACM Press/Addison-Wesley Publishing Co.
- Hornung, A. and Kobbelt, L. (2006). Robust reconstruction of watertight 3 d models from non-uniformly sampled point clouds without normal information. In *Symposium on geometry processing*, pages 41–50.
- Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al. (2011). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM.
- Kähler, O., Prisacariu, V. A., Ren, C. Y., Sun, X., Torr, P. H. S., and Murray, D. W. (2015). Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Transactions on Visualization and Computer Graphics (Proceedings International Symposium on Mixed and Augmented Reality 2015)*, 22(11).
- Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):29.
- Kazhdan, M. M. et al. (2005). Reconstruction of solid models from oriented point sets. In *Symposium on Geometry Processing*, pages 73–82.
- Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., and Kolb, A. (2013). Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *3DTV-Conference, 2013 International Conference on*, pages 1–8. IEEE.
- Kolluri, R. (2008). Provably good moving least squares. *ACM Transactions on Algorithms (TALG)*, 4(2):18.

- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM.
- Manson, J., Petrova, G., and Schaefer, S. (2008). Streaming surface reconstruction using wavelets. In *Computer Graphics Forum*, volume 27, pages 1411–1420. Wiley Online Library.
- Mur-Artal, Raúl, M. J. M. M. and Tardós, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE.
- Nguyen, C. V., Izadi, S., and Lovell, D. (2012). Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 524–530. IEEE.
- Nießner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. (2013). Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169.
- Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., and Seidel, H.-P. (2005). Multi-level partition of unity implicits. In *Acm Siggraph 2005 Courses*, page 173. ACM.
- OpenKinect.org (2018). Kinect depth ROS. https://openkinect.org/wiki/Imaging_Information.
- Öztireli, A. C., Guennebaud, G., and Gross, M. (2009). Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, volume 28, pages 493–501. Wiley Online Library.
- Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H., and Guibas, L. J. (2008). Discovering structural regularity in 3d geometry. In *ACM transactions on graphics (TOG)*, volume 27, page 43. ACM.

- Rajput, A., Funk, E., Börner, A., and Hellwich, O. (2018). A regularized volumetric fusion framework for large-scale 3d reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 141:124–136.
- Rajput, M. A. A., Funk, E., Börner, A., and Hellwich, O. (2016). Recursive total variation filtering based 3d fusion. In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications - Volume 5: SIGMAP*, pages 72–80.
- Ranftl, R., Gehrig, S., Pock, T., and Bischof, H. (2012). Pushing the limits of stereo using variational stereo estimation. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 401–407. IEEE.
- Roth, H. and Vona, M. (2012). Moving volume kinectfusion. In *BMVC*, volume 20, pages 1–11.
- Rusinkiewicz, S. and Levoy, M. (2000). Qsplat: A multiresolution point rendering system for large meshes. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 343–352. ACM Press/Addison-Wesley Publishing Co.
- Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941.
- Schreiner, J., Asirvatham, A., Praun, E., and Hoppe, H. (2004). Inter-surface mapping. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 870–877. ACM.
- Sharf, A., Lewiner, T., Shklarski, G., Toledo, S., and Cohen-Or, D. (2007). Interactive topology-aware surface reconstruction. *ACM Transactions on Graphics (TOG)*, 26(3):43.
- Sheffer, A., Praun, E., Rose, K., et al. (2007). Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2):105–171.
- Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.
- Steinbruecker, F., Sturm, J., and Cremers, D. (2014). Volumetric 3d mapping in real-time on a cpu. In *Int. Conf. on Robotics and Automation*, Hongkong, China.

- Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., and Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. Ieee.
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*.
- Tufte, E. R. (1990). *Envisioning information*. Graphics press.
- Walder, C., Schölkopf, B., and Chapelle, O. (2006). Implicit surface modelling with a globally regularised basis of compact support. In *Computer Graphics Forum*, volume 25, pages 635–644. Wiley Online Library.
- Wang, Y. and Feng, H.-Y. (2015). Outlier detection for scanned point clouds using majority voting. *Computer-Aided Design*, 62:31–43.
- Wasenmüller, O., Meyer, M., and Stricker, D. (2016). Corbs, comprehensive rgb-d benchmark for slam using kinect v2. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, volume ., page . IEEE.
- Whelan, T. (2018). ICPCUDA. <https://github.com/mp3guy/ICPCUDA>.
- Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. (2012). Kintinuous: Spatially extended kinectfusion.
- Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., and Davison, A. (2015). Elasticfusion: Dense slam without a pose graph. *Robotics: Science and Systems*.
- Yingze Bao, S., Chandraker, M., Lin, Y., and Savarese, S. (2013). Dense object reconstruction with semantic priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1264–1271.
- Zhang, W., Zhang, Y., Bai, X., Liu, J., Zeng, D., and Qiu, T. (2016). A robust fuzzy tree method with outlier detection for combustion models and optimization. *Chemometrics and Intelligent Laboratory Systems*, 158:130–137.

- Zhao, M., Tan, F., Fu, C.-W., Tang, C.-K., Cai, J., and Cham, T. J. (2013). High-quality kinect depth filtering for real-time 3d telepresence. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE.
- Zwicker, M. B., Pfister, H., and Gross, M. H. (2003). Visibility splatting and image reconstruction for surface elements. US Patent 6,639,597.

List of Figures

1.1	a) Laser depth sensor (LiDAR) mounted on top of autonomous vehicle, b) 3D points acquired from LiDAR, c) Stereo camera based IPS system, d) Color-coded depth map from IPS and e) Reconstructed 3D model of <i>mine</i> using stream of depth images with highlighted surface deformities.	2
1.2	Incremental 3D fusion and reconstruction process.	3
1.3	Proposed 3D reconstruction framework	5
2.1	The α -shapes algorithm. a) Input samples, b-d) reconstruction with increasing α Edelsbrunner and Mücke (1994).	11
2.2	a) Smooth surface model via NURBS. b) A set of parametric shapes combined to a global consistent surface. Schreiner et al. (2004).	12
2.3	A Signed Distance Function (SDF) on a fine grid.	13
2.4	Surface splatting of a scan of a human face, textured terrain, and a complex point-sampled object with semi-transparent surfaces. (Zwicker et al. (2003)) . .	14
2.5	Detecting repetitive structures (a) enables hole filling (b) in structured environments. (Pauly et al. (2008)).	15
2.6	Similar objects are scaled to match repetitive patterns. Red: strong deformations. Green: small deformations. (Berner et al. (2011)).	15
2.7	Learning priors from images for reconstruction (Yingze Bao et al. (2013)). . . .	16
2.8	Local surface approximation by Alexa et al. (2001).	17
2.9	Smoothness of point-to-plane blending controlled by ϵ by Kolluri (2008). . . .	18
2.10	a) Sphere fitting from sparse samples Öztireli et al. (2009) b) MLS without and with outliers Ohtake et al. (2005).	18
2.11	The implicit function $f(x)$ approximating surface points.	19

2.12	<i>Narrow-band</i> voxel grid around samples for graph-cut, Hornung and Kobbelt (2006).	20
2.13	A smooth and global implicit shape extracted via radial basis functions, Walder et al. (2006).	21
2.14	Surface reconstruction using SSDF by Calakli and Taubin (2011).	21
2.15	Surface normals from raw depth image (left) vs smooth depth image (right). Canelhas et al. (2013)	22
2.16	1 st row: 3D surface from Raw Kinect depth image, 2 nd row: Using bilateral smoothing and 3 rd row: Smoothing with Zhao et al. (2013)	23
2.17	a) Ground truth surface, b) Total variation regularization and c) Minimal-surface regularization by Graber et al. (2015).	24
2.18	a) A range surface along x-axis from sensor position b) two range-surface are integrated to form a new zero crossing.	25
2.19	a and b) SDF values from multiple views in cross-section of volumetric grid c) integrated SDF values to form compound iso-surface Curless and Levoy (1996).	26
2.20	a) Slice of SDF volume demonstrating potential truncation mechanism and b) overall 3D volume (Newcombe et al. (2011)).	26
2.21	Camera tracking information visualized around region of interest (left) and reconstructed 3D model (right) (Newcombe et al. (2011)))	27
2.22	Initial TSDF volume (left), updated TSDF volume after integration (middle) and movement tracking information (right) (Roth and Vona (2012)))	27
2.23	Large-scale 3D reconstructed model of apartment from <i>Kintinous</i> (Whelan et al. (2012))	28
2.24	Streaming in/out process of voxel blocks from Nießner et al. (2013)	29
2.25	Typical point cloud with highlighted depth outliers	30
2.26	Statistical measures to detect outliers	31
3.1	Framework applied on RGB-D image stream	38
3.2	The process of acquiring quantitative measures among reconstructed and ground truth 3D models	40

3.3	Process of calculating error distance between sampled ground truth 3D points and the reconstructed model and resulting an absolute surface errors in a color coded error map.	43
3.4	Typical error histogram (left) and cumulative error distribution (right).	43
3.5	Synthetic 3D and 2D function represented as point cloud followed by respective implicit representation with and without added depth noise.	44
3.6	The interior of a synthetic living room scene (color removed to highlight geometry)	45
3.7	Arbitrary sampled instances of registered color and depth images from IPS sensor	47
3.8	Light pattern projection based range image acquisition using two cameras (Wasenmüller et al. (2016)).	47
3.9	AnnieWAY with mounted multi-sensor system set-up.	48
4.1	Truncated SDF function adapted from logistic function.	53
4.2	a) Plot of $g(x, y)$ from origin and green contour line shows all the points with distance equal to $r = 5$ in τ units from origin , b) Truncated implicit surface $\hat{D}(x, y)$ and c) Respective weighting function $W(x, y)$ with projected green contour lines highlighting the suspected surface.	54
4.3	Projection of world information onto depth and color cameras.	55
4.4	Cross-section of a voxel-grid with color coded SDF values.	57
4.5	a) Error-prone depth measurements represented as one-dimensional TSDF function and b) Respective weight values generated using standard Gaussian function.	58
4.6	a) Fused TSDF function and b) Updated weighting function.	59
4.7	Mean absolute surface error convergence with incremental fusion.	60
4.8	Undesirable holes in reconstructed model from <i>ICL-LR2</i> trajectory.	61
4.9	Large-scale 3D reconstruction using hashed voxel grid.	62
5.1	Mean absolute surface error convergence with incremental fusion.	70
5.2	Error-prone depth measurements (y_0 and y_1) represented as one-dimensional TSDF function.	71
5.3	Fused TSDF function with the help of Equation 5.8.	72

5.4	Comparison of the convergence between RLSFusion and traditional 3D fusion.	72
5.5	Capability of RLSFusion to accommodate less noisy measurements.	74
5.6	Mean absolute surface error convergence with incremental fusion.	79
5.7	Illustration of volumetric regularization and integration process using color coded voxel values.	80
5.8	a) Erroneous depth measurements represented with SDF signals and b) Estimated SDF signal from traditional incremental methods and RFusion. . . .	82
5.9	Comparison of traditional fusion (upper row) and proposed regularized fusion (bottom row) after fusing 10 depth images	83
5.10	a) A synthetic surface and corresponding 3D points with additive outliers and b) Illustrated SORF passes.	84
5.11	Comparison of outliers removal using proposed-SORF vs PCL-SOR	85
5.12	Block diagram of SmoothFusion with online and offline processing scenarios. .	88
5.13	Modular design of SmoothFusion to handle multiple sensors and their respective 3D reconstructed models.	89
5.14	a) SmoothFusion with IPS module to use provided sensor pose b) SmoothFusion with stereo matching module combined with ORB-SLAM2 for real-time processing.	89
6.1	3D reconstruction of noisy depth images from <i>ICL0</i> trajectory a-c) Pseudo color coded 3D samples representing absolute surface error from ground truth for the three methods InfiniTAM, FastFusion and SmoothFusion d) Color scale representing absolute surface error in a-c, e) Error histogram, f) Cumulative error distribution and g) Median error comparison.	93
6.2	3D reconstruction of noisy depth images from <i>ICL2</i> trajectory a-c) Pseudo color coded 3D samples representing absolute surface error from ground truth for the three methods InfiniTAM, FastFusion and SmoothFusion d) Color scale representing absolute surface error in a-c, e) Error histogram, f) Cumulative error distribution and g) Median error comparison.	94
6.3	a) Textured ground-truth 3D model, b) color removed to highlight geometry, f) color scale representing absolute surface error in c,d,e g) error histogram, h) cumulative error distribution and i) median error comparison.	95

6.4	Per-frame memory consumption of the reconstruction framework for <i>KITTI-06</i> trajectory.	97
6.5	Per-frame memory consumption of the reconstruction framework for <i>mine</i> trajectory.	97
6.6	Per-frame memory consumption of the reconstruction framework for <i>ICL-2</i> trajectory.	98
6.7	Effects of employing proposed-TVR smoothing with InfiniTAM (upper row) and FastFusion (bottom row).	100
6.8	Reconstructed models from <i>mine</i> dataset captured with IPS	101
6.9	Reconstructed models from <i>corridor</i> dataset captured with IPS	102
6.10	Reconstructed model from <i>LR0</i> trajectory with clean depth images	103
6.11	Reconstructed model from <i>LR2</i> trajectory with noisy depth images	104
6.12	Reconstructed model from kitti dataset sequence 06 with two close-up screenshots.	105
6.13	Reconstructed 3D model from <i>electric cabinet</i> trajectory from CoRBS dataset.	106
6.14	Per-frame processing time in large scale environment for <i>mine</i> dataset (a) and small scale synthetic environment <i>LRI</i> (b) (lower is better)	108

List of Tables

2.1	State-of-the-art 3D shape reconstruction approaches	33
2.2	Incremental 3D Fusion frameworks	34
3.1	3D depth sensors and their respective characteristics	50
5.1	Comparison of memory consumption (in bytes) among dense, sparse vs RLSFusion.	73
6.1	Mean absolute surface error (in mm) for <i>living-room</i> dataset trajectories	96

Chapter A

Appendix

A.1 Formulation of D and C matrices

D and C matrices are used to approximate the second order difference for particular voxel location given the SDF signal. For the sake of notation simplicity, it is presumed that voxel-grid is represented in 2D. Therefore, each cell and respective neighboring cells can be accessed by their respective spatial information (i.e. row and column values in case of 2D). For each voxel value a_k (where $0 \leq k \leq \text{support}$) in the SDF-signal v , assuming that i and j are index values of row and column respectively for accessing a_k in Equation (A.1), finite difference in vector form can be written as

$$\nabla a_k = \begin{bmatrix} \nabla_{xx} \\ \nabla_{yy} \\ \nabla_{xy} \\ \nabla_{yx} \end{bmatrix}$$

$$\begin{bmatrix} \nabla_{xx} \\ \nabla_{yy} \\ \nabla_{xy} \\ \nabla_{yx} \end{bmatrix} = \begin{bmatrix} a(i-1, j) - 2a(i, j) + a(i+1, j) \\ a(i, j-1) - 2a(i, j) + a(i, j+1) \\ \frac{a(i+1, j+1) - a(i+1, j) - a(i, j+1) + 2a(i, j) - a(i-1, j) - a(i, j-1) + a(i-1, j-1)}{2} \\ \frac{a(i+1, j+1) - a(i+1, j) - a(i, j+1) + 2a(i, j) - a(i-1, j) - a(i, j-1) + a(i-1, j-1)}{2} \end{bmatrix} \quad (\text{A.1})$$

Elements of Equation (A.1) can be separated depending upon whether the elements are

available in the incident ray which is currently being fused or in neighboring cell. The separated elements can then be written using multiple matrix form as

$$\nabla a_k = D_k v + C_k \quad (\text{A.2})$$

where

$$D_k = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ -2 & 0 & 0 & \dots & 0 \\ 1 & 0.5 & 0 & \dots & 0 \\ 1 & 0.5 & 0 & \dots & 0 \end{bmatrix}$$

$$C_k = \begin{bmatrix} a(i-1, j) \\ a(i, j-1) + a(i, j+1) \\ \frac{a(i+1, j+1) - a(i+1, j) - a(i-1, j) - a(i, j-1) + a(i+1, j+1)}{2} \\ \frac{a(i+1, j+1) - a(i+1, j) - a(i-1, j) - a(i, j-1) + a(i+1, j+1)}{2} \end{bmatrix}$$

D_k and C_k matrix in Equation (A.2) are only valid¹ for a_k (where $k = 1$). However by using the same method, composite D and C matrices can be formulated and written as

$$\nabla v = \begin{bmatrix} \nabla a_1 \\ \nabla a_2 \\ \dots \\ \nabla a_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_n \end{bmatrix} v + \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_n \end{bmatrix}$$

$$\nabla v = Dv + C \quad (\text{A.3})$$

Matrix C from Equation (A.3) is used in the later stages of RFusion to incorporate the integrated smoothing.

¹Values of D and C matrices are calculated at run time. Actual formulation depends upon the angle of ray from camera, size of SDF-signal etc.

A.2 Technical Requirements

Actual implementation of proposed contributions and complete framework is carried in modern C++ programming language using object oriented constructs to ensure that final design is modular and extendable in cross-platform development and deployment environment to ensure compatibility in Windows and Linux operating systems. The final implementation have been tested to work on a desktop computer having following specifications:

- Intel Core i7-4790
- 8 GB RAM
- Windows 7 (64-bit) and Linux 14.04 operating system.

Functionality of following softwares and open-source libraries are utilized in implementation:

- **Softwares**
 - MeshLab
 - CloudCompare
 - GNUPlot
 - CMake
- **Open-source C++ libraries**
 - OpenCV
 - OpenVDB
 - Eigen
 - OpenGL
 - Pangolin
 - Boost

