PhD-FSTC-2016-35
The Faculty of Sciences, Technology and Communication

DISSERTATION

Presented on 06/09/2016 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Hassan AFZAL
Born on 30th October 1986 in Muzaffarabad AJK, (Pakistan)

FULL 3D RECONSTRUCTION OF DYNAMIC NON-RIGID SCENES: ACQUISITION AND ENHANCEMENT

Declaration of Authorship

I, Hassan AFZAL, declare that this thesis titled, 'Full 3D Reconstruction of Dynamic Non-Rigid Scenes: Acquisition and Enhancement' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

Abstract

Recent advances in commodity depth or 3D sensing technologies have enabled us to move closer to the goal of accurately sensing and modeling the 3D representations of complex dynamic scenes. Indeed, in domains such as virtual reality, security, surveillance and e-health, there is now a greater demand for affordable and flexible vision systems which are capable of acquiring high quality 3D reconstructions. Available commodity RGB-D cameras, though easily accessible, have limited field-of-view, and acquire noisy and low-resolution measurements which restricts their direct usage in building such vision systems. This thesis targets these limitations and builds approaches around commodity 3D sensing technologies to acquire noise-free and feature preserving full 3D reconstructions of dynamic scenes containing, static or moving, rigid or non-rigid objects. A mono-view system based on a single RGB-D camera is incapable of acquiring full 360° 3D reconstruction of a dynamic scene instantaneously. For this purpose, a multi-view system composed of several RGB-D cameras covering the whole scene is used. In the first part of this thesis, the domain of correctly aligning the information acquired from RGB-D cameras in a multi-view system to provide full and textured 3D reconstructions of dynamic scenes, instantaneously, is explored. This is achieved by solving the extrinsic calibration problem. This thesis proposes an extrinsic calibration framework which uses the 2D photometric and 3D geometric information, acquired with RGB-D cameras, according to their relative (in)accuracies, affected by the presence of noise, in a single weighted bi-objective optimization. An iterative scheme is also proposed, which estimates the parameters of noise model affecting both 2D and 3D measurements, and solves the extrinsic calibration problem simultaneously. Results show improvement in calibration accuracy as compared to state-of-art methods. In the second part of this thesis, the domain of enhancement of noisy and low-resolution 3D data acquired with commodity RGB-D cameras in both mono-view and multi-view systems is explored. This thesis extends the state-of-art in mono-view template-free recursive 3D data enhancement which targets dynamic scenes containing rigid-objects, and thus requires tracking only the global motions of those objects for view-dependent surface representation and filtering. This thesis proposes to target dynamic scenes containing non-rigid objects which introduces the complex requirements of tracking relatively large local motions and maintaining data organization for view-dependent surface representation. The proposed method is shown to be effective in handling non-rigid objects of changing topologies. Building upon the previous work, this thesis overcomes the requirement of data organization by proposing an approach based on view-independent surface representation. View-independence decreases the complexity of the proposed algorithm and allows it the flexibility to process and enhance noisy data, acquired with multiple cameras in a multi-view system, simultaneously. Moreover, qualitative and quantitative experimental analysis shows this method to be more accurate in removing noise to produce enhanced 3D reconstructions of non-rigid objects. Although, extending this method to a multi-view system would allow for obtaining instantaneous enhanced full 360° 3D reconstructions of non-rigid objects, it still lacks the ability to explicitly handle low-resolution data. Therefore, this thesis proposes a novel recursive dynamic multi-frame 3D super-resolution algorithm together with a novel 3D bilateral total variation regularization to filter out the noise, recover details and enhance the resolution of data acquired from commodity cameras in a multi-view system. Results show that this method is able to build accurate, smooth and feature preserving full 360° 3D reconstructions of the dynamic scenes containing non-rigid objects.

Acknowledgements

I would like to express my gratitude to my PhD supervisor Professor Björn Ottersten for his expert guidance, advice and encouragement throughout the course of my doctoral research. I admire greatly his intellectual prowess, and a keen eye for detail. I am thankful to him for allowing me to work under his supervision at the University of Luxembourg. I would also like to thank my industrial supervisor Dr. Bruno Mirbach for his guidance and advice. I really appreciate that he always found the time to have a discussion which helped me greatly in understanding various scientific problems.

I owe a special thanks to my co-supervisor Dr. Djamila Aouada for her daily supervision, her patience, her continuous help and guidance which played a key role in helping me to successfully carry out my research work. I would like to thank Professor David Fofi for his support and guidance which helped me in advancing my research, and also for being part of my PhD Defense Committee. I would like to acknowledge Professor Leon van der Torre for agreeing to chair the PhD Defense Committee and for evaluating my work.

I would also like to acknowledge the support I got from the Computer Vision team especially from Dr. Kassem Al Ismaeil and Dr. Michel Antunes during the course of my work. I would like to thank Dr. François Destelle for his cooperation. I would also like to thank my friends and colleagues Maha, Ahmad and Girum for their share in making my stay in Luxembourg a pleasant one.

The successful completion of this thesis would not have been possible without the support of my family. I really appreciate the lifelong hard work, love, support and prayers of my parents without which none of this would have been possible. I owe a special thanks to my wife Arooj Un Nissa for her love and support, and for bearing with the long working hours and tough work schedules. I thank my brothers, and my sister-in-law for their support and prayers.

This work was supported by the National Research Fund (FNR), Luxembourg, under the CORE project C11/BM/1204105/FAVE/Ottersten.

Contents

Declaration of Authorship **i**

Abstract **ii**

Acknowledgements **iv**

Contents **v**

Abbreviations **viii**

Notation **x**

1 Introduction **1**
- 1.1 Motivation and Scope 1
  - 1.1.1 Calibration of RGB-D Multi-View Systems 3
  - 1.1.2 Enhancement of 3D Dynamic Videos 4
- 1.2 Objectives and Contributions 5
- 1.3 Thesis Outline 7

2 Background **9**
- 2.1 Sensing via Commodity RGB-D Cameras 9
  - 2.1.1 Depth Sensing Technologies 10
  - 2.1.2 Depth Sensing Challenges 11
- 2.2 RGB-D Multi-View System 12
- 2.3 RGB-D Multi-System Calibration 13
  - 2.3.1 Problem Formulation 13
  - 2.3.2 Background and Previous Work 14
    - 2.3.2.1 Bundle Adjustment 14
    - 2.3.2.2 Iterative Closest Point 14
- 2.4 Enhanced 3D Reconstruction 15
  - 2.4.1 Problem Formulation 15
  - 2.4.2 Background and Previous Work 17
    - 2.4.2.1 KinectFusion 17

3 RGB-D Multi-View System Calibration for Full 3D Scene Reconstruction 20
3.1 Introduction . . . 20
3.2 Background and Problem Formulation . . . 22
3.3 Proposed Approach . . . 23
3.4 Experimental Results . . . 25
  3.4.1 Setup & Data Acquisition . . . 25
  3.4.2 Implementation Details . . . 26
  3.4.3 Results and Analysis . . . 27
3.5 Conclusion . . . 31

4 Bi-objective Framework for Sensor Fusion in RGB-D Multi-View Systems: Acquisition and Modeling 32
4.1 Introduction . . . 32
4.2 Problem Formulation . . . 34
4.3 Bi-Objective Extrinsic Calibration . . . 35
4.4 Weighting Factor Estimation . . . 37
4.5 Experiments with Synthetic Data . . . 38
  4.5.1 Evaluation Methodology and Parameters . . . 39
  4.5.2 System Composed of Two Sensors . . . 40
    4.5.2.1 Varying Noise Levels . . . 40
    4.5.2.2 Varying Data Ratio . . . 41
  4.5.3 System Composed of Four Sensors . . . 42
    4.5.3.1 Varying Noise Levels . . . 43
    4.5.3.2 Varying Number of Points . . . 46
4.6 Experiments with Real Data . . . 46
4.7 Conclusion . . . 49

5 KinectDeform: Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects 50
5.1 Introduction . . . 50
5.2 Proposed Approach . . . 52
  5.2.1 Non-Rigid Registration . . . 53
  5.2.2 TSDF Volume Creation and Fusion . . . 55
5.3 Experiments and Results . . . 56
5.4 Conclusion . . . 59

6 View-Independent Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects 61
6.1 Introduction . . . 61
6.2 Background and Problem Formulation . . . 62
  6.2.1 Background . . . 62
  6.2.2 Point Set Surfaces . . . 63
6.3 Proposed Technique . . . 65
  6.3.1 Sampling and MLS Based Projection . . . 65
  6.3.2 Fusion . . . 66
6.4 Experiments and Results . . . 68
6.5 Conclusion and Future Work . . . 72

7 Full 3D Reconstruction of Non-Rigidly Deforming Objects 73
7.1 Introduction . . . 74
7.2 Problem Formulation . . . 76
7.3 Proposed Approach . . . 78
7.3.1 Overview . . . 78
7.3.2 Per-point Refinement via Tracking . . . 79
7.3.2.1 Depth Dependent Measurement Noise . . . 80
7.3.3 Proposed 3D BTV Deblurring . . . 82
7.4 Experiments and Results . . . 83
7.4.1 Evaluation on Synthetic Data . . . 84
7.4.2 Evaluation on Real Data . . . 88
7.5 Conclusion . . . 89

8 Point Cloud Denoising via 3D Bilateral Total Variation Regularization 94
8.1 Introduction . . . 94
8.2 Background and Problem Formulation . . . 96
8.3 3D Bilateral Total Variation Regularization . . . 97
8.4 Experiments and Results . . . 100
8.5 Conclusion . . . 102

9 Conclusions and Future Work 104

A Non-linear Optimization for Proposed Bi-Objective Framework 107

Bibliography 109

Abbreviations

BA	**B**undle **A**djustment
BMD	**B**ilateral **M**esh **D**enoising
BTV	**B**ilateral **T**otal **V**ariation
CPD	**C**oherent **P**oint **D**rift
DLT	**D**irect **L**inear **T**ransform
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
FBS	**F**oward **B**ackward **S**plitting
FOV	**F**ield **O**f **V**iew
GT	**G**round **T**ruth
HR	**H**igh **R**esolution
ICP	**I**terative **C**losest **P**oint
IR	**I**nfra-**R**ed
LM	**L**evenberg **M**arquardt
LR	**L**ow **R**esolution
LS	**L**east **S**quares
ML	**M**aximum **L**ikelihood
MLE	**M**aximum **L**ikelihoodt **E**stimation
MLS	**M**oving **L**east **S**quares
MSE	**M**ean **S**quared **E**rror
NIR	**N**ear **I**nfra-**R**ed
PSS	**P**oint **S**et **S**urfaces
RANSAC	**RAN**dom **Sa**mpling **C**onsensus
RBF	**R**adial **B**asis **F**unction
RGB	**R**ed **G**reen **B**lue
RGB-D	**R**ed **G**reen **B**lue-**D**epth

RMSE	**R**oot **M**ean **S**quared **E**rror
SBA	**S**parse **B**undle **A**djustment
SCAPE	**S**hape **C**ompletion and **A**nimation of **P**eople
SDF	**S**igned **D**istance **F**unction
SIFT	**S**cale-**I**nvariant **F**eature **T**fransform
SL	**S**tructured-**L**ight
SLAM	**S**imultaneous **L**ocalization and **M**apping
SR	**S**uper **R**esolution
ToF	**T**ime-**o**f-**F**light
TORO	**T**ree based Netw**OR**k **O**ptimizer
TSDF	**T**runcated **S**igned **D**istance **F**unction
TV	**T**otal **V**ariation
UP	**UP**sampled

Notation

$\mathtt{w}$	world reference frame
$\mathbf{A}$	matrix
$\mathbf{p}$	vector
n, N	scalars
$\mathcal{U}$	point-set, surface, plane
$\mathbf{I}_n$	identity matrix of dimensions $n \times n$.
$\mathbf{0}_{n \times n}$	null matrix of dimensions $n \times n$.
$\mathbf{0}_n$	null vector of dimensions $n \times 1$.
$\mathbf{A}^\intercal$	transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$
$tr(\mathbf{A})$	trace of matrix $\mathbf{A}$
$\hat{\mathbf{A}}$	estimate of matrix $\mathbf{A}$
$\mathbf{T}\,\|_{\mathtt{a}}^{\mathtt{b}}$:	matrix transformation from $\mathtt{a}$ to $\mathtt{b}$
$\\|\mathbf{p}\\|_2$	L_2 norm of $\mathbf{p}$
$\\|\mathbf{p}\\|_1$	L_1 norm of $\mathbf{p}$
$\arg\min$	the minimizing argument
$\mathcal{N}(\cdot, \cdot)$	Gaussian distribution
$\mathcal{U} \uparrow$	upsampling of point-set $\mathbf{U}$
$sgn(\cdot)$	sign function
$cov(a, b)$	covariance of a and b
$filt(\cdot, \cdot)$	data filtering function
∇	gradient operator
$\nabla\cdot$	discrete divergence operator

Dedicated to my parents Amir and Tehmina Afzal. May you live a long life full of happiness and health.

Chapter 1

Introduction

1.1 Motivation and Scope

Sensing and modeling the 3D world around us has been one of the fundamental goals of research in computer vision and robotics. The acquired 3D models/reconstructions find their applications in various domains such as security and surveillance, virtual reality [1–3], 3D printing [4], 3D telepresence systems [5], creation of viewpoint free 3D videos [6], simultaneous localization and mapping [7], etc. Most of these applications require the 3D reconstructions to be noise-free while containing complete geometric and photometric information from scenes with static or dynamic objects.

Reconstruction of the 3D world has traditionally been achieved by using a single moving camera, or multiple static photometric cameras with overlapping field-of-views (FOVs). It requires the detection of projections of same 3D points in 2D images across different views. This makes the 3D reconstruction process highly sensitive to lighting conditions; thus, limiting the usage of photometric cameras to elaborate setups [8]. These limitations can be relaxed for reconstructing specific objects for which pre-built templates are available or can be built a priori [9–17].

Recently available RGB-D or 3D cameras equipped with commodity depth sensing technologies based on structured-light or time-of-flight principles, such as Microsoft Kinect version 1 and 2 [18], and Asus Xtion Pro Live [19], have opened further the possibilities of research in this domain. They provide, in addition to photometric 2D information, 2.5D range information which can directly be converted into 3D point clouds. Such cameras, with their 3D sensing capabilities, have diminished the barriers for 3D acquisition and reconstruction. However, the partial 3D measurements of the scenes acquired by these cameras are noisy and have limited resolution [20, 21]. Hence, the goal of using commodity cameras in an easy-to-build and flexible setup for acquiring enhanced and high quality, i.e., high accuracy and features-preserving, full 360° 3D reconstructions of scenes remains unfulfilled.

The acquisition of full 3D reconstructions of scenes via commodity RGB-D cameras can be accomplished by using a single moving camera, known as a mono-view system, with its location constantly being tracked [1, 4, 7]. This method is simple and attractive; however, it has the drawback of not allowing to fully reconstruct dynamic scenes, i.e., scenes containing objects in motion, at each time-step. This problem can be solved by using multiple fixed RGB-D cameras. This setup is known as a multi-view system where the FOVs of all cameras together cover the entire scene [2, 3, 5, 22]. In this case, the relative poses of all cameras are required for aligning the partial 3D reconstructions. The problem of estimating the relative poses of cameras in a multi-view system is known as extrinsic calibration. To solve this problem researchers have usually employed classical extrinsic calibration methods which either use photometric information, also referred to as 2D or RGB information [5, 23, 24], or 3D geometric information [25–27] separately instead of using them together to complement the extrinsic calibration process and achieve more accurate results.

Once all the relative poses of cameras are estimated in a multi-view system, the 3D information acquired from each camera can be put into a single reference frame. This raw information has limited resolution and suffers from high noise contamination, which inhibits its direct use in various aforementioned applications. Research has been carried out to improve the quality of information acquired via commodity depth sensing technologies, in both mono-view and multi-view systems. On the one hand, there are template based methods which recursively fuse the captured frame with a smooth template to provide high quality reconstructions of non-rigid objects undergoing local motions (also called deformations). These methods are limited to the class of objects for which templates are available or can be constructed [28–30]. On the other hand, there are template-free methods which recursively fuse a specified number of captured frames to produce high quality 3D reconstructions [1, 2, 31]. The downside of these methods, using both mono-view [1] and multi-view systems [2], is their inability to tackle non-rigid objects undergoing local deformations [32, 33]. This means that they can only reconstruct rigid or quasi-rigid objects which are either static, or are undergoing global deformation [34–36].

The scope of this thesis is to address the above mentioned limitations of state-of-art to synthesize high-quality, and full 3D reconstructions of dynamic scenes, containing rigid or non-rigid objects, from the data acquired via commodity RGB-D or 3D cameras. The first part of this thesis deals with the construction of an RGB-D multi-view system by efficiently exploiting the acquired 2D photometric and 3D geometric information together to solve the extrinsic calibration problem, and provide full textured 3D reconstructions of scenes at each time-step. The second part of this thesis deals with template-free and online enhancement of accuracy and details of, noisy and low-resolution (LR), 3D reconstructions acquired via commodity RGB-D or 3D cameras based mono-view and multi-view systems. We target 3D reconstructions of dynamic scenes containing both rigid or non-rigid objects, undergoing local and global motions, respectively. In the next two sections we expand on the challenges which lie in the way achieving our objectives.

1.1.1 Calibration of RGB-D Multi-View Systems

As mentioned before extrinsic calibration, or simply calibration in the context of a multi-view system, is the process of finding relative poses of all cameras to correctly align the partial 3D reconstructions acquired by them. It is performed by extracting the information of common points, known as feature points, from the acquisitions of different cameras in the multi-view system. Feature points are usually extracted from photometric 2D or geometric 3D acquisitions of objects with known textural and geometric properties [2, 22, 23, 37, 38].

Most of the techniques for extrinsic calibration of RGB-D multi-view systems rely on well established 2D camera based calibration routines and pose refinement procedures [5, 23, 39, 40]. A well established method is Bundle Adjustment (BA) [3, 41, 42] which uses 2D feature points extracted from the RGB or infra-red (IR) images [5, 23, 23, 24, 24, 27]. A major drawback of 2D only calibration approaches is their inability to tackle noise specific to depth sensors. This causes problems in alignment of 3D data from multiple cameras. Researchers have tried to remedy this problem via explicit depth correction for each camera separately [2, 3, 5, 43]. On the other hand, a final refinement step based on Iterative Closest Point (ICP) [44] algorithm is also introduced [27]. ICP solves the extrinsic calibration problem using 3D data only. This final refinement step, based on ICP, tries to mitigate the pose misalignment problem due to depth specific sensor noise. The extrinsic calibration methods mentioned here belong to the classical state-of-art for extrinsic calibration of multi-view systems composed of either only 2D, or 3D cameras. An RGB-D camera provides both 2D photometric (RGB and IR) and 3D geometric (depth) information simultaneously. Therefore, it is interesting to investigate, develop and analyze calibration methods, tailored for RGB-D multi-view systems, which utilize both 2D and 3D information with respect to the noise present in them, to produce more accurate calibration results. In this regard, some weighted bi-objective (pair-wise) pose estimation schemes, mainly in the field of robotics, have been proposed. These methods are restricted to selecting the relative importance (weight) given to each type of information manually [3, 7, 45] or empirically [46] without explicitly taking into account their relative accuracies.

In this thesis we extend the state-of-art to propose a calibration framework tailored for RGB-D multi-view systems. It combines the utilization of both 2D and 3D information in a single weighted bi-objective optimization. We propose an automated scheme which estimates the relative accuracy of 2D and 3D information for computing the weight to be used in the proposed optimization for accurate estimation of relative camera poses.

1.1.2 Enhancement of 3D Dynamic Videos

As mentioned before the raw 3D data acquired via commodity depth sensing technologies suffers from high magnitude of noise and has limited resolution. Therefore, there is a need to enhance this data in terms of removal of noise and recovery or preservation of details to accurately reconstruct generic dynamic scenes without the use of any priori information.

Template-free enhancement of 3D dynamic videos, via recursive temporal data fusion, containing rigid objects has been shown to produce noise-free 3D reconstructions in both mono-view [1, 33, 34, 47, 48] and multi-view systems [2]. The focus of research has since shifted towards tackling non-rigid objects undergoing local deformations. This requires robust and efficient tracking of local changes in each object's topology thus making the recursive data fusion task considerably more challenging. Several recent techniques have targeted high quality and complete 3D reconstructions of quasi-rigid objects, undergoing minimal local motions, by recursively fusing filtered information from different views [49–52]. For tackling non-rigid objects and noisy camera acquisitions, some offline and computationally expensive methods have also been proposed. They use as input a whole sequence of acquired frames and produce as output a sequence containing enhanced and complete 3D reconstruction for per-frame [53–56]. These methods are also known as 4D spatio-temporal reconstruction methods. They are not suitable for online or real-time applications and face limitations in handling large local motions [57, 58].

In this thesis, we extend the state-of-art in the domain of online methods based on recursive temporal data fusion for producing enhanced and complete 3D reconstructions of dynamic scenes containing non-rigid objects undergoing large local motions. We propose methods based on robust non-rigid object tracking and recursive data fusion to remove noise in 3D dynamic videos acquired from mono-view systems. We extend our work to noise-removal and resolutions enhancement of 3D dynamic videos obtained from multi-view systems to obtain enhanced and full 3D reconstructions of dynamic scenes.

1.2 Objectives and Contributions

The objective of this thesis is to tackle the limitations and challenges mentioned in Section 2.3.1 and Section 1.1.2. Specifically, our aim is to investigate, develop and analyze online techniques for enhanced and complete 3D reconstructions of dynamic scenes containing non-rigidly deforming objects using commodity RGB-D or depth cameras. The main contributions of this thesis are as follows:

1. **RGB-D Multi-View System Calibration:** One of the most crucial requirements for building a multi-view system is the estimation of relative poses of all cameras. An approach tailored for an RGB-D camera based multi-view system is missing. We propose a method termed BAICP+ which combines the BA [41] and ICP [59] algorithms to take into account both 2D photometric and 3D geometric information in a weighted bi-objective minimization formulation to estimate relative pose parameters of each camera. BAICP+ can be easily adapted to varying quality of 2D and 3D data. We propose to model the measurement noise in 2D and 3D features points, and use it to model the noise in the corresponding cost functions derived from BA and ICP, respectively. This allows us to analytically derive the proposed weighted bi-objective cost function via the Maximum Likelihood (ML) method. The weighting factor appears as a function of noise in 2D and 3D measurements and takes into account the effect of residual errors on the optimization. We propose an iterative scheme to estimate noise variances in 2D and 3D measurements, in order to simultaneously compute the weighting factor together with the camera poses. Quantitative and qualitative evaluation of the proposed approach, on simulated and real data, shows improved calibration accuracy as compared to refinement schemes which use only 2D or 3D measurement information.

   This work has the following associated publications:

   - **H. Afzal**, D. Aouada, D. Fofi, B. Mirbach, and B. Ottersten. RGB-D Multi-view System Calibration for Full 3D Scene Reconstruction. In 22nd International Conference on Pattern Recognition (ICPR), pages 2459-2464, Aug 2014.
   - **H. Afzal**, D. Aouada, D. Fofi, M. Antunes, B. Mirbach, and B. Ottersten. Bi-objective Framework for Sensor Fusion in RGB-D Multi-View Systems: Applications in Calibration (Under review in The Visual Computer). 2016.

2. **Mono-View Enhancement of 3D Dynamic Videos:** Recursive and template-free enhancement techniques for mono-view dynamic depth or 3D videos, such as

KinectFusion and its derivatives [1, 47, 48], are limited to rigid objects only. In this thesis, we propose KinectDeform, an algorithm which targets enhanced 3D reconstruction of scenes containing non-rigid objects, undergoing local motions. It is the first non-rigid extension of KinectFusion and combines a fast local scene tracking algorithm based on octree data representation, and hierarchical voxel associations with a recursive data filtering mechanism. We analyze its performance on both real and simulated data and show improved results in terms of smoothness and feature preserving 3D reconstructions with reduced noise. While KinectDeform shows satisfactory performance, it is based on a view-dependent data representation scheme due to which it requires organized data. Non-rigid registration destroys the data organization and an expensive re-organization step needs to be carried out. Therefore, we propose a view-independent technique. It uses octrees based space subsampling and explicit projection-based Moving Least Squares (MLS) surface representation. This improved technique is called VI-KinectDeform. Moreover, the empirical weighted filtering scheme in KinectDeform is replaced by an automated fusion scheme based on a Kalman filter [60]. We analyze the performance of KinectDeform and VI-KinectDeform both qualitatively and quantitatively and show that both are able to produce enhanced and feature preserving 3D reconstructions.

This work has the following associated publications:

- **H. Afzal**, K. A. Ismaeil, D. Aouada, F. Destelle, B. Mirbach, and B. Ottersten. KinectDeform: Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects. In The 3DV Workshop on Dynamic Shape Measurement and Analysis, December 2014.
- **H. Afzal**, D. Aouada, F. Destelle, B. Mirbach, and B. Ottersten. View-Independent Enhanced 3D Reconstruction of Non-rigidly Deforming Objects. In 16th International Conference on Computer Analysis of Images and Patterns (CAIP), September 2-4, 2015.

3. **Multi-View Enhancement of 3D Dynamic Videos:** Several approaches for enhanced and full, or complete, 3D reconstructions of non-rigid objects have been proposed in the literature, but they suffer from several limitations due to requirement of a template [29, 50, 61], inability to tackle large local deformations [53, 54], inability to tackle highly noisy and LR data [3, 62], and inability to produce online results [57]. Although our proposed mono-view approach, namely VI-KinectDeform, is able to handle most of these challenges and can be extended to multi-view systems easily, it requires space subsampling based on octrees several times per iteration; thus, making it expensive for use in real-time applications.

Moreover, it does not explicitly target noisy LR data. In this thesis we propose a novel recursive and dynamic multi-frame 3D super-resolution scheme which produces high-resolution (HR), high-quality and complete 3D reconstructions at every time-step by fusing the current acquisition, from a multi-view commodity 3D camera setup, and the result of the previous iteration. The proposed approach is template-free and works directly on 3D points, thus giving it flexibility to the types of objects being reconstructed, and the ability to capture their characteristics, i.e., position and motion in the 3D world, more accurately. To handle system blur and recover smooth position and motion estimates, a novel and efficient multi-level 3D Bilateral Total Variation (BTV) regularization is proposed which is used to correct per-point position and motion estimates, at every iteration. Detailed experimental, quantitative and qualitative, evaluations have been carried out using both simulated and real data. Results show that the proposed dynamic scheme outperforms the state-of-art filtering algorithms and produces feature-preserving and smooth reconstructions.

This work has the following associated publications:

- **H. Afzal**, D. Aouada, B. Mirbach, and B. Ottersten. Full 3D Reconstruction of Non-Rigidly Deforming Objects (To be submitted to Computer Vision and Image understanding). 2016.

1.3 Thesis Outline

The organization of this dissertation is as follows:

- **Chapter** 2: An overview of the commodity depth sensing technologies and the challenges they face in the way of acquiring accurate depth measurements has been presented. This is followed by backgrounds on data acquisition from RGB-D multi-view systems, RGB-D multi-view system calibration and enhanced 3D reconstruction, respectively.
- **Chapter** 3: A sensor-fusion technique, called BAICP+, tailored for calibration of RGB-D multi-view systems is presented which combines Bundle Adjustment (BA), which makes use of 2D photometric information, and Iterative Closest Point (ICP) algorithm, which makes use of 3D geometric information. Experiments with simulated and real data show improved performance as compared to single modality based state-of-art methods.

- **Chapter** 4: Building on BAICP+, a completely automated bi-objective sensor fusion framework for RGB-D multi-view system calibration is presented. It analytically derives a weighted bi-objective cost for estimation of calibration parameters. The cost function depends on measurement noise in 2D and 3D information which can also be estimated automatically in conjunction with calibration parameters. A comprehensive qualitative and quantitative analysis of the performance of proposed technique is presented.

- **Chapter** 5: KinectDeform, a mono-view recursive method for enhanced 3D reconstruction of non-rigid objects undergoing local deformations is presented. It combines an efficient local non-rigid registration method with view-dependent implicit surface representation and is capable of handling generic objects undergoing large local deformations.

- **Chapter** 6: VI-KinectDeform, which improves upon KinectDeform by replacing the view-dependent implicit surface representation with a view-independent explicit surface representation is presented. It also provides an improved data fusion with the help of Kalman filter. Proposed improvements are verified via qualitative and quantitative performance analysis of both algorithms.

- **Chapter** 7: A framework which proposes a multi-frame recursive dynamic 3D super-resolution algorithm is presented. The goal of this framework is the enhancement of resolution and quality of full 360° 3D reconstructions of dynamic scenes, containing non-rigid objects, acquired with commodity 3D cameras based multi-view systems. This framework also targets system blur and achieves globally smooth point clouds by using a novel 3D bilateral total variation (BTV) regularizer. A comprehensive qualitative and quantitative performance analysis of the the proposed framework on real and simulated data is presented and discussed.

- **Chapter** 8: A novel 3D bilateral total variation (BTV) regularization for filtering and smoothing 3D point clouds is presented. The regularizer uses a gradient operator built upon exploiting surface properties in local point patches.

- **Chapter** 9: Conclusions drawn from contributions resulting from research carried out during the course of this thesis together with perspectives for future work are presented.

Chapter 2

Background

This chapter provides an overview of the basic concepts and assumptions underlying the construction of RGB-D multi-view systems. Moreover, recursive 3D data fusion for producing enhanced and full 3D reconstructions of dynamic scenes containing non-rigidly deforming objects is also discussed.

We start by reviewing the acquisition methodologies of commodity RGB-D cameras, with a focus on depth sensing, based on structured-light and time-of-flight principles. Moreover, we discuss the systematic and non-systematic factors affecting measurements of these cameras. After that we discuss the use of such cameras in the construction of a multi-view systems and formulate the extrinsic calibration problem while briefly describing state-of-art methods, namely Bundle Adjustment [41] and Iterative Closest Point (ICP) [44] algorithm, for solving this problem. Similarly we formulate the recursive 3D data enhancement problem and provide a brief description of state-of-art mono-view data enhancement algorithm called KinectFusion [1].

2.1 Sensing via Commodity RGB-D Cameras

The recent and ubiquitous spread of affordable depth/3D sensing technologies has largely been due to the introduction of commodity RGB-D cameras such as Microsoft Kinect version 1 and 2 [18], and Asus Xtion Pro Live [19]. Such cameras are equipped with an RGB camera and a depth camera, and are able to simultaneously acquire mapped RGB and depth images of the scene in their field-of-view (FOV). The acquired RGB and depth images can be used to produce textured 3D reconstructions as shown in Figure 2.1. The depth sensing system present in commodity RGB-D cameras, and other depth only cameras such as PMD Camboard Nano [20], uses active sensing technology,

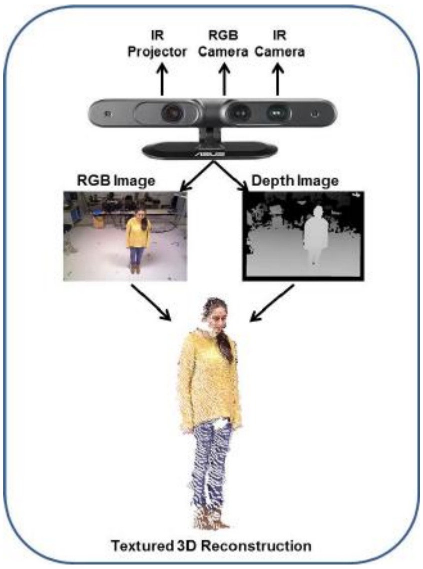


FIGURE 2.1: RGB-D camera components and acquisition. An RGB-D camera, such as Asus Xtion Pro Live [19] shown here, is composed of an RGB camera for acquiring an RGB image and an infra-red (IR) camera for acquiring an IR image which, together with the knowledge of the IR pattern/signal projected from the IR projector, is used to compute the depth image. RGB and depth images are used to generate textured 3D reconstructions.

via an infra-red (IR) camera and an IR projector, based on either structured-light [19] or time-of-flight [20] principles. We give an overview of these principles to understand their working principle for depth sensing. This is followed by an overview of the challenges faced by them in acquiring accurate depth measurements.

2.1.1 Depth Sensing Technologies

1. **Structured-light Cameras:** Structured-light (SL) based depth sensing available in commodity cameras such as Kinect version 1 uses active stereo-vision technology. A near infra-red (NIR) laser projector projects a known pattern onto the scene. The projected pattern gets deformed due to geometry of the scene. The scene is then observed by a monochrome intensity camera from a different direction [63], as shown in Figure 2.2. By analyzing the distortion of the pattern in the observed image with respect to the original projected pattern a per pixel disparity value d is computed [63]. Assuming knowledge of cameras's horizontal focal length f and the baseline b between camera and projector, the depth value z for each image pixel can be computed via $z = \frac{b \cdot f}{d}$. Here both d and f are given in pixel-units while b is in the units of length.

2. **Time-of-Flight Cameras:** Time-of-Flight (ToF) based depth sensing available in commodity cameras such as Kinect version 2 uses NIR intensity modulated

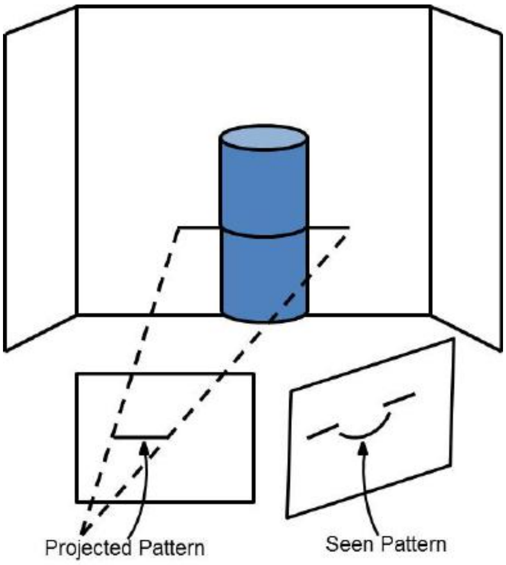


FIGURE 2.2: Illustration of the basic principle of depth/3D sensing with a structured-light camera. 3D geometry is computed by using deformation in the observed and known pattern projected onto the scene (a straight line in this case). (Reproduced from [64]).

periodic light signal to actively illuminate the scene [63]. The optical camera captures the reflected light signal per pixel, and correlates it with the projected signal to compute the phase shift $\Delta\varphi = \varphi(t) - \varphi(t+\delta)$, as illustrated in Figure 2.3. The phase difference $\Delta\varphi$ is equivalent to the time shift in a periodic signal [63]. It is then used to compute the depth z for the corresponding pixel via $z = \frac{c.\Delta\varphi}{2\pi}$, where c is the speed of light.

2.1.2 Depth Sensing Challenges

In this section we briefly describe the challenges faced by the ToF and SL based depth sensing technologies in acquiring accurate depth measurements. This topic has been extensively discussed in the literature [63, 65, 66].

1. **Systematic Depth Errors:** Both SL and ToF cameras suffer from systematic errors in depth measurements due to inadequate intrinsic calibration and limited resolution [63]. The errors due to limited resolution are directly proportional to the increase in distance of points being measured. Moreover, due to mixing of different optical signals with the reference signal, ToF cameras have to approximate the original sinusoidal signal shape, or phase demodulation function, which causes further systematic errors [63]. Moreover, the use of active-light depth sensing technologies in a multi-view system causes interference problems resulting in

inaccurate or undefined depth measurements. These problems can be tackled by for example using "Shake'n'Sense" setup in the case of SL cameras [2, 67], and by using unique modulation frequencies, in the case of ToF cameras [68].

2. **Non-Systematic Depth Errors:** Long exposure times can cause over-saturation of light which may cause difficulties in pattern detection and hence distance/depth computation in SL cameras. Over-saturation can affect the depth measurements in ToF cameras as well, but they are usually equipped with band-pass filters to suppress background light, hence making them more robust and more suitable for use in outdoor environments [63]. Moreover, light scattering or multi-path effects are another source of errors in depth measurements acquired from ToF or SL based active-light systems. Furthermore, object boundaries suffer from inhomogeneous depth measurements due to occlusion from the projected pattern, in the case of SL cameras, and mixing of foreground and background signals, in case of ToF cameras. The pixels containing such inaccurate boundary measurements are known as flying pixels. Temperature-drift is also a source of error in both types of devices wherein the depth measurements vary before and after the warm-up period.

2.2 RGB-D Multi-View System

Herein we introduce the model of the considered RGB-D multi-view system. Let us consider a multi-view system composed of N intrinsically calibrated RGB-D cameras with intersecting FOVs, as shown in Figure 2.4. At a fixed time t, every RGB-D camera l, with $l = 1, \cdots, N$, acquires mapped RGB and depth images of resolution $m \times n$, where $m, n \in \mathbb{N}^*$, denoted by $\mathbf{C}_{(l,t)}$ and $\mathbf{D}_{(l,t)}$, respectively. Each pixel i is defined by its position in the image plane where $\mathbf{q}_l^i = (u_l^i, v_l^i)^\intercal$, and $u_l^i \in \{1, \cdots, n\}$, $v_l^i \in \{1, \cdots, m\}$ and $i \in \{1, \cdots, mn\}$. We assume that the intrinsic parameters of each camera l are known and can be represented by the matrix:

$$\mathbf{K}_l = \begin{pmatrix} f_l^u & 0 & c_u^u \\ 0 & f_v^v & c_l^v \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.1}$$

where (f_l^u, f_l^v) represent the focal lengths of camera l, and (c_l^u, c_l^v) represent the center of its imager, in horizontal and vertical directions, respectively. Using the matrix $\mathbf{K}_l$, depth image $\mathbf{D}_{(l,t)}$ can be converted to a 3D vertex map $\mathbf{V}_{(l,t)}$ of dimensions $3 \times m \times n$. The tensor $\mathbf{V}_{(l,t)}$ represents the partial 3D reconstruction of the scene, acquired with camera l, and contains 3D geometric points $\mathbf{p}_{(l,t)}^i \in \mathbb{R}^3$, such that $\mathbf{p}_{(l,t)}^i = \Psi(\mathbf{q}^i, \mathbf{D}_{(l,t)}, \mathbf{K}_l)$ where

$\Psi(\cdot)$ uses camera's intrinsic parameters to convert a depth value at pixel location $\mathbf{q}^i$ in $\mathbf{D}_{(l,t)}$ to the corresponding 3D point $\mathbf{p}^i_{(l,t)}$.

2.3 RGB-D Multi-System Calibration

2.3.1 Problem Formulation

In this section, we formulate the extrinsic calibration problem for an RGB-D multi-view system. We drop the subscript for time t as it is fixed for this problem. In order to correctly align the partial 3D reconstructions $\{\mathbf{V}_l\}$, where $l = 1, \cdots, N$, acquired by N RGB-D cameras, it is necessary to accurately estimate their positions with respect to a global reference frame, referred to as `world` and denoted by `w`, as shown in Figure 2.4. Each camera's relative position with respect to `w` is defined by:

$$\mathbf{T}_l = \begin{pmatrix} \mathbf{R}_l & \mathbf{t}_l \\ \mathbf{0}_3^\intercal & 1 \end{pmatrix}, \tag{2.2}$$

where $\mathbf{T}_l \in SE(3)$ represents the rigid transformation, from camera l to `w`. The matrix $\mathbf{R}_l$ is rotation matrix in $SO(3)$ and $\mathbf{t}_l \in \mathbb{R}^3$ is translation vector. Therefore the same point $\mathbf{p} \in \mathbb{R}^3$ in `w` viewed by camera l as $\mathbf{p}_l$ and by cameras k as $\mathbf{p}_k$ can be related to the cameras' reference frames as follows:

$$\mathbf{R}_l\mathbf{p}_l + \mathbf{t}_l = \mathbf{R}_k\mathbf{p}_k + \mathbf{t}_k. \tag{2.3}$$

Similarly, for a given point $\mathbf{r} \in \mathbb{R}^3$ in `w`, its projection on each camera's image plane results in 2D pixel coordinates $\mathbf{q}_l$, such that:

$$\mathbf{q}_l = \psi\left(\mathbf{K}_l, \mathbf{T}_l, \mathbf{r}\right), \quad \forall l, \tag{2.4}$$

where $\psi(.)$ is `world` to image plane projection function.

The problem at hand may therefore be stated as follows. Given N RGB-D cameras in a multi-view system with acquired RGB images $\{\mathbf{C}_1, \cdots, \mathbf{C}_N\}$ and 3D vertex maps $\{\mathbf{V}_1, \cdots, \mathbf{V}_N\}$, we assume knowledge of $H \leq mn$ matching points in each camera's RGB image plane referred to as 2D features and denoted as $[\mathbf{q}_l^1, \cdots, \mathbf{q}_l^H]$. Similarly, we assume knowledge of $J \leq mn$ matching 3D points in each camera's 3D vertex map called 3D features and denoted as $[\mathbf{p}_l^1, \cdots, \mathbf{p}_l^J]$. Moreover we assume knowledge of each camera's intrinsic parameters, $\mathbf{K} = [\mathbf{K}_1, \cdots, \mathbf{K}_N]$. Using this information, we want to find the estimates of the parameters $\mathbf{T} = [\mathbf{T}_1, \cdots, \mathbf{T}_N]$.

2.3.2 Background and Previous Work

In this section, we introduce two state-of-art pose refinement algorithms namely Bundle Adjustment (BA) [41] and Iterative Closest Point(ICP) [44] algorithm, which use the 2D and 3D features respectively, to solve the extrinsic calibration problem described in Section 2.3. Bundle Adjustment (BA) has been the method of choice for problems related to multi-view 3D reconstruction and pose refinement based on 2D features extracted from RGB images [41], while Iterative Closest Point (ICP) algorithm has been the de facto solution for pose refinement problems when only 3D features are available [44, 59].

2.3.2.1 Bundle Adjustment

Bundle Adjustment (BA) requires an initial estimate of the pose parameters. Moreover, it also requires an estimate of 3D points i.e., $[\mathbf{r}^1, \cdots, \mathbf{r}^H]$, corresponding to available 2D feature points $[\mathbf{q}_l^1, \cdots, \mathbf{q}_l^H]$. These estimates are then refined by computing the error of projection of estimate of each 3D point $\mathbf{r}^h$, $h = 1, \cdots, H$, corresponding to the 2D feature point $\mathbf{q}_l^h$ to camera l via:

$$\mathbf{a}_l^h(\mathbf{S}_l^h) = \mathbf{q}_l^h - \psi\left(\mathbf{K}_l, \mathbf{T}_l, \mathbf{r}^h\right), \tag{2.5}$$

where $\mathbf{a}_l^h(\mathbf{S}_l^h) \in \mathbb{R}^2$ and $\mathbf{S}_l^h = \left(\mathbf{T}_l, \mathbf{r}^h\right)$ [1]. Therefore, the total BA cost to be minimized for the refinement of estimates of each camera's pose parameters together with the estimates of 3D points corresponding to 2D feature points is given as:

$$V_{BA}(\mathbf{S}) = \sum_{l=1}^{N} tr(\mathbf{A}_l^\intercal(\mathbf{S}_l)\mathbf{A}_l(\mathbf{S}_l)), \tag{2.6}$$

where $\mathbf{S} = (\mathbf{T}, \mathbf{r})$, $\mathbf{S}_l = (\mathbf{T}_l, \mathbf{r})$, $\mathbf{r} = [\mathbf{r}^1, \cdots, \mathbf{r}^H]$ and $\mathbf{A}_l(\mathbf{S}_l) = [\mathbf{a}_l^1(\mathbf{S}_l^1), \cdots, \mathbf{a}_l^H(\mathbf{S}_l^H)]$.

2.3.2.2 Iterative Closest Point

Iterative Closest Point (ICP) algorithm also uses initial estimates of the pose parameters and minimizes the Euclidean distance between corresponding 3D feature points from different views, such that:

$$\mathbf{b}_{l,k}^j(\mathbf{T}_l, \mathbf{T}_k) = (\mathbf{R}_l\mathbf{p}_l^j + \mathbf{t}_l) - (\mathbf{R}_k\mathbf{p}_k^j + \mathbf{t}_k), \tag{2.7}$$

---

[1] BA can also refine the estimate of intrinsics $\mathbf{K}_l$ if required

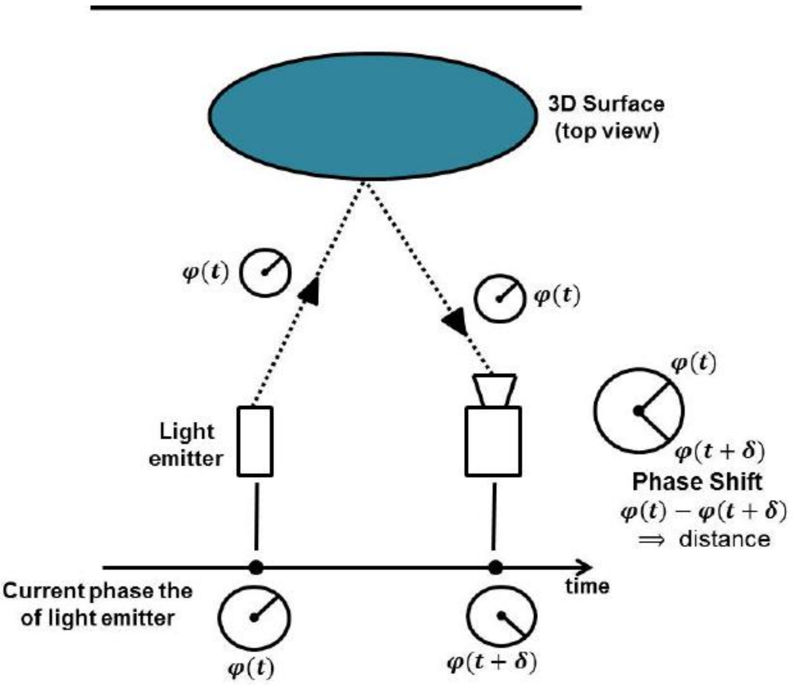


Figure 2.3: Illustration of the basic principle of depth/3D sensing via time-of-flight (ToF) camera. 3D geometry of the scene is computed by using the phase delay between projected and sensed light signal. (Reproduced from [64]).

where $\mathbf{b}^j_{l,k}(\mathbf{T}_l, \mathbf{T}_k) \in \mathbb{R}^3$ and $j \in [1, \cdots, J]$. Therefore, the total ICP cost to be minimized for refinement of each camera's pose parameters is given as:

$$V_{ICP}(\mathbf{T}) = \sum_{\substack{1<l,k<N \\ l\neq k}} tr(\mathbf{B}^{\intercal}_{l,k}(\mathbf{T}_l, \mathbf{T}_k)\mathbf{B}_{l,k}(\mathbf{T}_l, \mathbf{T}_k)), \tag{2.8}$$

where $\mathbf{B}_{l,k}(\mathbf{T}_l, \mathbf{T}_k) = [\mathbf{b}^1_{l,k}(\mathbf{T}_l, \mathbf{T}_k), \cdots, \mathbf{b}^J_{l,k}(\mathbf{T}_l, \mathbf{T}_k)]$.

2.4 Enhanced 3D Reconstruction

2.4.1 Problem Formulation

We herein formulate the problem of obtaining enhanced 3D reconstruction of dynamic scenes using the data acquired with RGB-D or depth only cameras using template-free recursive data fusion. We start by formulating this problem for a mono-view system and discuss its extension to a multi-view system in order to obtain full and high quality 3D reconstructions of dynamic scenes.

A fixed and fully calibrated depth camera acquires a sequence of consecutive noisy measurements of a dynamic scene in the form of depth maps $\{\mathbf{D}_t\}$ and their corresponding

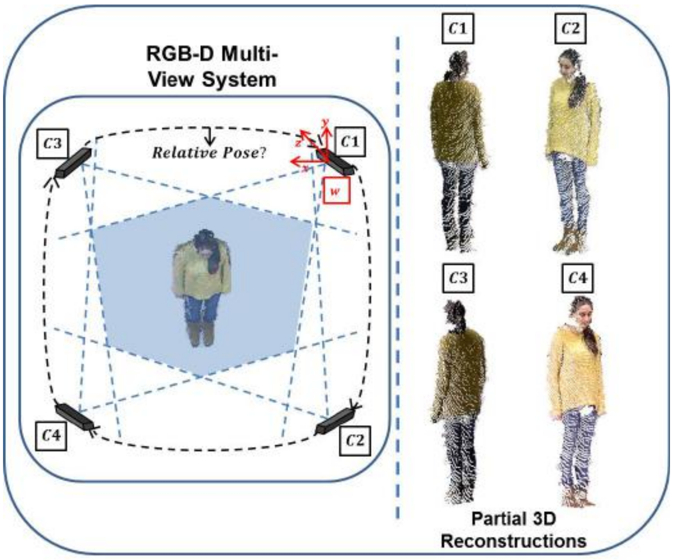


FIGURE 2.4: A multi-view system composed of 4 RGB-D cameras with overlapping field-of-views (FOVs) to capture the full scene. The poses of these cameras relative to each other, or to a common global reference frame `w` assumed to be lying in camera $\boldsymbol{C1}$, are unknown therefore the partial 3D reconstruction acquired by them cannot be correctly aligned.

3D vertex maps $\{\mathbf{V}_t\}$. This data may represent deformable moving surfaces in the depth camera's FOV.

Each vertex map $\mathbf{V}_t$ is related to the previous vertex map $\mathbf{V}_{t-1}$ via:

$$\mathbf{V}_t = h_t\left(\mathbf{V}_{t-1}\right) + \boldsymbol{\epsilon}_t, \tag{2.9}$$

where $h_t(\cdot)$ is the deformation that transforms $\mathbf{V}_{t-1}$ to its consecutive vertex map $\mathbf{V}_t$. The additional term $\boldsymbol{\epsilon}_t$ represents the error map due to the acquisition system depending on factors discussed in Section 2.1.2.

The problem at hand is therefore to attenuate $\boldsymbol{\epsilon}_t$, and recover an enhanced sequence $\{\mathbf{V}_t^f\}$ starting from the acquisition $\{\mathbf{V}_t\}$.

As a solution, a recursive filtering function $filt(\cdot,\cdot)$ may be defined by sequentially fusing the current measurement $\mathbf{D}_t$ and the resulting enhanced vertex map $\mathbf{V}_{t-1}^f$ of the previous time-step such that:

$$\mathbf{V}_t^f = \begin{cases} \mathbf{V}_t & \text{for } t = 0, \\ filt(\mathbf{V}_{t-1}^f, \mathbf{D}_t) & t > 0. \end{cases} \tag{2.10}$$

We now consider a fully calibrated RGB-D multi-view setup as explained in Sections 2.2 and Section 2.3. With the knowledge of pose parameters $\mathbf{T}$ for the N cameras, we can correctly align all the 3D information acquired by them, at a given time t, by

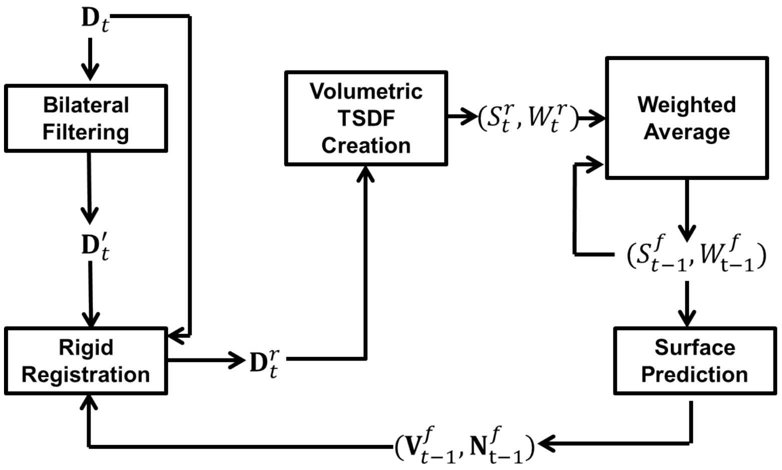


FIGURE 2.5: Detailed pipeline of KinectFusion. $\mathbf{D}_t$: input depth map at time t, $\mathbf{D}'_t$: result of bilateral filter on $\mathbf{D}_t$, $(\mathbf{V}^f_{t-1}, \mathbf{N}^f_{t-1})$: filtered vertex map and corresponding normal map at time $t-1$, $\mathbf{D}^r_t$: result of rigid registration of $\mathbf{D}'_t$ to $\mathbf{V}^f_{t-1}$, (S^r_t, W^r_t) and (S^f_{t-1}, W^f_{t-1}): TSDF volumes corresponding to vertex maps $\mathbf{V}^r_t$ and $\mathbf{V}^f_{t-1}$ respectively. For more details please see Section 2.4.2.1.

transforming them to a global reference frame `w`, and produce a full, albeit noisy, 3D reconstruction of the scene. Let us denote by $\mathcal{L}_t$ the point-set representing the full 3D reconstruction obtained by concatenating all the aligned partial reconstructions, from the multi-view system at time t. $\mathcal{L}_t$ contains M 3D points, where $M = Nmn$. Given a sequence of full 3D reconstructions $\{\mathcal{L}_t\}$ acquired with the RGB-D multi-view system, the task is to obtain noise-free 3D reconstructions $\{\mathcal{L}^f_t\}$ via a similar template-free recursive filtering function as given in (2.10).

2.4.2 Background and Previous Work

The KinectFusion algorithm [1], and its derivatives [2, 47, 48], try to solve the problem formulated in the Section 2.4.1 for obtaining 3D reconstructions of dynamic scenes, using both mono-view and multi-view RGB-D systems, and achieve impressive results but are restricted to reconstructing rigid objects only [1]. In what follows we provide a brief overview of the KinectFusion algorithm for template-free and recursive enhancement of 3D reconstructions of dynamic scenes containing rigid objects.

2.4.2.1 KinectFusion

The KinectFusion algorithm provides a practical solution for the recursive problem defined in (2.10) for the special case where the deformation h_t is global, i.e., when the transformation between $\mathbf{V}_{t-1}$ and $\mathbf{V}_t$ is a single rotation and translation with 6 degrees of freedom [1].

Figure 2.5 shows the detailed pipeline of the KinectFusion algorithm. In the first step, a 2D bilateral filter is applied to the input depth map $\mathbf{D}_t$ resulting in a filtered map $\mathbf{D}_t^{'}$ [1, 69]. The new depth map $\mathbf{D}_t^{'}$ is then given as input to the registration module where its corresponding vertex map $\mathbf{V}_t^{'}$ is computed. The normal map $\mathbf{N}_t^{'}$ is also computed for each 3D point in $\mathbf{V}_t^{'}$ using neighboring points.

The registration step uses a multi-resolution point-plane error metric coupled with a projective data association–based variation of the ICP algorithm to estimate the camera (or conversely object) pose [1, 70]. This second step estimates the global deformation between $\mathbf{V}_t^{'}$ and $\mathbf{V}_{t-1}^{f}$ using their corresponding normal maps $\mathbf{N}_t^{'}$ and $\mathbf{N}_{t-1}^{f}$, respectively. This transformation is applied to $\mathbf{V}_t$ (computed from $\mathbf{D}_t$) to get $\mathbf{V}_t^{r}$, which is back projected to image plane using camera's intrinsic matrix $\mathbf{K}$ in order to obtain $\mathbf{D}_t^{r}$. It is then fused with a global surface representation to get an enhanced 3D surface reconstruction. We note that the reason for using $\mathbf{D}_t$ instead of $\mathbf{D}_t^{'}$ for fusion is to preserve the details which might have been lost due to bilateral filtering. For the last step of data fusion or filtering, KinectFusion uses a method based on the signed distance function (SDF) representation of a surface in 3D [1, 71]. An SDF $S_t(.)$ corresponding to a vertex map $\mathbf{V}_t$ represents points on surface as zeros, and free spaces in front of and behind the surface as positive and negative values, respectively. These values increase as distance from the surface increases. The SDF is formally defined as:

$$\begin{aligned} S_t: \quad \mathbb{R}^3 \quad &\to \mathbb{R} \\ \mathbf{p} \quad &\mapsto \begin{cases} d(\mathbf{p}, \mathbf{V}_t) & \mathbf{p} \text{ lies in front of } \mathbf{V}_t, \\ 0 & \mathbf{p} \in \mathbf{V}_t, \\ -d(\mathbf{p}, \mathbf{V}_t) & \mathbf{p} \text{ lies behind } \mathbf{V}_t, \end{cases} \end{aligned}$$

where $d(.)$ calculates the shortest distance between a given 3D point $\mathbf{p}$ and $\mathbf{V}_t$. KinectFusion uses a volumetric representation of the truncated SDF (TSDF). It is called TSDF because the SDF is truncated using a limiting value of $\pm\mu$. A continuous TSDF is sampled by a volume of resolution $(Z \times Z \times Z)$ with $Z \in \mathbb{N}^*$, lying in the camera's reference frame. The volume consists of volumetric elements called voxels where each voxel is represented by its 3D centroid $\mathbf{p}$, such that $\mathbf{p} \in \mathbb{R}^3$. A TSDF volume corresponding to $\mathbf{V}_t^{r}$ is defined by two values computed for each of its voxels $\mathbf{p}$; one is the TSDF value itself $S_t^{r}(\mathbf{p})$, and second is the weight $W_t^{r}(\mathbf{p})$, using camera parameters $\mathbf{K}$ such that:

$$S_t^{r}(\mathbf{p}) = \Omega(\|\mathbf{p}\|_2 - \|\mathbf{p}'\|_2), \tag{2.11}$$

where $\mathbf{p}' = g(\mathbf{q}, \mathbf{D}_t^r, \mathbf{K})$ and $\mathbf{q} = \psi\left(\mathbf{K}, \mathbf{I}_4, \mathbf{p}\right)$, and

$$\Omega(\eta) = \begin{cases} min\{1, \frac{\eta}{\mu}\} \cdot sgn(\eta) & \text{iff } \eta \geq -\mu, \\ 0 & \text{otherwise,} \end{cases} \tag{2.12}$$

where μ is the truncation distance and $sgn()$ is the sign function. Note that $\mathbf{q}$ represents a location on the 2D grid of $\mathbf{D}_t^r$. The weight $W_t^r(\mathbf{p})$ should be proportional to the measure of similarity of pixel ray direction from $\mathbf{q}$ to $\mathbf{p}$ to local surface normal at point $\mathbf{p}'$ but Newcombe et al. show that keeping the weight $W_t^r(\mathbf{p}) = 1$ works well for their filtering scheme of KinectFusion which will be discussed next [1]. For filtering, KinectFusion follows a scheme of weighted average of all TSDF volumes computed for $\mathbf{V}_t^r$ resulting in one global filtered TSDF volume where each voxel in the filtered volume is represented by $S_t^f(\mathbf{p})$ and $W_t^f(\mathbf{p})$ such that:

$$S_t^f(\mathbf{p}) = \frac{W_{t-1}^f(\mathbf{p}) S_{t-1}^f(\mathbf{p}) + W_t^r(\mathbf{p}) S_t^r(\mathbf{p})}{W_t^f(\mathbf{p})}, \tag{2.13}$$

where

$$W_t^f(\mathbf{p}) = W_{t-1}^f(\mathbf{p}) + W_t^r(\mathbf{p}). \tag{2.14}$$

It is to be noted that $W_t^f(\mathbf{p})$ is reset to a default value after a fixed number of iterations. The vertex map $\mathbf{V}_t^f$ is computed from the current filtered volume for the next iteration using surface prediction via ray casting [1, 72]. The normal map $\mathbf{N}_t^f$ is also computed using the gradient of the TSDF values in the filtered volume. The final extraction of the surface or the point cloud in 3D from the filtered volume can be carried out by using zero crossings or iso-surfaces in the TSDF volume followed by linear interpolation of points.

Chapter 3

RGB-D Multi-View System Calibration for Full 3D Scene Reconstruction

Chapter 3 presents a sensor fusion approach for extrinsic calibration of an RGB-D multi-view system. The proposed solution makes use of the 2D photometric and the 3D geometric information acquired with RGB-D cameras. Both pieces of information are used in a single weighted bi-objective optimization problem for pose estimation. The weighting factor decides the relative importance given to 2D or 3D information in the optimization. This formulation combines two well known pose estimation/refinement frameworks, namely, Bundle Adjustment, which uses the 2D photometric information, and the Iterative Closest Point algorithm, which uses the 3D geometric information. In the experiments on simulated data, the weighting factor is varied manually to show that more accurate results can be achieved via the proposed sensor fusion framework instead of using data from different modalities, independently.

3.1 Introduction

The advent of commodity depth cameras such as structured-light based RGB-D cameras, e.g., Asus Xtion Pro Live and Microsoft Kinect powered by PrimeSense technology [73], has provided researchers with various opportunities to explore the domain of fast, accurate and holistic 3D reconstructions of scenes in a multi-view setup. One of the most crucial steps towards building such a multi-view system is estimating relative poses of all cameras in the system. Only then the independently acquired data from each camera can be put in a single reference frame to form a holistic 3D reconstruction of the scene.

Kuster et al. have proposed to use visual features obtained from color coded markers to determine intrinsic and relative pose parameters of cameras in a multi-view RGB-D network [6]. Kainz et al. have proposed a system called OmniKinect (based on KinectFusion [1]) for high quality dense volumetric reconstruction of static scenes using multiple Kinect cameras with highly overlapping FOVs [2]. The initial pose estimates are generated by extracting 2D features from a cube shaped target with special patterns which is followed by further refinement using depth information. Some other approaches were proposed where after an initial pose estimation using visual features only [5], further refinement is achieved by loop closing using 3D points [49]. Furuakwa et al. proposed to add a third refinement step using the Bundle Adjustment (BA) framework to minimize the back-projection error of selected 3D points [74]. Although 2D photometric features and 3D geometric features are both used for multi-view pose estimation and reconstruction, most of the work done follows a similar pattern of using them independently.

Researchers in the field of robot mapping have explored combining 2D visual information and 3D or depth information captured with an RGB-D camera in an analogous problem where a single RGB-D camera is mounted on a robot for mapping an unknown environment. Indeed, in their work called RGB-D Mapping, Henry et al. use visual features extracted via Scale-Invariant Feature Transform (SIFT), or Features from Accelerated Segment Test (FAST), from RGB images to perform an initial pair-wise alignment based on Random Sample Consensus (RANSAC) method [7]. If enough visual features are not found, then a joint optimization based on 2D visual features and 3D points from input point clouds using RANSAC and dense Iterative Closest Point (ICP) framework is carried out. Globally consistent alignments can be reached by making use of a pose graph optimization technique such as Tree-based Network Optimizer (TORO) or sparse BA [7]. Similar works on environment mapping and localization were presented in [75–77]. A comparative technique is proposed in [78] which uses only depth images acquired using an RGB-D camera. Penelle et al. have extended the idea of using 3D information corresponding to 2D visual features for alignment of two RGB-D cameras facing each other [26]. They use a two sided chessboard pattern to detect corners in RGB images which act as the visual features [79].

A generic and holistic technique tailored for estimating the relative poses of all RGB-D cameras in an n-camera multi-view system is missing. Such a technique should be able to use all the available information acquired by RGB-D cameras in a single framework and should be flexible enough to be able to adapt to changing conditions. It should allow shape information to compensate for inadequate or noisy visual information and vice versa.

In this work, we propose an algorithm called BAICP+ which, as the name suggests, is based on two well known refinement frameworks. First is BA, which takes into account

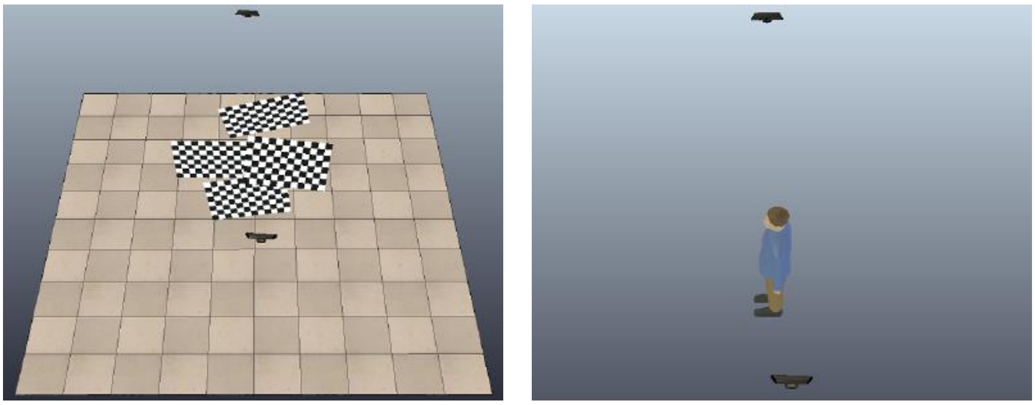
2D visual and its corresponding 3D shape information and is capable of estimating camera intrinsic and pose parameters while refining the 3D shape. Second is ICP, which takes into account the 3D point clouds from different views and tries to register them in a common reference frame while refining the relative poses. Moreover, the generic nature of our algorithm allows for any type of 2D visual features to be easily incorporated in the refinement framework. It can also allow for the estimation of intrinsic parameters of RGB or depth cameras and the extrinsic parameters relating RGB and depth cameras in an RGB-D camera.

3.2 Background and Problem Formulation

The extrinsic calibration problem for a multi-view system composed of N RGB-D cameras has been described in detail in Section 2.2. For simplicity, we assumed in Section 2.2 that each RGB-D camera l, with $l = 1, \cdots, N$, is composed of an RGB camera and a depth camera that are fully mapped hence, requiring a single associated intrinsic matrix $\mathbf{K}_l$, and a single transformation matrix $\mathbf{T}_l$ relating RGB-D camera's relative position to the `world` or `w`. However, for the sake of generalization, in this work, we assume independent or unmapped RGB and depth cameras constituting each RGB-D camera. Therefore, for each RGB-D camera l, let $\mathbf{K}_l^c$ and $\mathbf{K}_l^d$ be the intrinsic matrices associated with its RGB and depth cameras, respectively. Similarly, let $\mathbf{T}_l\,|_c^{\mathtt{w}}$ and $\mathbf{T}_l\,|_d^{\mathtt{w}}$ be the rigid transformation matrices representing the poses of its RGB and depth cameras to `w`, respectively.

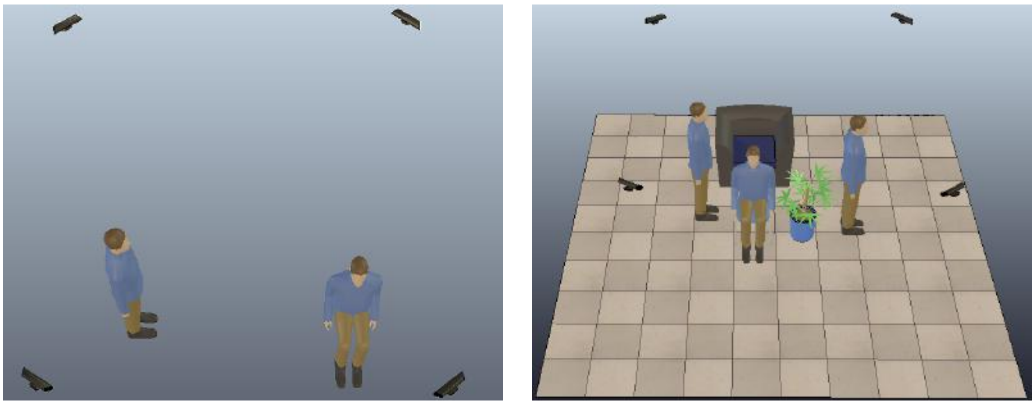
The problem at hand may now be stated as follows. Given N RGB-D cameras in a multi-view system, we assume the knowledge of $H \leq mn$ matching 2D photometric features $\mathbf{q}_l^h$ extracted from the acquired RGB images, where $h = 1, \cdots, H$, $m \times n$ is the resolution of acquired RGB and depth images and $m, n \in \mathbb{N}^*$. Similarly we assume the knowledge of $J \leq mn$ matching 3D geometric features $\mathbf{p}_l^j$ captured by the depth cameras, where $j = 1, \cdots, J$. Moreover, we assume the knowledge of intrinsic parameters, $\mathbf{K}^c = [\mathbf{K}_1^c, \cdots, \mathbf{K}_N^c]$ and $\mathbf{K}^d = [\mathbf{K}_1^d, \cdots, \mathbf{K}_N^d]$. Using this information, we want to find the estimates of the pose parameters $\mathbf{T}^c = [\mathbf{T}_1\,|_c^{\mathtt{w}}, \cdots, \mathbf{T}_N\,|_c^{\mathtt{w}}]$ and $\mathbf{T}^d = [\mathbf{T}_1\,|_d^{\mathtt{w}}, \cdots, \mathbf{T}_N\,|_d^{\mathtt{w}}]$.

As discussed in Section 2.3.2.1 and Section 2.3.2.2, given an initial estimate of pose parameters and of 3D points i.e., $\mathbf{r}^h$, corresponding to the available 2D photometric features, the BA algorithm refines these estimates with help of 2D features only. The 2D feature extraction needs to be robust enough to accurately detect the same features in multiple views [41]. The ICP algorithm on the other hand, refines the initial pose estimates with the help of only 3D geometric features extracted from the scene but it



(a) 2 camera multi-view system with calib. patterns

(b) 2 camera multi-view system with Test Scene 1



(c) 4 camera multi-view system with Test Scene 2

(d) 4 camera multi-view system with Test Scene 3

FIGURE 3.1: Simulated multi-view System with locations of calibration patterns and test scenes.

requires large overlap between multiple views and also might not perform well if the scene does not contain sufficient shape textures [59]. When both types of information are available, as is the case with RGB-D cameras, it becomes interesting to use both BA and ICP in a single framework to compliment each other.

3.3 Proposed Approach

The main idea of this work is to make use of both 2D photometric and 3D geometric or shape information acquired via an RGB-D camera to estimate poses of all cameras in a multi-view system. The final goal is to get a holistic 3D reconstruction of static or dynamic scenes. For this purpose we propose BAICP+, a technique based on BA and ICP simultaneously. To that end, we consider the cost functions of BA defined in (2.6) and of ICP defined in (2.8), and redefine the pose estimation problem as a weighted bi-objective optimization by introducing a new cost function V_{BAICP} such that:

$$V_{BAICP}(\mathbf{S}') = \frac{(1-\alpha)}{a} V_{ICP}(\mathbf{T}^d) + \frac{s\alpha}{b} V_{BA}(\mathbf{S}), \tag{3.1}$$

where $\mathbf{S}' = \big(\mathbf{T}^c, \mathbf{T}^d, \mathbf{r}\big)$, $\mathbf{S} = (\mathbf{T}^c, \mathbf{r})$ and $\mathbf{r} = [\mathbf{r}^1, \cdots, \mathbf{r}^h]$. The BA cost function V_{BA} introduced in (2.6) is redefined in (3.2) to make it robust to the situations where a camera is not able to view or detect any number of 2D feature points such that:

$$V_{BA}(\mathbf{S}) = \sum_{l=1}^{N} tr(\mathbf{W}_l(\mathbf{A}_l^\intercal(\mathbf{S}_l)\mathbf{A}_l(\mathbf{S}_l))), \tag{3.2}$$

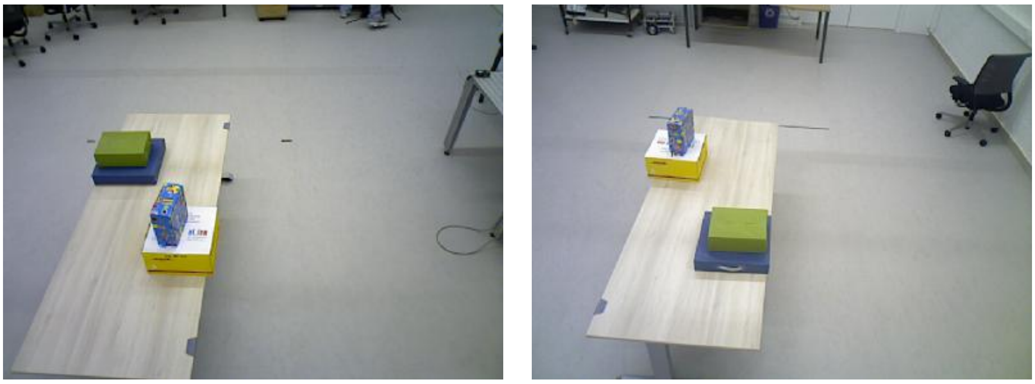
where $\mathbf{S}_l = (\mathbf{T}_l^c, \mathbf{r})$ and $\mathbf{W}_l$ is a diagonal matrix, of dimensions $H \times H$, which contains 1 on its h^{th} diagonal element if camera l is able to view the feature point $\mathbf{r}^h$, and 0 otherwise, for $h = 1, \cdots, H$. Similarly, the ICP cost function V_{ICP} introduced in (2.8) is redefined as follows:

$$V_{ICP}(\mathbf{T}^d) = \sum_{\substack{1<l,k<N \\ l \neq k}} tr(\mathbf{W}_{(l,m)}(\mathbf{B}_{l,k}^\intercal(\mathbf{T}_l^d, \mathbf{T}_k^d)\mathbf{B}_{l,k}(\mathbf{T}_l^d, \mathbf{T}_k^d))), \tag{3.3}$$

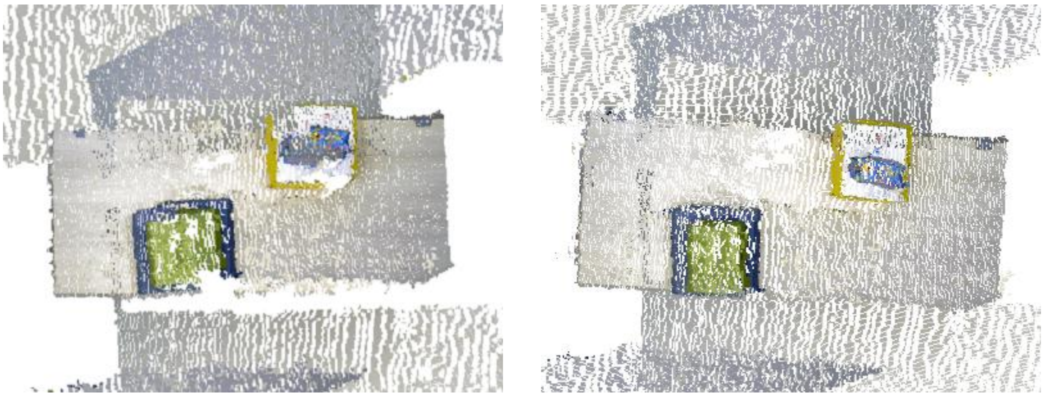
where $\mathbf{W}_{(l,m)}$ is a diagonal matrix, of dimensions $J \times J$, which contains 1 on its j^{th} diagonal element if camera l and camera m are able to view the feature point $\mathbf{p}^h$, and 0 otherwise, , for $j = 1, \cdots, J$.

The parameters a and b in (3.1) denote the total number of 3D point correspondences and 2D feature points across all views. Two new factors s and α have been introduced where s is a scale factor used to unify the units of V_{ICP} and V_{BA}. Indeed, while the cost functions of BA computes Euclidean distances in pixels, the cost function of ICP computes the distance in the units of 3D coordinates. Therefore, the parameter s is a factor which makes the cost computed by BA to be approximately in the same unit as ICP. The scale factor s is defined as $s = (\frac{m^d}{m^f})^2$, where m^d is the average depth of all 3D points in `w` and m^f is the average focal length (in pixels) of all cameras per iteration. As mentioned before, the reason for introducing new optimization parameters, namely the transformations $\mathbf{T}^d$, is to show the generality of our formulation. In this work, we are still mainly concerned with estimating the relative pose parameters with respect to all RGB cameras, i.e., $\mathbf{T}^c$, and hence in experimental evaluations, use data mapped to RGB cameras.

The weight α is a factor that introduces the flexibility in BAICP+ to treat the costs of BA and ICP as the same or one greater than the other. The choice of α may be adaptive to the data and the system. It can, therefore, give insight on the relative importance between BA and ICP, i.e., between visual features and depth/3D information. For



(a) View from camera 1 (b) View from camera 2



(c) Registration with Stereo (d) Registration with BAICP+

FIGURE 3.2: A comparison of registration of 2 views between Stereo and BAICP+ on real data. Only two chessboard acquisitions were used for calibration.

example, when very poor or no shape information is available BAICP+ should behave as BA algorithm by setting $\alpha \approx 1$, or when very poor or no 2D visual information is available BAICP+ should behave as ICP algorithm by setting $\alpha \approx 0$. Solving a minimization similar to (3.1) with an adaptive search for the optimal α will be discussed in Chapter 4.

3.4 Experimental Results

3.4.1 Setup & Data Acquisition

For carrying out a detailed performance analysis (both qualitative and quantitative) of BAICP+, a two pronged approach of using both real and simulated data is followed. The tool which is used to generate simulated data is called V-REP which is developed by Coppelia Robotics [80]. A two sided chessboard pattern is used in both real and simulated experiments as shown in Figure 3.1(a) to overcome the problem of lack of overlap between views. The corners can act as easily detectable 2D features (54 corners/acquisition in real data and 77 corners/acquisition in simulated data) whereas the 3D points belonging to the pattern are extracted and act as 3D shape features. Only poses are estimated in both types of experiments and intrinsic parameters are assumed to be fixed.

For real experiments we use Asus Xtion Pro Live [19] two RGB-D cameras. These cameras are mounted on a lift fixed with the roof and placed almost opposite to each other about 4 meters apart. They are tilted towards the floor to capture the scene as shown in Figure 3.2(a) and Figure 3.2(b). Each camera acquires an RGB image and depth map mapped in RGB camera's reference frame. BAICP+ can be easily adapted to this kind of setup.

We simulate multi-view systems based on 2 or 4 RGB-D cameras in V-REP as shown in Figure 3.1. Acquired data (both RGB and depth) are already mapped in RGB cameras' reference frames. Three test scenes are used as shown in Figure 3.1(b), 3.1(c) and Figure 3.1(d) with varying complexity. We also vary the number of calibration pattern acquisitions, the weighting factor α for BAICP+, and amount of noise (and outliers) in data.

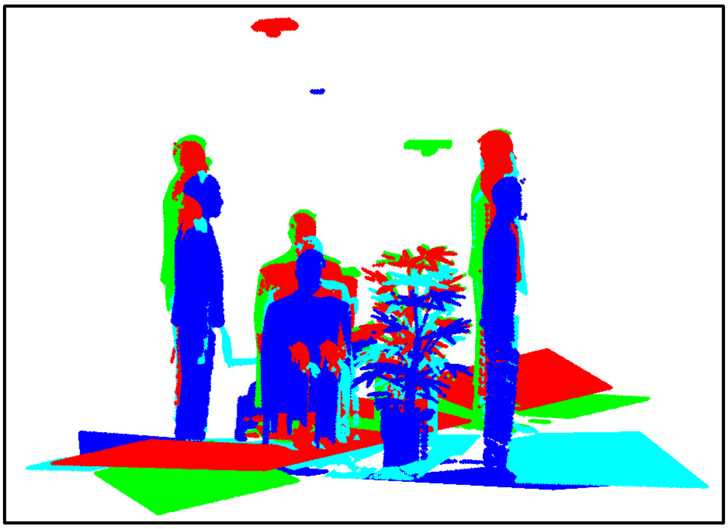
We assume that 2D features extracted from RGB images are affected by the systematic noise due to quantization, as discussed in Section 2.1.2. We add noise to the depth measurements, and hence to 3D feature points, via disparity following the principles of depth sensing using structured-light technology as discussed in Section 2.1.1. For a particular point/pixel in the depth image, noise n_d in disparity d, effects the the depth measurement z as follows:

$$n_z = -\frac{z^2}{f \cdot b} n_d, \tag{3.4}$$

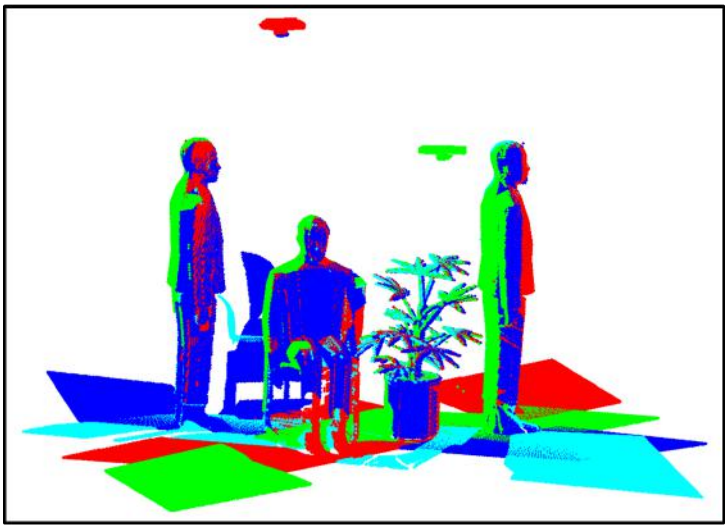
where n_z represents additive noise in depth z such that $\tilde{z} = z + n_z$, where $\tilde{z}$ is the noisy z. These noisy depth values can then be used to compute the corresponding noisy 3D points. We use Gaussian distribution with zero mean and standard deviation of 0.2887 to generate a noise distribution to be used as disparity noise or n_d. The reason for choosing this specific value is the similarity of resulting noisy data with real world acquisitions [81].

3.4.2 Implementation Details

We now discuss the implementation details for carrying out the experimental evaluation of BAICP+. Chessboard corners are detected and extracted from RGB images using Bouguet's Camera Calibration Toolbox in MATLAB [40]. Shape information corresponding to extracted corners is used to extract all 3D points belonging to the planar pattern via RANSAC. Initial pose estimation based on stereo calibration using 2D visual



(a) Registration with Stereo



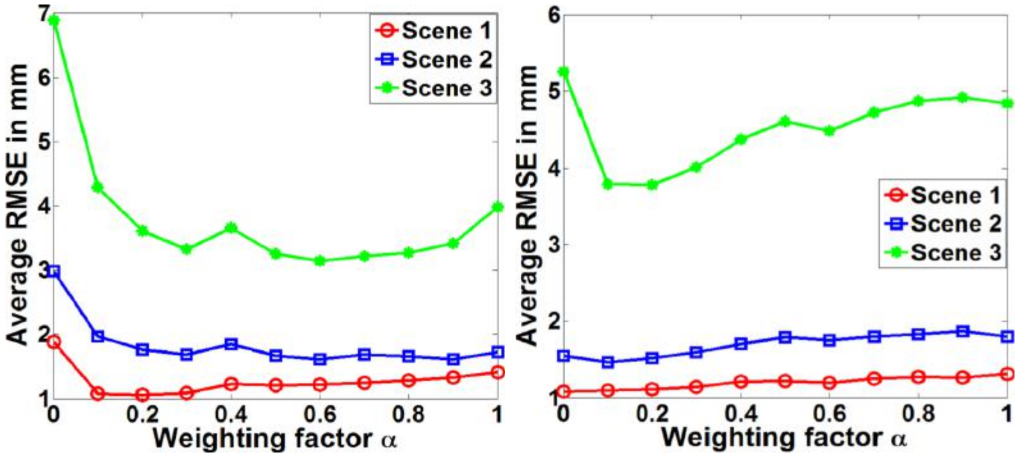
(b) Registration with BAICP+

FIGURE 3.3: A comparison of registration of 4 views (each view represented with a different color) between Stereo and $BAICP+$ using 2 Calibration Acquisitions & Test Scene 3.

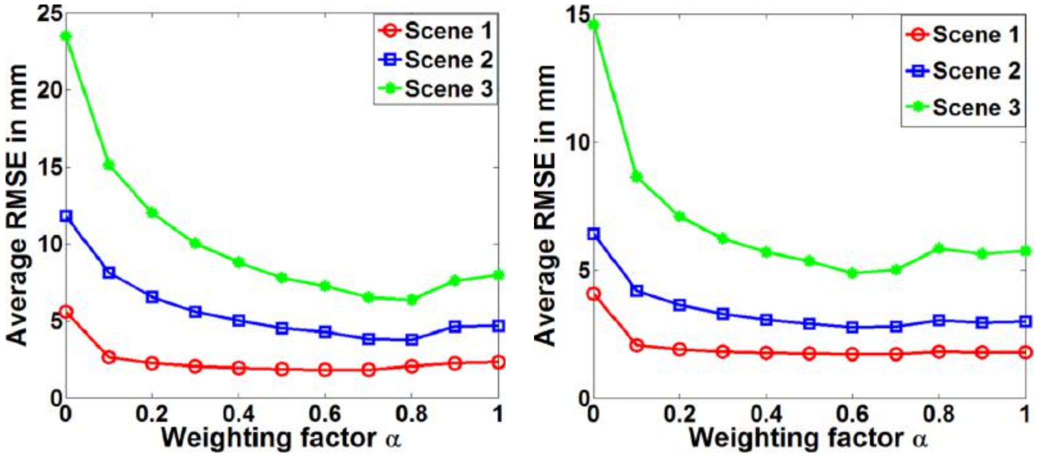
features is performed using the OpenCV library in C++ [79] while the Point Cloud Library (PCL) is used for visualization [82]. The rest of the implementation takes place in MATLAB. We use the non-linear optimization scheme based on Levenberg-Marquardt algorithm to solve BAICP+ with varying weights. We run our method for 20 iterations.

3.4.3 Results and Analysis

For real data we perform initial stereo calibration based pose estimation, that we refer to simply as "Stereo" in our experiments, and then use BAICP+ with $\alpha = 0.5$ to give



(a) 2 cameras & 2 calibration acquisitions (b) 2 cameras & 4 calibration acquisitions

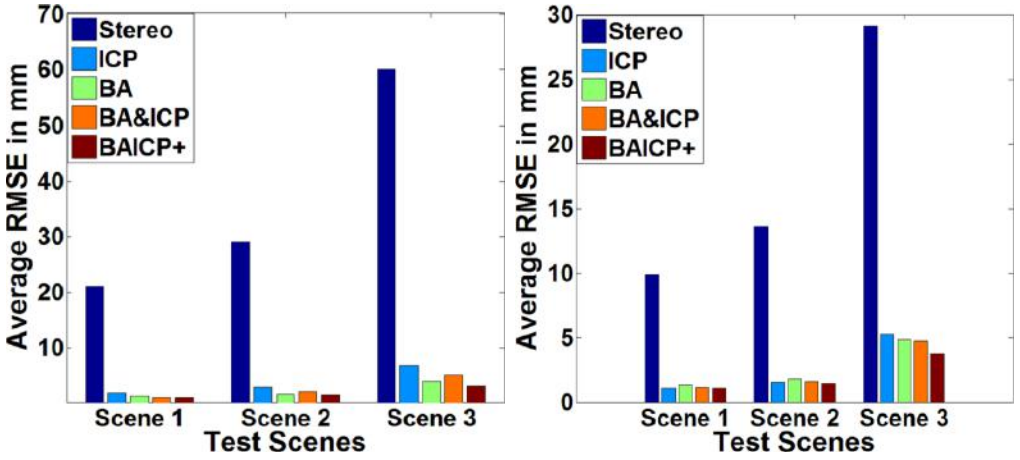


(c) 4 cameras & 2 calibration acquisitions (d) 4 cameras & 4 calibration acquisitions

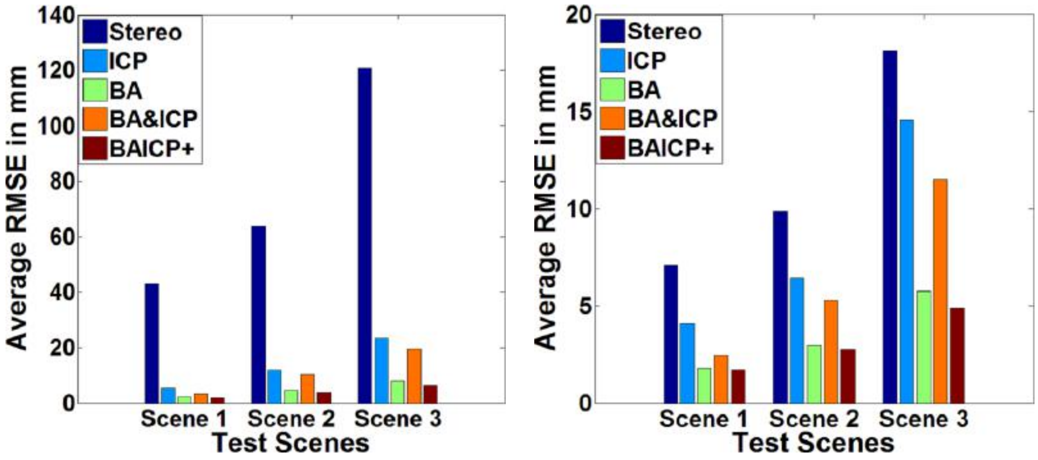
FIGURE 3.4: Weighting factor α vs Average RMSE on test scenes for varying number of cameras and number of calibration acquisitions.

a proof of concept of our method. We vary the number of calibration acquisitions, thus varying the number of feature points. Figure 3.2(c) and Figure 3.2(d) show a significant improvement over initial pose estimates based on stereo calibration method using only 2 calibration chessboard acquisitions [40].

Similar results can be seen for simulated data in terms of visual quality as shown in Figure 3.2. Visually, it is difficult to see the difference when results of BAIPC+ are compared with the results of BA, ICP, and BA followed by ICP (BA&ICP). That is why the availability of ground truth data in this case allows for deeper quantitative analysis and performance comparison of BAICP+. For error calculation, the principle of back projection of noise free 3D data from camera to a global reference frame using perfect and estimated parameters is used. A 3D point $\mathbf{p}_l^m$ acquired by camera l, for $m = 1, \cdots, M$, and M is total number of 3D points acquired by camera l, is projected back to $\mathtt{w}$ via the estimated and perfect pose parameters to give $\hat{\mathbf{p}}_m$ and $\mathbf{p}_m$, respectively. Next step is to compute the Root Mean Squared Error (RMSE) based on the point-wise



(a) 2 cameras & 2 calibration acquisitions (b) 2 cameras & 4 calibration acquisitions



(c) 4 cameras & 2 calibration acquisitions (d) 4 cameras & 4 calibration acquisitions

FIGURE 3.5: Performance comparison of state-of-the-art methods with BAICP+ on test scenes for varying number of cameras and number of calibration acquisitions

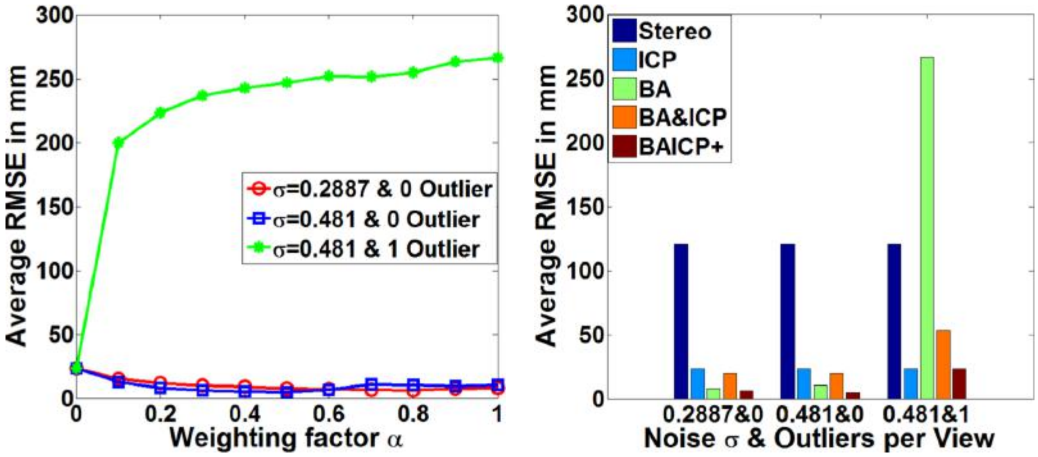
Euclidean distance between $\hat{\mathbf{p}}_m$ and $\mathbf{p}_m$ such that:

$$err_l = \sqrt{\frac{1}{M}(\sum_{m=1}^{M} \|\mathbf{p}_l^m - \hat{\mathbf{p}}_l^m\|^2)}. \tag{3.5}$$

For N cameras the total error becomes:

$$err = \frac{1}{N}\sum_{l=1}^{N} err_l. \tag{3.6}$$

In the first set of experiments, we analyze the performance of BAICP+ in setups composed of 2 and 4 cameras, and 2 and 4 calibration acquisitions. We calculate the average RMSE per setup for the 3 considered test scenes, as shown in Figure 3.1, and analyze the performance of BAICP+ by varying α between 0 and 1 as shown in Figure 3.4. By analyzing these results we can draw the conclusion that since with $\alpha = 0$ and $\alpha = 1$,



(a) 4 cameras and 2 calibration acquisitions with varying α

(b) 4 cameras & 2 calibration acquisitions performance comparison

Figure 3.6: Effects of increased noise and outliers using Test Scene 3

BAICP+ behaves as only ICP, and only BA, respectively, these results show that the combination of these two methods with α between 0 and 1 gives the best performance in all cases but one (Test Scene 1 in Figure 3.4(b)). The results also show that α gives BAICP+ the ability to adapt itself and behave as ICP only or BA only when required. In the second set of experiments, we present a comparison of performance of state-of-the-art methods with BAICP+. Again setups with 2 and 4 cameras were used with 2 and 4 calibration acquisitions and 3 test scenes. We compare Stereo, ICP (which is BAICP+ with $\alpha = 0$), BA (which is BAICP+ with $\alpha = 1$), BA&ICP (performing BA then ICP for 10 iterations each) and BAICP+. For BAICP+, we use best result from Figure 3.4. The results are shown in Figure 3.5 from which it is clear that if the right weighting factor is selected then BAICP+ gives the optimal performance.

In another experiment, using a specific setup of 4 cameras and 2 calibration acquisitions, we increased noise in depth data by increasing the standard deviation σ of Gaussian distribution representing noise in disparity to 0.481. After that we add 1 outlier per view in the 2D visual features. We tested the results on Test Scene 3. The results are shown in Figure 3.6. We can see in Figure 3.6(a) that increased noise has small effect on the performance of BAICP+ whereas adding 1 outlier per view has a large impact as α goes on assigning more weight to the cost of the BA term. Figure 3.6(b) shows that BAICP+ still has the optimal performance and can be easily adapted to behave as ICP when outliers in 2D features are present.

3.5 Conclusion

We have presented a generic formulation for RGB-D cameras based multi-view system calibration. Our approach called BAICP+ combines BA and ICP in a single minimization framework thus making use of both 2D visual and 3D geometric information. It can be used to estimate both relative camera poses and structure parameters. Results over different setups and test scenes show that with right weighting factor BAICP+ has optimal performance compared to both BA and ICP when used independently and sequentially for pose estimation. Moreover, the generic nature of BAICP+ does not restrict it to RGB-D cameras only but allows it to be used with other multi-view systems based, for example, on perspective cameras, depth only cameras, unaligned RGB and depth cameras, etc.

Chapter 4

Bi-objective Framework for Sensor Fusion in RGB-D Multi-View Systems: Acquisition and Modeling

In this chapter we present an automated sensor fusion framework. It is based on a weighted bi-objective optimization for refinement of extrinsic calibration of an RGB-D multi-view system. We build upon the work presented in Chapter 3 and derive an analytical expression for the weighting factor, in the bi-objective optimization, in terms of noise in the RGB and depth measurements. In the absence of information regarding measurement noise, a completely automated and iterative scheme is proposed, which alternates between camera pose estimation, and the computation of measurement noise levels. The proposed framework is shown to perform better than state-of-art methods on both simulated and real data.

4.1 Introduction

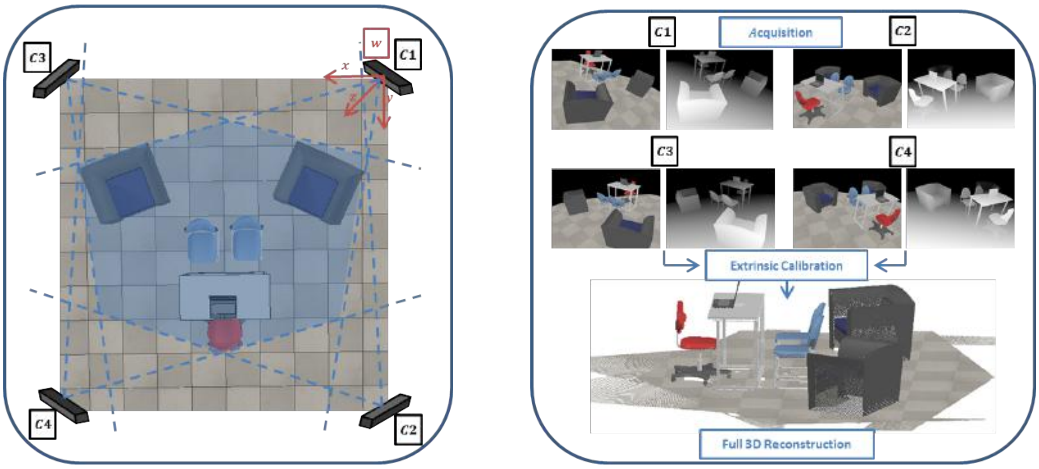
RGB-D cameras provide simultaneous image and range data of the environment, offering enhanced sensing capabilities when compared to using single sensor modality. The acquisition of complete and textured 3D models of dynamic scenes can be achieved by using several RGB-D cameras with overlapping FOVs in a multi-view system. The task at hand is to find the relative poses of these cameras; also known as extrinsic calibration.

Most of the works for extrinsic calibration of RGB-D multi-view systems rely on well established 2D camera based calibration routines [39, 40] and pose refinement procedures, e.g., Bundle Adjustment (BA) [3, 41, 42], using 2D feature points extracted from the RGB images [5, 23, 24]. The 3D information from the depth sensor has mainly been used in subsequent refinement steps using, e.g., the Iterative Closest Point (ICP) algorithm [25–27]. In this regard, the following question arises: how to optimally use both sources of complementary information.

Dou and Fuchs [3], in their work on multi-view 3D reconstruction, proposed to combine 2D and 3D information in a weighted bi-objective optimization scheme derived from their previous work on pair-wise pose tracking for mono-view 3D reconstruction [45]. They propose to use matching feature points extracted via SIFT from RGB images with matching planes extracted from 3D/depth images in a weighted bi-objective BA scheme. The weighting factor is selected empirically for all experiments. A similar approach is proposed by Henry et al. [7], using a global ICP scheme to align 2D visual feature points and 3D/depth measurements from multiple views but the weights are, again, selected empirically. Tykkala et al. [46] use what they call an image based direct ICP approach for pairwise pose estimation. They propose to compute the weighting factor via a heuristic measure using ratio of the median intensity and the depth values of selected points. Michot et al. [83] propose to use a weighted bi-objective BA scheme for the multi-sensor Simultaneous Localization and Mapping (SLAM) problem. They discuss the dependence of the weighting factor on the ratio of the noise variance for each sensor's measurement and formulate their bi-objective optimization by using a mean squared error (MSE) based cost function from individual sensors. They investigate three methods for automatic weight computation namely L-Curve, L-Tangent Norm and cross validation with experiments showing that the L-Curve based method performs better than the others.

In Chapter 3, we introduced BAICP+ [84] which combines BA and ICP in a heuristically constructed weighted bi-objective refinement approach. In our experiments, we varied the weight factor manually and showed that combining 2D and 3D information provides better results than state-of-art refinement approaches based on cost functions using only 2D or 3D information. We also hypothesized that the weight factor depends on the quality of the 2D and 3D information available.

In this chapter we extend and consolidate the work presented in Chapter 3 by investigating a formal strategy for RGB-D sensor fusion for the extrinsic calibration of multiple cameras. We analytically derive a Least Squares (LS) based cost function, via the Maximum Likelihood (ML) method, that optimally combines the BA based 2D cost function with the ICP based 3D cost function, in a weighted bi-objective optimization scheme.



(a) RGB-D Multi-View System (4 cameras)

(b) Full 3D Scene Reconstruction

FIGURE 4.1: RGB-D Multi-View System with full scene 3D reconstruction in a simulated setup. (a) RGB-D Multi-View System (4 cameras) with field of view (FOV) of each camera. The highlighted region represents overlapping FOVs of all cameras. The global reference frame `w` is aligned with camera $\boldsymbol{C1}$. (b) Steps required for Full 3D Scene Reconstruction using an RGB-D Multi-View System. Each camera acquires a RGB image and a depth image, which are used to estimate the relative pose of each camera with respect to `w`. After extrinsic calibration, estimated poses are used to put all acquisitions in `w` to get complete reconstruction.

The sensor fusion is achieved by using a weighting factor that depends on two types of noise; the one affecting the 2D feature locations in the RGB images, and the second one affecting the 3D point positions provided by the depth sensor. The experiments suggest that using the proposed joint cost for relative pose refinement provides more accurate results than the refinement schemes using 2D and 3D information separately.

In the absence of information regarding noise levels in the 2D and 3D feature points we propose an iterative scheme which simultaneously estimates the noise along with the estimation of calibration parameters. The proposed scheme is completely automated requiring no manual intervention and no heuristic parameter setting. The quantitative and qualitative experiments show that the proposed scheme is able to perform sensor fusion for accurate camera calibration without any prior information about noise characteristics.

4.2 Problem Formulation

The extrinsic calibration problem for an RGB-D multi-view system with N cameras, as illustrated in Figure 2.4 and Figure 4.1, has been formulated in Section 2.3.1. As a reminder, we assume the knowledge of $H \leq mn$ matching 2D photometric feature points

$\mathbf{q}_l^h$, for $l = 1, \cdots, N$ and $h = 1, \cdots, H$, extracted from RGB images of resolution $m \times n$ and the knowledge of $J \leq mn$ 3D geometric feature points $\mathbf{p}_l^j$, for $j = 1, \cdots, J$, extracted from 3D/depth images also of resolution $m \times n$. With mapped RGB and depth images we also assume the knowledge of per camera intrinsic parameters, $\mathbf{K} = [\mathbf{K}_1, \cdots, \mathbf{K}_N]$. Using this information we want to find the estimates of per camera pose parameters $\mathbf{T} = [\mathbf{T}_1, \cdots, \mathbf{T}_N]$, where a matrix $\mathbf{T}_l$ represents the rigid transformation from camera l to the `world` `w`, and is composed of a rotation matrix $\mathbf{R}_l$ and translation vector $\mathbf{t}_l$ as defined in (2.2). The cost functions defined in Section 2.3.2.1 and Section 2.3.2.2 use the 2D and 3D information to estimate $\mathbf{T}$, separately. These cost functions assume to have the knowledge of initial estimate of pose parameters and of 3D points i.e., $\mathbf{r} = [\mathbf{r}^1, \cdots, \mathbf{r}^H]$, corresponding to the available 2D feature points, which are then refined.

4.3 Bi-Objective Extrinsic Calibration

In this section, we present the bi-objective optimization for refinement of the extrinsic calibration parameters in an RGB-D multi-view system. We use cost functions defined in Section 2.3.2.1 and Section 2.3.2.2 which use 2D and 3D feature points extracted from RGB images and vertex maps, respectively.

In this work, we propose to formally analyze and derive an expression for the cost function, based on ML estimations, of the bi-objective optimization taking into account the noise affecting both 2D and 3D measurement/feature points. We assume the presence of independent additive Gaussian noise in each coordinate of the 3D feature points such that:

$$\tilde{\mathbf{p}}_l^j \sim \mathcal{N}\left(\mathbf{p}_l^j, \sigma_{3D}^2 \mathbf{I}_3\right), \tag{4.1}$$

where $\tilde{\mathbf{p}}_l^j$ is the noisy 3D point and $\mathbf{p}_l^j$ is the noise free point. Similarly for 2D feature points we have:

$$\tilde{\mathbf{q}}_l^h \sim \mathcal{N}(\mathbf{q}_l^h, \sigma_{2D}^2 \mathbf{I}_2), \tag{4.2}$$

where $\tilde{\mathbf{q}}_l^h$ is the noisy 2D point and $\mathbf{q}_l^h$ is the noise free point. This means that we have to use the noisy 2D and 3D feature points to estimate the pose parameters. This leads to redefining the 3D error function $\mathbf{b}_{l,m}^j(\mathbf{T}_l, \mathbf{T}_m)$, given in (2.7), such that it computes the error between noisy points $\tilde{\mathbf{p}}_l^j$ and $\tilde{\mathbf{p}}_l^m$ projected to `w`, from camera l and m, using the pose parameters $\mathbf{T}_l$ and $\mathbf{T}_m$, respectively. Similarly the 2D error function $\mathbf{a}_l^h(\mathbf{S}_l^h)$, given in (2.5) where $\mathbf{S}_l^h = (\mathbf{T}^l, \mathbf{r}^h)$, is redefined such that it computes the 2D error between back projection of the estimated 3D point $\mathbf{r}^h$ to camera l, using $\mathbf{T}_l$ and $\mathbf{K}_l$, and the corresponding noisy 2D feature point $\tilde{\mathbf{q}}_l^h$.

Now, we can define the distribution the 3D error $\mathbf{b}_{l,m}^{j}(\mathbf{T}_l, \mathbf{T}_m)$ is drawn from by considering the noise free 3D points $\mathbf{p}_l^j$ and $\mathbf{p}_m^j$ in (2.3) such that [25]:

$$
\begin{aligned}
\mathbf{b}_{l,m}^{j}(\mathtt{T}_l, \mathtt{T}_m) &\sim \mathcal{N}\left((\mathbf{R}_l\mathbf{p}_l^j + \mathbf{t}_l) - (\mathbf{R}_m\mathbf{p}_m^j + \mathbf{t}_m), \mathbf{R}_l\sigma_{3D}^2\mathbf{I}_3\mathbf{R}_l^\intercal + \mathbf{R}_m\sigma_{3D}^2\mathbf{I}_3\mathbf{R}_m^\intercal\right) \\
&= \mathcal{N}(\mathbf{0}_3, 2\sigma_{3D}^2\mathbf{I}_3).
\end{aligned}
\tag{4.3}
$$

Similarly, considering the noise free 2D measurements in (2.4), we have $\mathbf{a}_l^h(\mathbf{S}_l^h) \sim \mathcal{N}(\mathbf{0}_2, \sigma_{2D}^2\mathbf{I}_2)$. It is clear from (4.3) that since $\mathbf{b}_{l,m}^{j}(\mathbf{T}_l, \mathbf{T}_m)$, which is based on the ICP algorithm, uses two noisy 3D feature points, hence, the variance of the corresponding distribution is two times the variance of noise in each 3D feature point. This is in contrast to the variance of distribution corresponding to $\mathbf{a}_l^h(\mathbf{S}_l^h)$, which is based on the BA algorithm and uses only one noisy 2D feature point [83].

Using $\mathbf{b}_{l,m}^{j}(\mathbf{T}_l, \mathbf{T}_m)$ and $\mathbf{a}_l^h(\mathbf{S}_l^h)$, we want to find the likelihood cost function, maximum of which gives the Maximum Likelihood Estimate (MLE) of the parameters $\mathbf{S} = (\mathbf{T}, \mathbf{r})$. Since the MLE with Gaussian model is equivalent to the Least Squares Estimate (LSE) [85], we can directly get:

$$
\hat{\mathbf{S}} = \arg\min_{\mathbf{S}} \sum_{\substack{1<l,m<N \\ l\neq m}} \frac{1}{2\sigma_{3D}^2} tr\left(\mathbf{B}_{l,m}^\intercal(\mathbf{T}_l, \mathbf{T}_m)\mathbf{B}_{l,m}(\mathbf{T}_l, \mathbf{T}_m)\right) + \sum_{l=1}^{N} \frac{1}{\sigma_{2D}^2} tr\left(\mathbf{A}_l^\intercal(\mathbf{S}_l)\mathbf{A}_l(\mathbf{S}_l)\right).
\tag{4.4}
$$

Therefore the total cost to be minimized is:

$$
V(\mathbf{S}) = V_{ICP}(\mathbf{T}) + wV_{BA}(\mathbf{S}),
\tag{4.5}
$$

where $w = \frac{2\sigma_{3D}^2}{\sigma_{2D}^2}$ is the weighting factor. The cost function in (4.4) optimally combines information from RGB and depth sensors, to be used in the pose refinement scheme, by taking into account the noise levels in the 2D and 3D points. It formally defines the the relationship of measurement noise in the 2D and 3D feature points with the weighting factor w. In case the assumption of noise with same variances affecting all 2D and 3D points respectively, does not hold and information about the noise variances affecting each point is available, it can be incorporated in the proposed framework. Moreover, the use of the ICP based cost also allows the use of all the 3D points acquired by each sensor (with the help of nearest neighbor correspondence) in the optimization scheme when only 2D feature points are available.

The cost function (4.5) is a non-linear function of the parameters $\mathbf{S}$ and we resort to numerical search methods [86] to optimize the criterion. Please refer to Appendix A for further discussion.

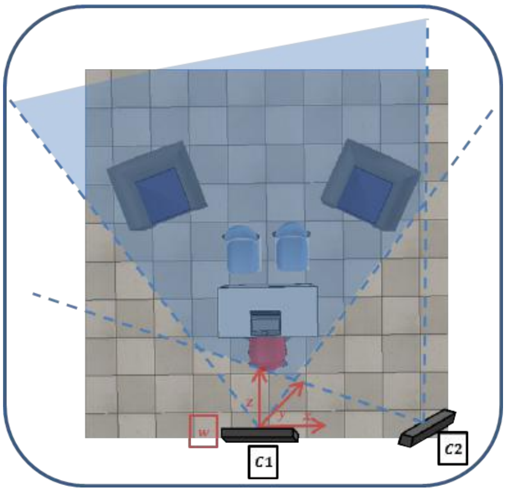


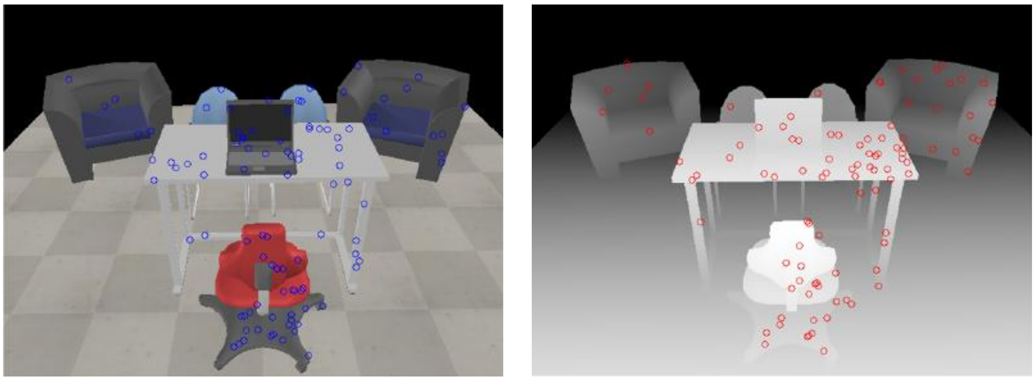
FIGURE 4.2: RGB-D Multi-View System (2 cameras) with field of view (FOV) of each camera. The highlighted region represents overlapping FOVs of all cameras. The global reference frame `w` is aligned with camera ***C*1**.

4.4 Weighting Factor Estimation

In this section we discuss the automatic and simultaneous estimation of the weighting factor w in (4.5), together with the camera poses in the absence of information regarding noise affecting both the 2D and 3D measurements. We propose an approach which alternates between camera pose estimation and estimation of the 2D and the 3D noise variances to arrive at a suitable solution.

In the previous section, the estimates of camera pose parameters and 3D points in `w` corresponding to 2D feature points, were computed based on known 2D and 3D feature points and the noise affecting them. We assumed the presence of Gaussian noise with zero mean and variances of σ^2_{2D} and σ^2_{3D} in 2D and 3D measurements, respectively. These parameters, in turn, define the weighting factor w which is instrumental in constructing the sensor fusion framework by optimally combining the 2D and 3D cost functions to estimate the camera poses. In real-world scenarios, however, information about the noise affecting one or both sensor measurements is often unavailable. This makes the computation of a correct w difficult. As mentioned in Section 4.1, researchers have tried to estimate the optimal weighting factor, for their proposed bi-objective schemes, for solving mainly the pair-wise pose estimation problem. The commonly used used methods range from using simple heuristic measures such as in the case of [46] to more complex methods, based on analysis of trade-off between residuals of two cost functions and based on learning via cross-validation, such as in the case of [83].

In this work, we propose to use a simple method for automatic estimation of the weighting factor w which finds its basis in finding the MLE of noise variances, σ^2_{2D} and σ^2_{3D},



(a) Features extracted from RGB image (b) Features extracted from depth image

FIGURE 4.3: Features extracted from RGB and depth images of camera $\boldsymbol{C1}$ in the multi-view system composed of 2 cameras as shown in Figure 4.2. The extracted feature points are also visible to camera $\boldsymbol{C2}$.

using the 2D and 3D feature points together with the current estimates of camera poses and 3D points in $\hat{\mathbf{S}}$. The MLE of the variance σ_{3D}^2 is given as [85]:

$$\hat{\sigma}_{3D}^2 = \sum_{\substack{1<l,m<N \\ l \neq m}} \frac{tr(\mathbf{B}_{l,m}^{\intercal}(\mathbf{T}_l, \mathbf{T}_m)\mathbf{B}_{l,m}(\mathbf{T}_l, \mathbf{T}_m))}{2a}, \tag{4.6}$$

where a is the total number of 3D feature correspondences across all views. Similarly, the MLE of the variance σ_{2D}^2 is computed via:

$$\hat{\sigma}_{2D}^2 = \sum_{l=1}^{N} \frac{tr(\mathbf{A}_l^{\intercal}(\mathbf{S}_l)\mathbf{A}_l(\mathbf{S}_l))}{b}, \tag{4.7}$$

where b is the total number of 2D feature points found across all views.

We follow an iterative approach whereby using 2D and 3D feature points and an initial estimate $\hat{\mathbf{S}}$, the MLE estimates of noise variances and hence of w are obtained via (4.6) and (4.7). This initial estimate of w is then used to find an updated estimate of $\mathbf{S}$ using (4.5) via non-linear optimization which, in turn, is used to update the estimate of w. This process is repeated for a fixed number of iterations until the estimates of $\mathbf{S}$ and w converge.

4.5 Experiments with Synthetic Data

In this section, we carry out a quantitative performance analysis of the proposed bi-objective refinement with a known and an unknown weighting factor.

4.5.1 Evaluation Methodology and Parameters

We use V-REP [87] to simulate 2 and 4 cameras based RGB-D multi-view systems, with overlapping FOVs, as shown in Figure 4.2 and Figure 4.1, respectively. In both cases, the global reference frame `w` lies in camera $\boldsymbol{C1}$. We simulate a scene containing several objects such as chairs, a table, sofas etc. The acquired noise-free data, in the form of RGB and depth images, is assumed to be perfectly mapped in each camera's RGB sensor's reference frame with known intrinsics. After data acquisition, random points, visible to all cameras, are extracted as feature points in both RGB and depth images as shown in Figure 4.3 (points on the floor are discarded). Features extracted from depth maps are converted to the corresponding 3D points via known intrinsics.

In the next step, noise is added to the extracted 2D and 3D feature points. We assume the presence of independent Gaussian noise in each coordinate of position of 2D feature points with zero mean and standard deviation σ_{2D} similar to [88]. The value of σ_{2D} is varied between 0.2 to 1.8 pixels with a step size of 0.4 pixels. Depth sensor measurements in RGB-D cameras suffer from different types of systematic and non-systematic errors as investigated in [81, 89]. For our scheme we propose to counter, beforehand, the systematic errors in depth measurements of each camera via a correction step, based on comparing known and measured depths [5, 43]. Therefore, for all remaining errors we assume the presence of additive independent Gaussian noise in each coordinate of 3D feature points in each view with zero mean and standard deviation σ_{3D}. The value of σ_{3D} is varied between 6 to 30 mm with a step size of 6 mm to keep it in the range of errors computed in [89].

We test the performance of the proposed scheme under various conditions by varying the number of cameras and their positions as shown in Figure 4.2 and Figure 4.1, by varying the noise magnitude in 2D and 3D feature points as explained above, and by varying the number of 2D and 3D feature points. For each configuration, 50 noise realizations are generated. For each noise realization, 2D feature points and their corresponding noisy 3D measurements from vertex maps are used to initialize the pose estimates via a Direct Linear Transform (DLT) based approach [40, 90]. Using the initial pose estimates, optimization is carried out via the proposed scheme, with known noise parameters as explained in Section 4.3, and with unknown noise parameters using the automatic iterative estimation scheme as explained in Section 4.4 (required 3 iterations to converge in most cases). Furthermore, optimization is also carried out via ICP algorithm using 3D feature points only, and via BA algorithm using 2D features points only.

Accuracy of the estimated poses is computed by comparison with the ground truth poses as done in [88]. Two measures of accuracy are computed. First is the angular magnitude of residual rotation computed via $\hat{\mathbf{R}}_l^T\mathbf{R}_l$, and second is the relative translation error which is computed via $\frac{\|\hat{\mathbf{t}}_l-\mathbf{t}_l\|}{\|\mathbf{t}_l\|}$. The results of 50 realizations showing the accuracy, of each initialization and of each refinement approach, for each configuration are plotted by using the function *boxplot* in MATLAB as shown in Figure 4.4 - 4.12. The horizontal line inside each box marks the median, the edges mark the 25th and the 75th percentiles, the whisker edges show most extreme data points with outliers plotted separately as red crosses.

The implementation of the proposed bi-objective optimization scheme and ICP is based on the non-linear optimization via Levenberg Marquardt (LM) algorithm [91], while the implementation of BA is based on a sparse variant of the LM algorithm called Sparse Bundle Adjustment (SBA) [41, 92].

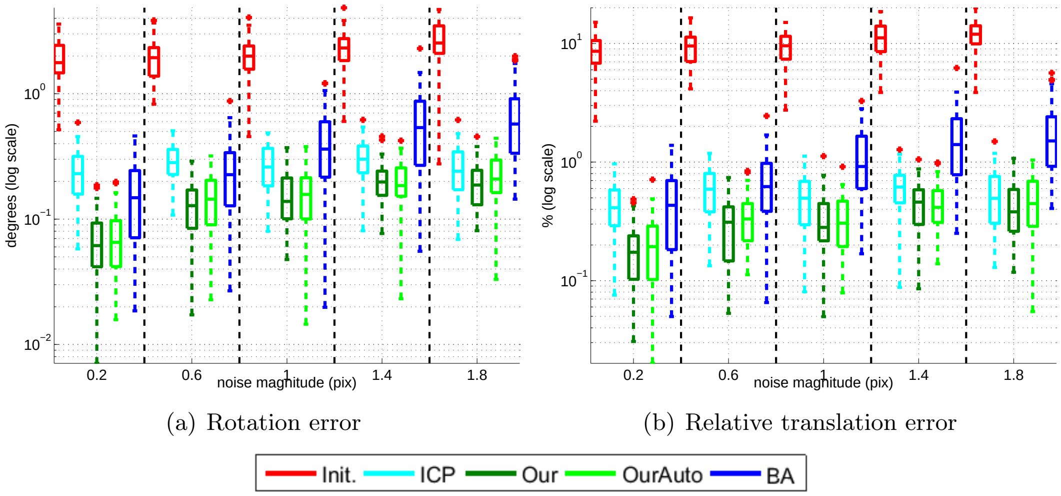
4.5.2 System Composed of Two Sensors

This section compares the performance of the proposed bi-objective optimization scheme, with known and unknown weighting factor, ICP and BA for refinement of camera pose parameters in a two camera setup shown in Figure 4.2. The pose of camera $\boldsymbol{C2}$ with respect to camera $\boldsymbol{C1}$ is estimated. After initialization, pose refinement is carried out using the four refinement methods and results are plotted in Figure 4.4 - 4.8.

4.5.2.1 Varying Noise Levels

In this experiment, the extrinsic calibration is carried out using 100 2D feature points and 100 3D feature points. Figure 4.4 shows the error distribution for fixed 3D noise and varying 2D noise, while Figure 4.5 shows the distribution in case the 2D noise is kept fixed, and the 3D noise is varied.

As expected, the accuracy of the extrinsic calibration decreases with increasing noise levels. Also, all pose refinement approaches are able to improve the initial pose estimates, explained by the fact that only inlier data points are generated (no wrong matching feature points are included). A careful analysis of the results shows that our bi-objective optimization scheme with known w, which uses simultaneously the 2D and 3D data, provides better pose estimations when compared to ICP and BA, where only 3D feature points and 2D feature points are used, respectively. The weighting factor based on the noise variance information in (4.5) automatically gives prominence to more reliable data, decreasing the impact of the other sensor modality. Moreover, it shows that our proposed automatic iterative estimation scheme used in the absence of information regarding noise



(a) Rotation error

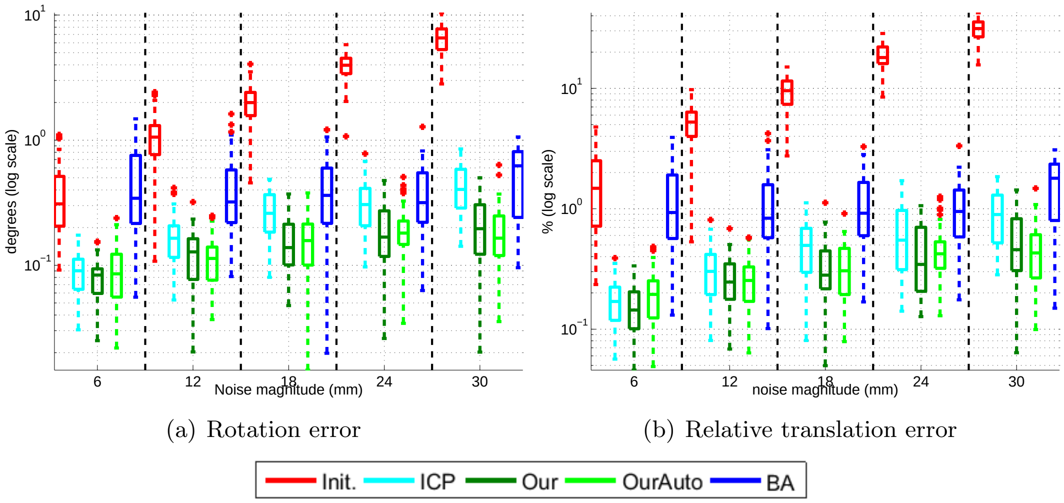
(b) Relative translation error

FIGURE 4.4: Error distribution of pose estimates for camera $\boldsymbol{C2}$ in a two camera setup. 100 2D and 100 3D feature points are used. The following methods are compared: Init. - Initial pose obtained using a DLT like approach (2D feature points and corresponding 3D points are used) [40, 90]; ICP - refinement of Init. using Iterative Closest Point (only 3D feature points are used); Our - refinement of Init. using our bi-objective optimization with known w (2D and 3D feature points are used); OurAuto - refinement of Init. using our bi-objective optimization with unknown w (2D and 3D feature points are used); BA - refinement of Init. using Bundle adjustment (only 2D feature points are used). Gaussian noise is added to the data, being the variance of the 3D noise fixed ($\sigma_{3D} = 18$mm), and the 2D noise σ_{2D} is varied between 0.2 and 1.8 pixels (horizontal axes).

parameters, and hence unknown w, is robust and also more accurate when compared to BA and ICP, and in most cases nearly as accurate as the method with known w.

4.5.2.2 Varying Data Ratio

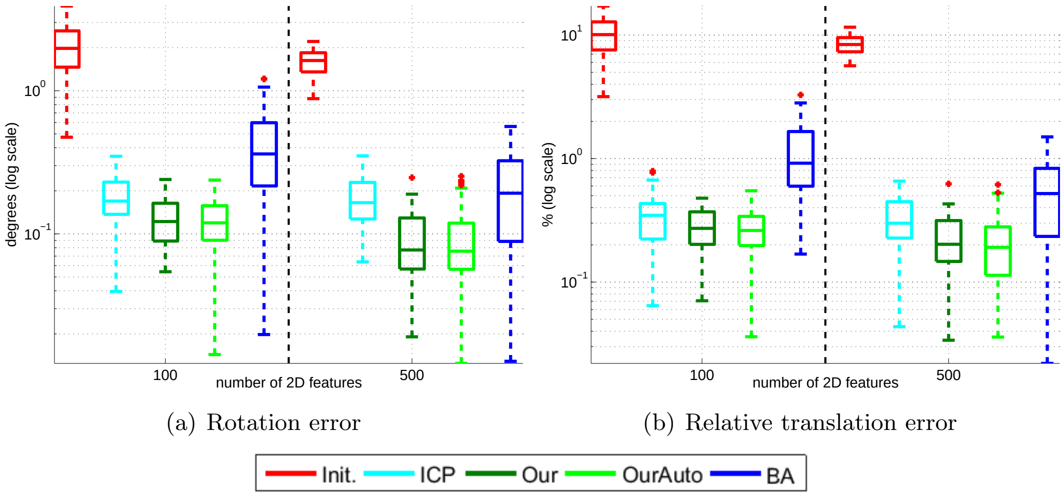
In this experiment, the extrinsic calibration is carried out using fixed noise variance ($\sigma_{2D} = 1$pix, $\sigma_{3D} = 18$mm). Figure 4.6 shows the error distribution for a fixed number of 3D points and a varying number of 2D points, while Figure 4.7 shows the distribution in case the 2D points are kept fixed, and the number of 3D points is varied. Since the initial poses are obtained by using 2D feature points and their corresponding 3D points, the initialization varies in Figure 4.6 as number of 2D feature points vary but stays approximately the same in Figure 4.7 as the number of 2D feature points remain fixed. The conclusions drawn in the previous section regarding improved accuracy of the proposed approaches hold, and these results show that the proposed scheme generalizes for different ratios between the number of 2D and 3D points. Increasing the number of data points of one of the sensor modalities always improves the extrinsic calibration accuracy for the algorithms using those modalities.



(a) Rotation error

(b) Relative translation error

FIGURE 4.5: Error distribution of pose estimates for camera $\boldsymbol{C2}$ in a two camera setup. 100 2D and 100 3D feature points are used. Gaussian noise is added to the data, being the variance of the 2D noise fixed (σ_{2D} = 1pix), and the 3D noise σ_{3D} is varied between 8mm and 30mm (horizontal axes).



(a) Rotation error

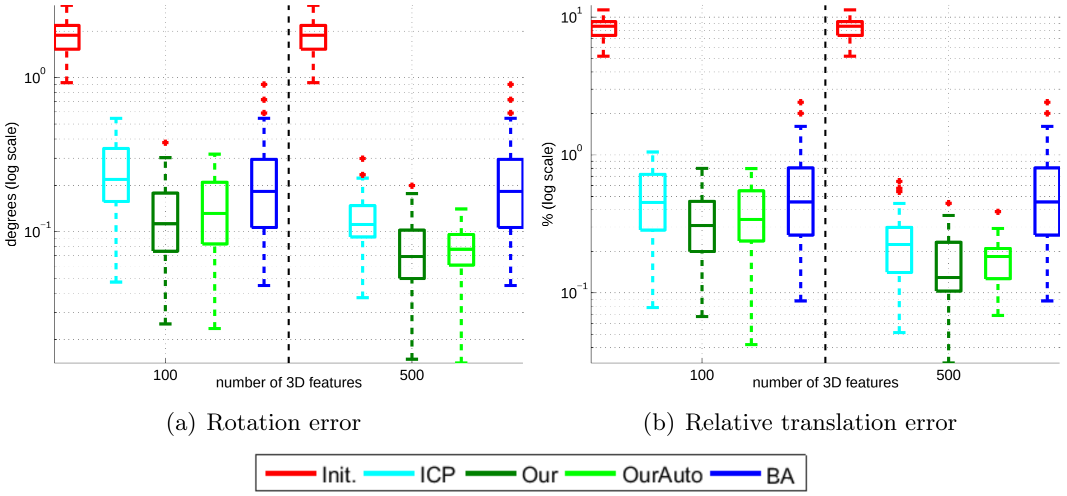
(b) Relative translation error

FIGURE 4.6: Error distribution of pose estimates for camera $\boldsymbol{C2}$ in a two camera setup. Gaussian noise is added to the data (σ_{2D} = 1pix, σ_{3D} = 18mm), 250 3D feature points and a varying number of 2D feature points (horizontal axes) is considered.

Moreover, the results in Figure 4.4, Figure 4.5, Figure 4.6, and Figure 4.7 show the increased robustness of the proposed approach and ICP to bad initialization as compared to BA.

4.5.3 System Composed of Four Sensors

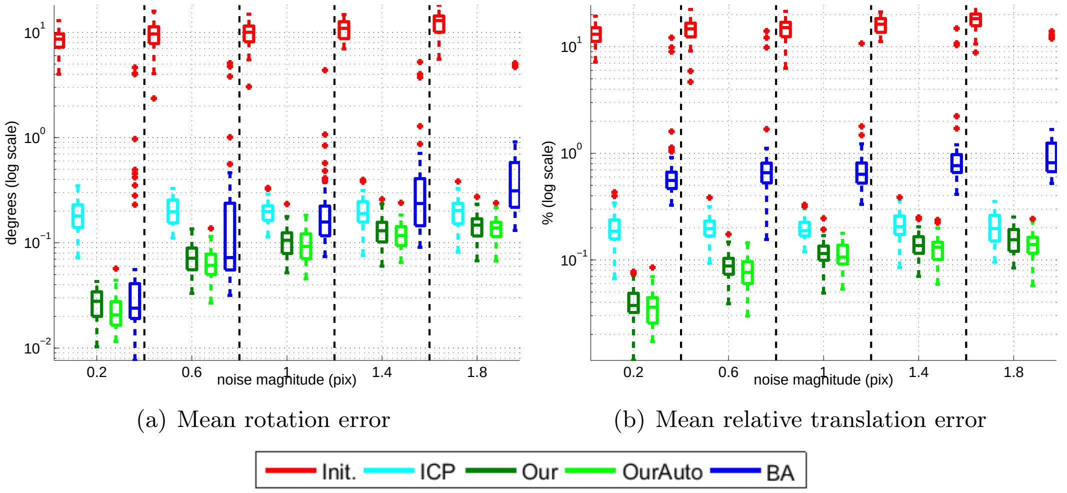
This section compares performance of the proposed bi-objective optimization scheme with ICP and BA for refinement of camera pose parameters in a four camera setup shown in Figure 4.1. The poses of cameras $\boldsymbol{C2}$, $\boldsymbol{C3}$ and $\boldsymbol{C4}$ are aligned with camera



(a) Rotation error

(b) Relative translation error

FIGURE 4.7: Error distribution of pose estimates for camera ***C*2** in a two camera setup. Gaussian noise is added to the data (σ_{2D} = 1pix),σ_{3D} = 18mm), 250 2D feature points and a varying number of 3D feature points (horizontal axes) is considered.



(a) Mean rotation error

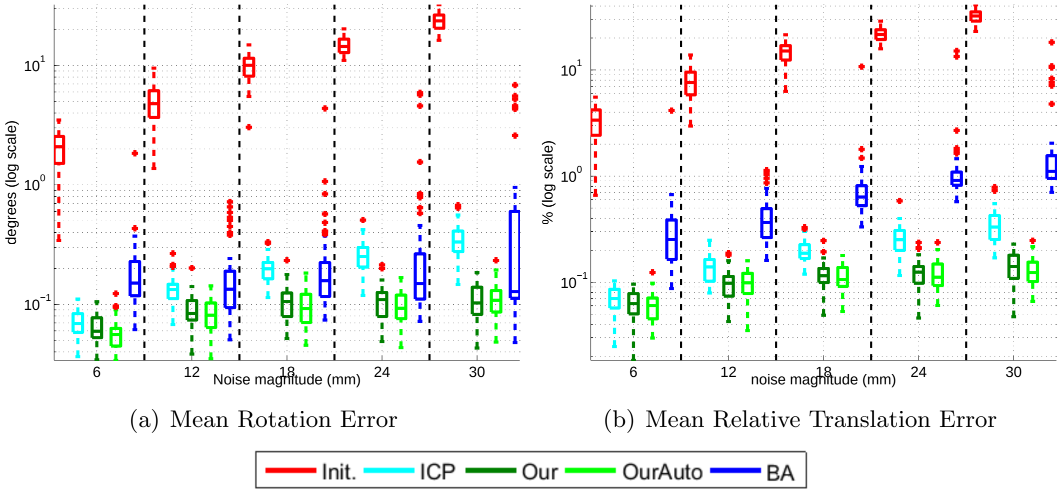
(b) Mean relative translation error

FIGURE 4.8: Mean error distribution, of pose estimates for cameras ***C*2**, ***C*3** and ***C*4**. in a four camera setup. 100 2D and 100 3D feature points are used. Gaussian noise is added to the data, being the variance of the 3D noise fixed (σ_{3D} = 18mm), and the 2D noise σ_{2D} is varied between 0.2 and 1.8 pixels (horizontal axes).

***C*1**. After initialization, pose refinement is carried out using the four refinement methods and results are plotted.

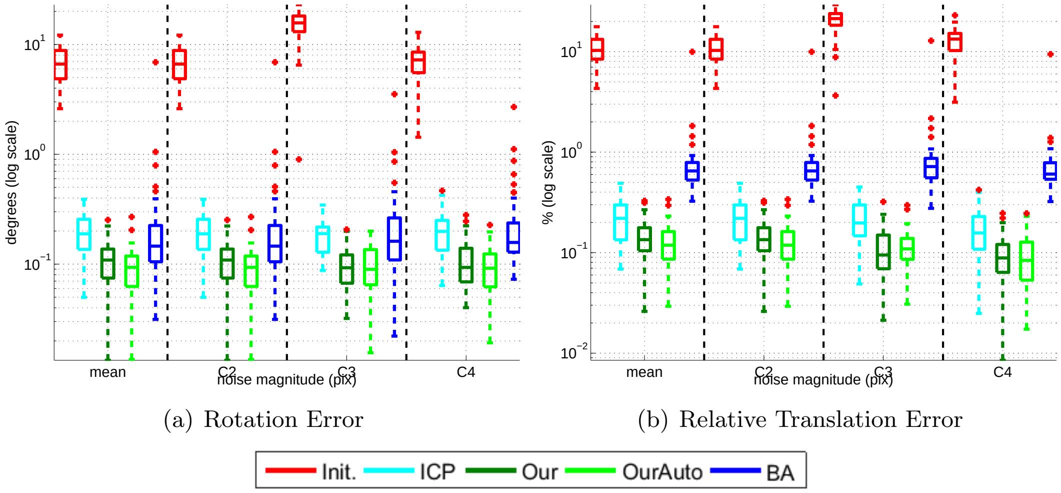
4.5.3.1 Varying Noise Levels

In this experiment, the extrinsic calibration is carried out using 100 2D feature points and 100 3D feature points. Figure 4.8 shows the mean error distribution for computed poses of all cameras, for fixed 3D noise and varying 2D noise. Figure 4.9 shows the mean distribution in case the 2D noise is fixed. These results again show the improved



(a) Mean Rotation Error (b) Mean Relative Translation Error

FIGURE 4.9: Mean Error distribution, of pose estimates for cameras $\boldsymbol{C2}$, $\boldsymbol{C3}$ and $\boldsymbol{C4}$. in a four camera setup. 100 2D and 100 3D feature points are used. Gaussian noise is added to the data, being the variance of the 2D noise fixed (σ_{2D} = 1pix), and the 3D noise σ_{3D} is varied between 8mm and 30mm (horizontal axes).

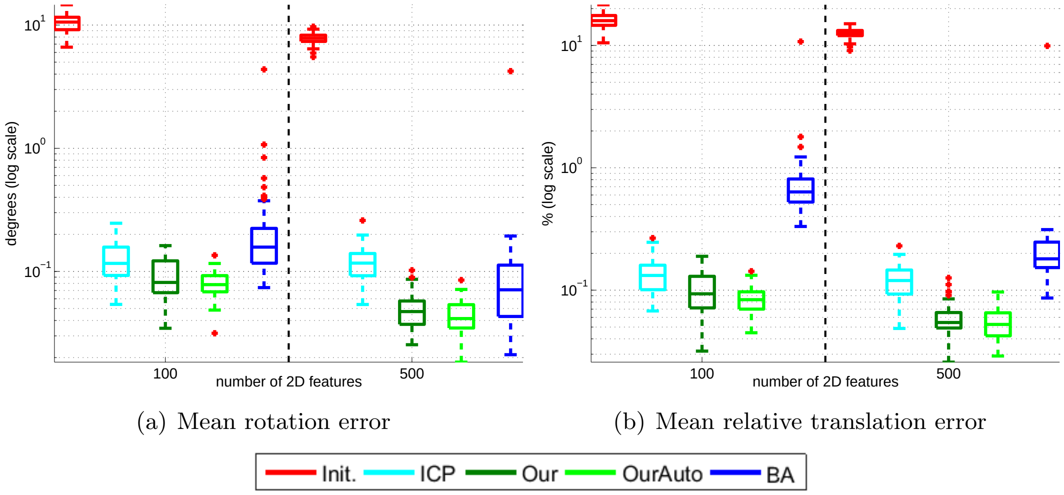


(a) Rotation Error (b) Relative Translation Error

FIGURE 4.10: Comparison of error distributions, of the extrinsic calibration of a four camera setup, using 100 2D and 100 3D feature points. The results are based on mean error distribution and error distribution for camera $\boldsymbol{C2}$, camera $\boldsymbol{C3}$ and camera $\boldsymbol{C4}$. Gaussian noise is added to the data, being the variance of the both 2D noise and 3D noise fixed (σ_{2D} = 1pix, σ_{3D} = 18mm).

performance of the proposed approaches due to the use of both 2D and 3D information together, with the help of correct weighting factor. The performance of all methods gets affected as the noise in 2D and 3D data increases. These results also show improvement in performance of all methods as compared to the multi-view system composed of two cameras due to increased number of 2D and 3D points available. Moreover, these results show that the proposed scheme generalizes for different numbers of cameras used in the multi-view system.

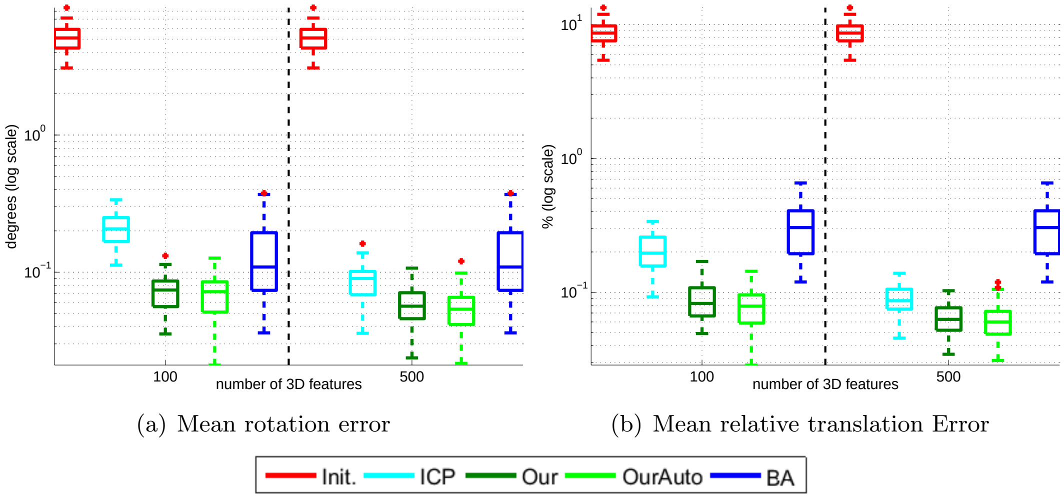
We also notice an interesting behavior where in some cases the proposed automatic



(a) Mean rotation error

(b) Mean relative translation error

FIGURE 4.11: Mean error distribution, of pose estimates for cameras $\boldsymbol{C2}$, $\boldsymbol{C3}$ and $\boldsymbol{C4}$. in a four camera setup. Gaussian noise is added to the data (σ_{2D} = 1pix), σ_{3D} = 18mm), 250 3D feature points and a varying number of 2D feature points (horizontal axes) is considered.



(a) Mean rotation error

(b) Mean relative translation Error

FIGURE 4.12: Mean error distribution, of pose estimates for cameras $\boldsymbol{C2}$, $\boldsymbol{C3}$ and $\boldsymbol{C4}$. in a four camera setup. Gaussian noise is added to the data (σ_{2D} = 1pix), σ_{3D} = 18mm), 250 2D feature points and a varying number of 3D feature points (horizontal axes) is considered.

iterative scheme based on alternative computation of camera poses and w gives better results compared to the scheme with known w. Apart from increase in the number of measurements per feature point, a reason for this can be that for the case of known w we are assuming that for all the 2D and 3D feature points the variances of noise affecting them are the same and constant; but depending on a particular realization, the noise will be a bit higher or lower than the fixed value. Therefore the automatic procedure which tries to compute the variances directly from the noisy data is, in many cases, better able to capture the noise characteristics. For BA, Figure 4.9 shows a decrease in its performance as the 3D noise increases. The reason being that apart from its dependence on the initial camera poses, the initial guess of the 3D points corresponding to 2D feature points also gets worse due to increased 3D noise.

In Figure 4.10, we compare the mean error distribution with error distributions of individual cameras for the single case of 2D and 3D noise variance ($\sigma_{2D} = 1$pix, $\sigma_{3D} = 18$mm). These results show that while the initial guess for camera $\boldsymbol{C3}$ is comparatively worse, the performance of optimization schemes is comparable across all views.

4.5.3.2 Varying Number of Points

In this experiment, the extrinsic calibration is carried out using a fixed noise variance ($\sigma_{2D} = 1$pix, $\sigma_{3D} = 18$mm). Figure 4.11 shows the mean error distribution for a fixed number of 3D points and a varying number of 2D points. Figure 4.12, on the other hand, shows the mean distribution in case the 2D points are kept fixed, and the number of 3D points are varied. Here, again, the conclusions drawn in the previous sections hold, while also showing that increasing the number of data points of one of the sensor modalities always improves the extrinsic calibration accuracy for the methods using those modalities.

4.6 Experiments with Real Data

In this section, we carry out a qualitative performance analysis of the proposed bi-objective refinement scheme using a real setup. Our setup consists of 4 Asus Xtion Pro Live cameras [19] with their positions shown in Figure 4.13. Each camera acquires an RGB image and a depth image which is mapped to the RGB image.

The first step is to perform intrinsic calibration to find the intrinsic and distortion parameters for each camera. For this purpose, we use the method proposed by Zhang [39] which uses 2D corners extracted from RGB images of a checkerboard pattern viewed at different poses to compute these parameters [40]. As mentioned before, the measurements of these RGB-D cameras suffer from inherent depth bias. Therefore, we perform a depth bias correction procedure, similar to the one used in [43], for each camera separately. This procedure requires placing the camera at known distances away from an object (a plane in our case). Using known and measured depth values, we estimate the coefficients of a polynomial which computes the depth correction as a function of measured depth value. These coefficients are unique to each camera and, hence, are used to correct the depth measurements acquired by that camera.

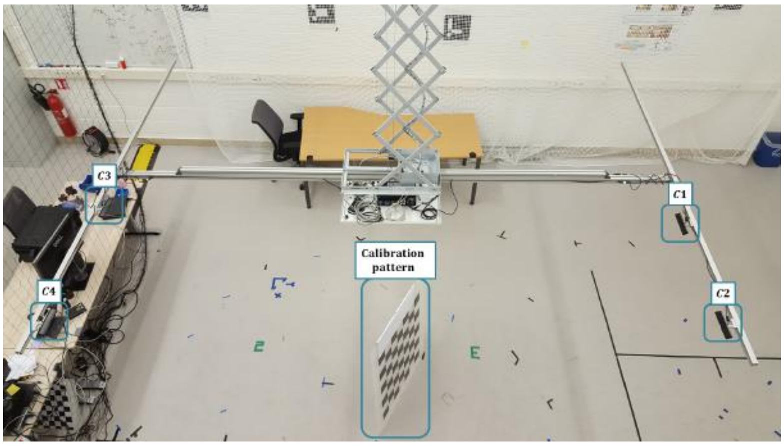


FIGURE 4.13: Multi-view system consisting of 4 Asus Xtion Pro Live Cameras $\boldsymbol{C1}$, $\boldsymbol{C2}$, $\boldsymbol{C3}$ and $\boldsymbol{C4}$ mounted on a ceiling lift. This system is used to acquire measurements of a real scene. A two-sided planar checkerboard calibration pattern used to extract feature points is also shown.

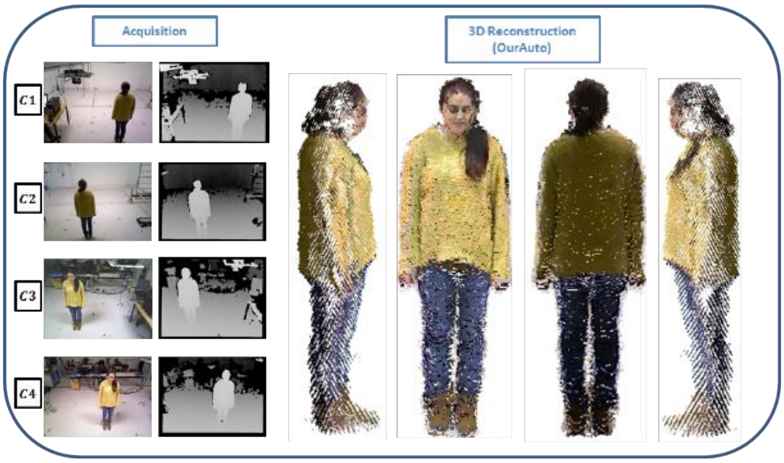


FIGURE 4.14: 3D reconstruction of a human using a real scene acquired from the multi-view system shown in Figure 4.13. Acquisition: Each of the 4 cameras acquire an RGB image and a depth image. 3D Reconstruction: Point clouds based 3D reconstruction using pose estimates refined by the proposed bi-objective scheme with the help of automated weighting.

After intrinsic calibration and depth bias correction, the next step is to perform the extrinsic calibration using the proposed bi-objective scheme. We first need to extract matching 2D and 3D feature points using RGB and depth images acquired by all 4 cameras. We again use different views of a (two-sided) planar checkerboard pattern as shown in Figure 4.13 and extract matching corners from RGB images to be used as 2D feature points and use the corresponding depth values from depth images to get the 3D feature points. The 3D feature points are filtered via a plane detection approach based on RANSAC algorithm to remove outliers if any exist. The initial pose estimates are generated in the same manner as explained in Section 4.5, via a Direct Linear Transform (DLT) based approach [40, 90]. These initial poses are then refined via the proposed iterative pose estimation and weight estimation approach explained in Section 4.4, BA

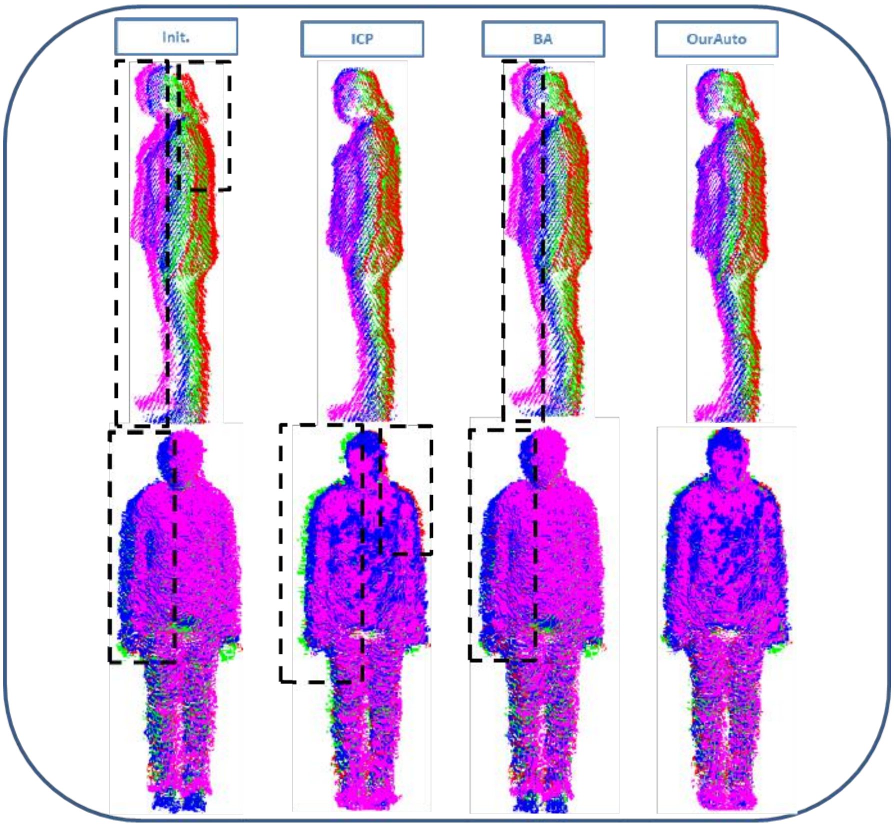


Figure 4.15: Comparison of 3D reconstructions of a human using a real scene as shown in Figure 4.13, via different calibration methods namely Init., ICP, BA and OurAuto described in Figure 4.5. The acquisitions from cameras $\boldsymbol{C1}$, $\boldsymbol{C2}$, $\boldsymbol{C3}$ and $\boldsymbol{C4}$ are assigned the colors red, green, blue and magenta, respectively. Misalignments are highlighted via black boxes. *Top Row* shows side view of the 3D reconstruction and misalignment of views in the results of Init. and BA can be seen clearly, while *Bottom Row* shows the frontal view and misalignment of views in the results of Init., ICP and BA are visible. It can also be seen that OurAuto gives better results compared to the other methods.

and ICP. Once the refined poses are obtained, they can be used to produce full, textured, 3D reconstructions using data acquired by all 4 cameras as shown in Figure 4.14. A qualitative comparison of 3D reconstructions obtained via different calibration methods is shown in Figure 4.15. It can be seen that the partial reconstructions are better aligned using the proposed method, which means that the quality of the extrinsic calibration is superior when compared to the other approaches. Note that we are only showing the alignment of the partial point clouds, and no post-processing step such as smoothing or meshing are applied. We chose to do so to better assess, visually, the accuracy of the extrinsic calibration.

4.7 Conclusion

In this work we have proposed a framework for RGB and depth sensor fusion based on bi-objective optimization, for refinement of extrinsic calibration in RGB-D multi-view systems. Our bi-objective optimization scheme makes use of a cost function from the BA algorithm for 2D feature points extracted from RGB images and a cost function from the ICP algorithm for 3D feature points extracted from depth images. We analytically derive an expression for the weighted bi-objective cost function. It also analytically relates the weighing factor to the noise in the 2D and 3D measurements, thus making the cost function free of any parameter that needs to be tuned. In case the information regarding measurement noise in 2D and 3D data is not available, we propose an iterative scheme which alternates between estimation of noise parameters assuming known poses, and estimation of camera poses assuming known noise parameters. Thus, it enables us to automatically compute the correct weighting factor when information about measurement noise is not available. A thorough investigation of the performance of the proposed approach for both synthetic and real data showed improved accuracy compared to refinement schemes which only use 2D or 3D information, and comparative performance of proposed approaches with known and unknown noise parameters. These experiments also showed the invariance of the proposed approach under various conditions which include varying the number and position of cameras, varying the 2D and 3D noise and varying the number of the 2D and 3D feature points.

Chapter 5

KinectDeform: Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects

In this part of the thesis we turn our attention to research, analysis, and development of methods which target online and template-free enhancement of noisy 3D data acquired with commodity 3D cameras. Our focus is on reconstructing scenes which contain non-rigid objects undergoing generic local deformations. For this purpose we propose KinectDeform, a recursive method which targets enhanced 3D reconstruction of dynamic scenes containing non-rigid objects. It provides an innovation to the existing class of mono-view algorithms which either target scenes with rigid objects only or allow for very limited local deformations or use precomputed templates to track them. KinectDeform combines a fast non-rigid scene tracking algorithm based on octree data representation and hierarchical voxel associations with a recursive data filtering mechanism. A performance analysis on real and simulated data shows that KinectDeform is able to produce smoothness and feature preserving 3D reconstructions with reduced noise.

5.1 Introduction

Reconstructing real objects accurately and efficiently is one of the major goals in the field of 3D computer vision. It opens doors to various applications from object detection to environment mapping, from gesture control to security and surveillance etc. Commodity depth cameras such as recently available structured light and time-of-flight cameras,

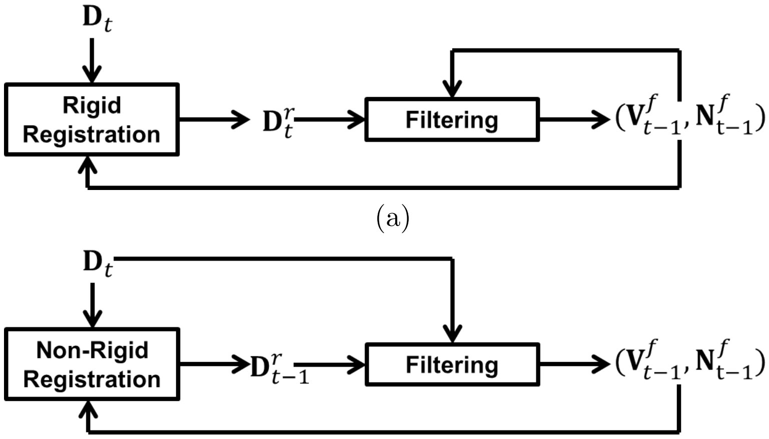


Figure 5.1: High-level pipeline of: (a) KinectFusion, and (b) the proposed KinectDeform. $\mathbf{D}_t$: input depth map at time t, $(\mathbf{V}_{t-1}^f, \mathbf{N}_{t-1}^f)$: filtered vertex map and corresponding normal map at time $t-1$, $\mathbf{D}_t^r$ and $\mathbf{D}_{t-1}^r$: resulting depth maps of rigid and non-rigid registration steps correspondingly. For more details please see Section 2.4.2.1 and Section 5.2.

though affordable and easily accessible, acquire noisy measurements with limited resolution, and hence provide 3D representations which are only suitable for a limited number of applications.

Many recent approaches try to solve the problem of attaining improved 3D reconstruction of scenes or objects from low quality raw data [28, 31]. One approach which stands out due to its performance, efficiency, and high quality results is the KinectFusion algorithm by Newcombe et al. [1, 93]. It either uses a moving RGB-D camera or considers objects moving in front of a static camera to obtain their high quality 3D reconstruction. Figure 5.1 (a) shows the high-level pipeline of KinectFusion where a rigid alignment of 3D data captured during sequential time-steps is followed by filtering or fusion of data accumulated over time. The key feature of KinectFusion is its run-time performance by using commodity graphics hardware, such that it is able to fuse and reconstruct data acquired at a rate which is as high as 30 frames per second in real-time.

KinectFusion became a cornerstone for various works which either built on it or used similar ideas, e.g., to map larger environments in one go by using a moving volume approach [47, 48], or by using octrees for memory efficient surface reconstruction [94, 95], or by using voxel hashing for even better accuracy and efficiency [33]. Kainz et al. modified the KinectFusion pipeline in order to incorporate multiple cameras for holistic 3D reconstruction of static objects [2]. Cerqueira et al. customized KinectFusion for real-time tracking and modeling of a human face [34]; whereas Sturm et al. used its components for full human body 3D reconstruction [36]. Moreover, improvements were also proposed in the real-time tracking module and pose computation by directly fusing depth maps with the truncated signed distance function (TSDF) volume [32], or by using visual features together with 3D information [47, 48, 94]. Similarly, textured 3D models were achieved by mapping visual texture information on the reconstructed 3D models [47, 48].

A limitation of the techniques mentioned above is that they target environments with rigid objects. This makes tracking such objects relatively simple by merely calculating a single global transformation for the whole object or scene. Non-rigid objects in otherwise rigid scene are considered as unstable regions, they are segmented and removed when detected [95, 96]. In the application of face modeling, facial expressions are required to be as consistent as possible throughout the scanning period [34]. Similarly, for full-body 3D reconstruction, the person to be scanned is required to be static with small non-rigidities handled by using a rough template from the first frame [36]. For the same body scanning applications, Cui et al. on the other hand, proposed to tackle non-rigidities by using a global non-rigid alignment based on joint constraints. Their technique however cannot handle large motions, and is also not very practical for real-time applications [35]. Recently, Zöllhoefer et al. [29] have proposed what they claim to be the first 'general purpose' non-rigid 3D reconstruction system which works in real-time and produces refined 3D reconstructions. It works by first acquiring a rigid template of the object to be reconstructed. This template is then used to track non-rigidities with high flexibility.

In this work, we propose a framework which is derived from KinectFusion with the ability to track and reconstruct, with high accuracy, without any template or constraint on motion, rigid as well as non-rigid moving objects. Figure 5.1 (b) shows the high-level pipeline of the proposed technique. Our key contributions consist of using tracking based on non-rigid registration of the result of the previous time-step to the newly acquired deformed data, followed by a recursive filtering mechanism based on the registered result and the newly acquired data. We make use of a generic tracking algorithm for non-rigid alignment which is efficient and can be easily parallelized [97]. We use both real and simulated data to validate the performance of the proposed technique.

5.2 Proposed Approach

In Section 2.4.1 we have formulated the problem of template-free recursive data fusion to obtain noise-free enhanced 3D reconstructions of the scene using the data acquired via a commodity depth camera. The KinectFusion algorithm which provides a solution to this problem under the constraints of global motion has been described in Section 2.4.2.1.

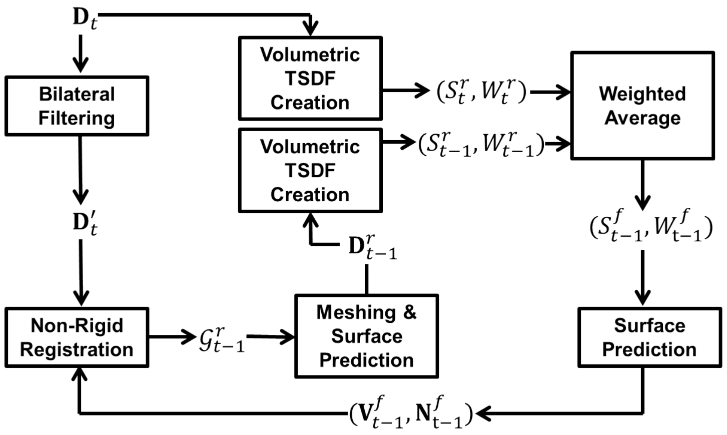
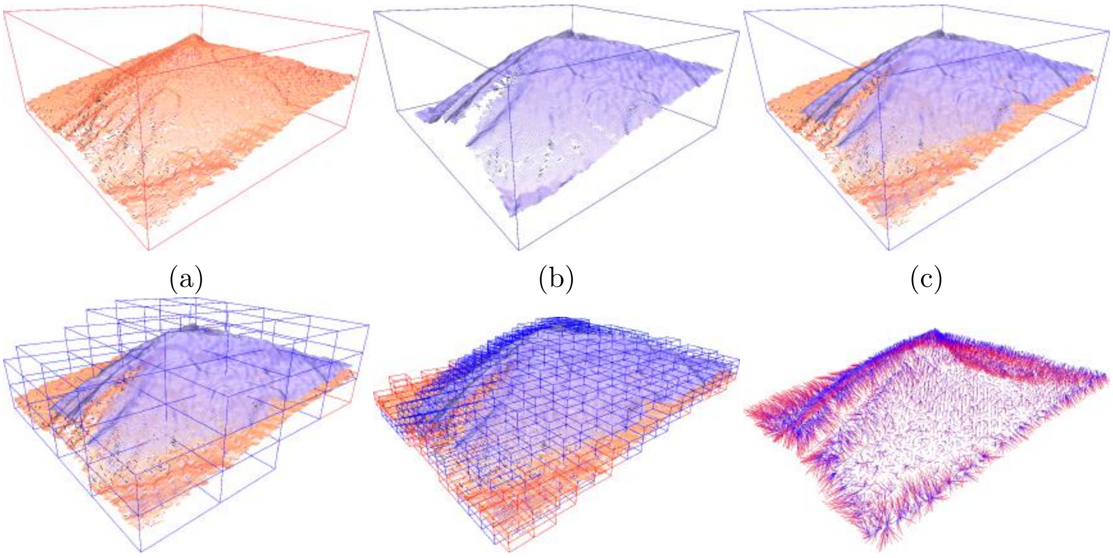


FIGURE 5.2: Detailed pipeline of the proposed KinectDeform. $\mathbf{D}_t$: input depth map at time-step i, $\mathbf{D}_t^{'}$: result of bilateral filter on $\mathbf{D}_t$, $(\mathbf{V}^f_{t-1}, \mathbf{N}^f_{t-1})$: filtered vertex map and corresponding normal map at time-step $i-1$, $\mathbf{G}^r_{t-1}$: unorganized point cloud which is the result of non-rigid registration of $\mathbf{V}^f_{t-1}$ to $\mathbf{D}_t^{'}$, $\mathbf{D}^r_{t-1}$: depth map corresponding to $\mathcal{G}^r_{t-1}$, $(S^r_t, W_{\mathbf{V}_t})$, $(S^r_{t-1}, W_{\mathbf{V}^r_{t-1}})$ and $(S^f_{t-1}, W_{\mathbf{V}^f_{t-1}})$ are TSDF volumes corresponding to vertex maps $\mathbf{V}_t$, $\mathbf{V}^r_{i-1}$ and $\mathbf{V}^f_{t-1}$ respectively. For more details please see Section 2.4.2.1 and Section 5.2.

We propose to modify the KinectFusion to achieve 3D tracking, and hence enhanced 3D reconstruction of not only rigid but also non-rigidly objects undergoing local deformations, as well. One of the main reasons for taking KinectFusion as a reference is its ease of parallelization for real-time implementation. We would like to maintain this feature in the proposed approach that we refer to as KinectDeform. As depicted in the high-level descriptions of Figure 5.1, KinectDeform modifies KinectFusion at two main levels; first, the registration which, from rigid, becomes non-rigid, and second, the reference frame in the filtering process changes where the newly acquired measurement is the one to act as a reference for the current state of the object and to which the resulting vertex map from the filtered TSDF from the previous iteration should be aligned and fused with. More details are provided in Figure 5.2, and described in what follows.

5.2.1 Non-Rigid Registration

Similarly to KinectFusion, for an improved registration, a bilateral filter is applied to the input depth map $\mathbf{D}_t$ as a first preprocessing step. We obtain a bilateral filtered depth map $\mathbf{D}'_t$, and its corresponding vertex map $\mathbf{V}'_t$. The next step is to register the resulting vertex map of the previous iteration, i.e., $\mathbf{V}^f_{t-1}$, with this new vertex map $\mathbf{V}'_t$. Conversely to other classical reconstruction methods, our pipeline captures non-rigid objects. As a consequence, this registration step aims to align two vertex maps describing the local deformation h_t in (2.9). This deformation is unknown but can be estimated locally by a patch-oriented method, describing the global non-rigid deformation by a set of local



(a) (b) (c)

Level 2 Level 4 Voxel Scene flow

FIGURE 5.3: Outline of the non-rigid registration algorithm used by our pipeline, from the first cloud (a) to the second one (b). As a first step, both clouds are mapped rigidly by centering their respective centroid (c). A common discrete space is then built using two separate octrees for which the root cell is the bounding box of the cloud couple. These octrees are then subdivided regularly until a fixed level S is reached. Finally, the algorithm described in [97] is used to create a voxel-to-voxel 3D scene flow, describing a global non-rigid deformation as a set of rigid ones.

rigid ones. As such, we propose to apply a modified scene-flow based tracking method from [97]. As opposed to other well-known techniques [30, 98–102], this algorithm offers real-time capabilities, and can handle non-rigidly deforming objects in a generic way without considering a specific motion or shape model. The proposed scene-flow tracking technique relies on several steps: the pair of vertex maps $\mathbf{V}^{f}_{t-1}$ and $\mathbf{V}'_{t}$ are first centered by joining their respective centroids. A double voxelization step then embeds each cloud considering as a first cell the bounding box of the two point clouds, i.e., sharing the same root cell. These octrees are aimed to be subdivided in a regular way considering each cut point as the cell center. Thus the subdivision of both clouds describes the same discrete coordinate space, see Figure 5.3. Then, a voxel-to-voxel scene flow is created using a local neighborhood relation among the voxels of the two octrees, several different hierarchical relations, and finally a local and computationally efficient algorithm to establish the relation from voxels of the first octree to the second one. KinectDeform uses the obtained voxel-to-voxel flow in order to register locally each point-based patch from $\mathbf{V}^{f}_{t-1}$, embedded in the first octree, to $\mathbf{V}'_{t}$, embedded in the second one. The result of the registration is $\mathcal{G}^{r}_{t-1}$, which is an unorganized 3D point cloud.

5.2.2 TSDF Volume Creation and Fusion

To create a TSDF volume using the approach explained in Section 2.4.2.1 from the information in $\mathcal{G}^r_{t-1}$, an organized point cloud or depth map needs to be extracted from it. An idea would be to simply back project points in $\mathcal{G}^r_{t-1}$ to the image plane using the camera matrix $\mathbf{K}$. This would result in several points in $\mathcal{G}^r_{t-1}$ being projected to the same pixel location in the image plane to which only one depth value is to be assigned. Hence, a lot of valuable information would be lost. To get a more accurate representation of $\mathcal{G}^r_{t-1}$ with respect to the camera, we perform surface reconstruction based on Delaunay triangulation [103]. The resulting mesh, is used for generating the depth map $\mathbf{D}^r_{t-1}$ by simulating a noise-free camera with the same pose and camera matrix $\mathbf{K}$ as the real camera used for acquiring the initial raw data and by performing ray-tracing [104]. Next step is to use the resultant depth map $\mathbf{D}^r_{t-1}$ and input depth map $\mathbf{D}_t$ to fuse them to get a filtered and enhanced reconstruction of the object at time t. Here again we use $\mathbf{D}_t$ for fusion and filtering instead of $\mathbf{D}'_t$ to avoid loss of important information due to bilateral filtering. For data fusion and filtering we also use the volumetric TSDF for surface representation as done by KinectFusion [1, 71]. The reason for choosing this representation scheme over other similar non-parametric representations is ease of surface extraction and parallelization of volumetric TSDF computation and fusion [1]. As mentioned in the begining of Section 5.2, for handling local deformations we cannot keep a globally consistent surface representation as reference and keep fusing newly acquired information to it. Instead we create TSDF volumes for both $\mathbf{D}^r_{t-1}$ and $\mathbf{D}_t$ using their corresponding $\mathbf{V}^r_{t-1}$ and $\mathbf{V}_t$ using (2.11) and (2.12) to get S^r_{t-1} and S_t, respectively.

We propose to modify the weighting scheme of KinecFusion in order to take the following factors into account. On one hand $\mathbf{V}^r_{t-1}$, which is the deformed version of $\mathbf{V}^f_{t-1}$, brings valuable information due to temporal filtering and also improved registration due to it being aligned to the filtered version of $\mathbf{V}_t$. On the other hand we also have to take into account errors during registration and also loss of some details in $\mathbf{V}'_t$ caused by bilateral filtering which in turn might cause loss of some details in $\mathbf{V}^r_{t-1}$. Similarly we should also consider the sensor or acquisition noise introduced in each acquisition $\mathbf{V}_t$. Therefore, to reflect these factors the weights W_t and W^r_{t-1} are initialized and updated as follows:

$$W_t(\mathbf{p}) = w(\sigma_c, \epsilon^s_t), \tag{5.1}$$

and

$$W^r_{t-1}(\mathbf{p}) = \begin{cases} w(\sigma_c, \epsilon^s_t) & \text{iff } t = 1, \\ w(\sigma_p, \epsilon^r_{t-1}) & \text{otherwise,} \end{cases} \tag{5.2}$$

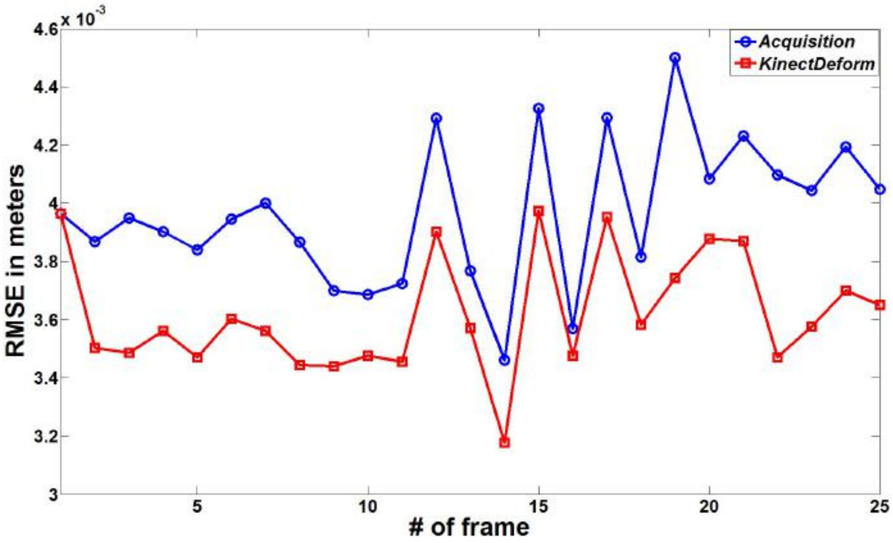


FIGURE 5.4: RMSE of raw and filtered data with ground truth for simulated "cloth" dataset

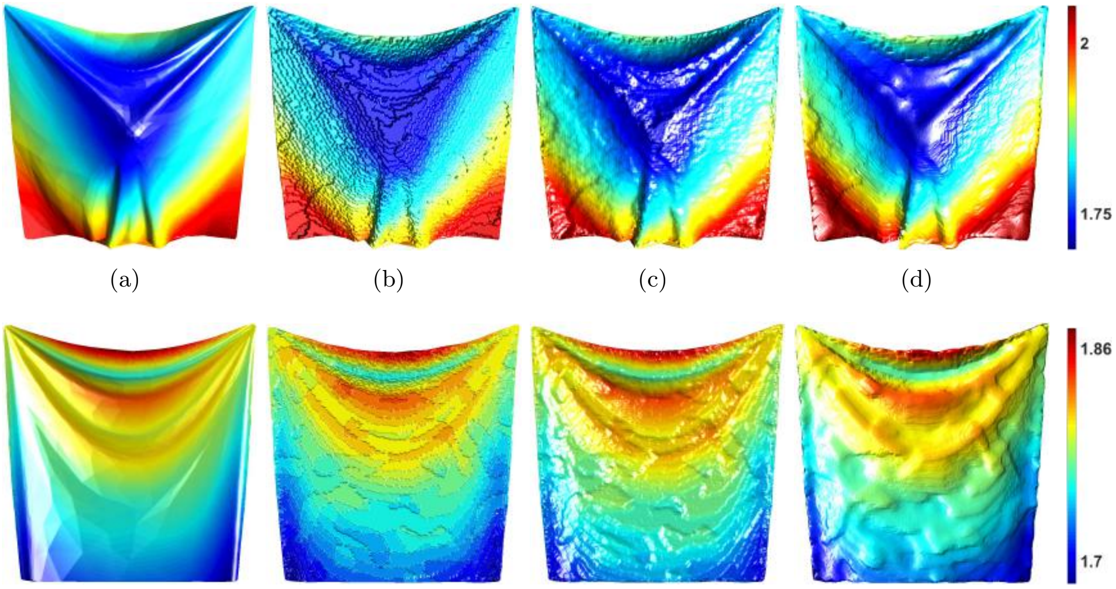
where $w(\sigma, x) = exp(-x^2\sigma^{-2})$ is the Gaussian weights function, σ_c and σ_p are standard deviations defining the Gaussian weight functions for current measurements and deformed results of previous iterations, respectively. ϵ_t^s is a global estimate of sensor noise in the current acquisition $\mathbf{D}_t$ and ϵ_{t-1}^r is defined as root-mean-square error (RMSE) based on point-wise Euclidean distances between $\mathbf{V}_t$ and $\mathbf{V}_{t-1}^r$:

$$\epsilon_{t-1}^r = \sqrt{\frac{1}{M}(\sum_{\mathbf{i}=1}^{M} \|\mathbf{V}_t^i - \mathbf{V}_{t-1}^{(r,i)}\|^2)}, \tag{5.3}$$

where M is the total number of points in $\mathbf{V}_t^i$, and ϵ_t^r is an estimate of the registration error and details lost during bilateral filtering, meshing and back projection in $\mathbf{V}_{t-1}^r$ with respect to $\mathbf{V}_t$ assuming that bilateral filtering removes the sensor noise from $\mathbf{V}_t^{'}$ and hence from $\mathbf{V}_{t-1}^r$. The parameters σ_c and σ_p are chosen empirically for now, taking into account the factors mentioned above by giving a higher weight to the temporally filtered deformed data compared to the raw input with increasing time. The two newly created volumes are fused by following (2.14) to get the filtered TSDF volume S_t^f which is used to extract the vertex map $\mathbf{V}_t^f$ and the normal map $\mathbf{N}_i^f$ for the next iteration using the same method as KinectFusion.

5.3 Experiments and Results

To analyze the performance of KinectDeform both quantitatively and qualitatively, we test it on both simulated and real non-rigidly deforming depth sequences. For quantitative analysis, we use two different data sources. The first one is the simulated deforming



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

(f) (g) (h) (i)

FIGURE 5.5: "Cloth" dataset. **Top row:** Frame 5 (a) Ground truth, (b) raw data, (c) result of KinectDeform, (d) result of KinectDeform after deblurring. **Bottom row:** Frame 20 (e) Ground truth, (f) raw data, (g) result of KinectDeform, (h) result of KinectDeform after deblurring. Display color-scale is based on the depth values of the 3D points and is in the units of meters.

"cloth" dataset acquired using the ArcSim simulator [105, 106], as shown in Figure 5.5. The second one is the high quality "facecap" dataset which was provided courtesy of the research group of Graphics, Vision & Video of the Max-Planck-Institute for Informatics [107], as shown in Figure 5.7.

In order to create Kinect v1 based acquired raw data, we simulate a realistic acquisition of the "cloth" sequence using Blensor by placing the camera at a distance of $1.8m$ [104]. We have used a sequence of 25 frames from this dataset. This noisy data is then filtered in KinectDeform with $\sigma_c = 18.5mm$ and $2.25mm \leq \sigma_p \leq 6.55mm$. From Blensor we can get an estimate of the sensor noise ϵ_n. The simulated noisy data and results of KinectDeform are compared with the ground truth data to compute RMSE based on Euclidean distances with nearest neighbors using CloudCompare [108]. The quantitative and qualitative improvements due to KinectDeform are shown in Figure 5.4. For qualitative evaluation we compare the reconstructions of frames 5 and 15 obtained using KinectDeform with the ground truth and the raw acquisitions as shown in Figure 5.5. Figure 5.5 (d) and Figure 5.5 (h) show the results of applying a deblurring filter on the results of KinectDeform to remove remaining artifacts and get more refined reconstructions [109]. Results show significant improvements in the 3D reconstructions as a result of KinectDeform both qualitatively and quantitatively.

For the "facecap" dataset we use a sequence of 21 frames, simulate a laser scanner in

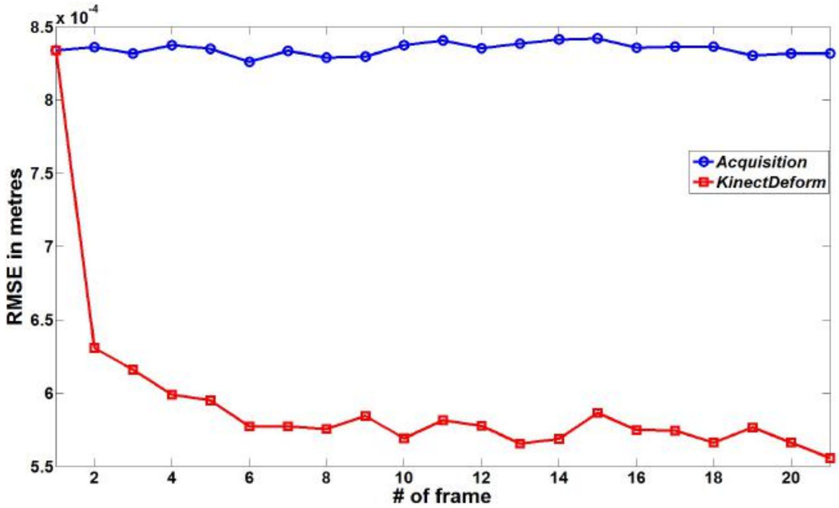
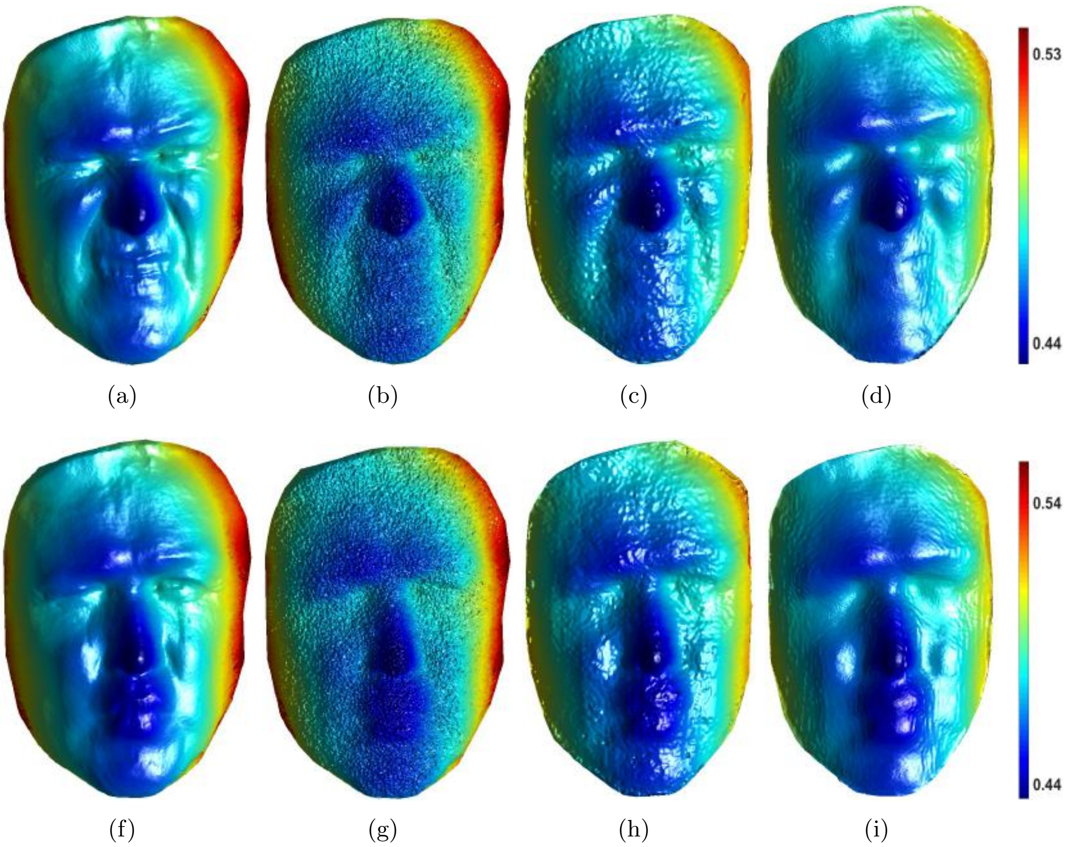


FIGURE 5.6: RMSE of raw and filtered data with ground truth for “facecap” dataset

V-REP with objects placed at $0.5m$ away from the camera [80] and add depth noise to the acquisitions based on Laplacian distribution with 0 mean and standard deviation of $0.25mm$. The standard deviation parameters chosen for the weighting scheme of KinectDeform are $\sigma_c = 0.4mm$ and $0.4mm \geq \sigma_p \leq 0.425mm$. The results are shown in Figure 5.6 and Figure 5.7. Though similar improvements in 3D reconstructions can be seen in this case as well, an important factor apparent here is the effect of temporal filtering due to which the error decreases gradually as shown in the Figure 5.6.

To explain this difference in the temporal effect of filtering between two sequences, a closer look at the deformations introduced in both sequences is required. Figure 5.5(a) and Figure 5.5(f) show a large amount of deformation between frames of the “cloth” sequence. Large deformations break the temporal effect of filtering because of factors such as self occlusions and by significantly changing geometry of the incoming reference frame thus reducing the value of important details brought by the result of previous iterations. That is why when the rate of deformation is small as in the sequence of “facecap” dataset as shown in Figure 5.7(a) and Figure 5.7(f) the effect of temporal filtering is clearly visible as shown in Figure 5.6.

We also tested KinectDeform on real data captured by the Asus Xtion Pro Live camera using a plain cloth being waved in front of it. In this case we tested the empirical weighting scheme similar to KinectFusion in which the weight of reference is increased by 1 after every iteration until a threshold is reached. KinectDeform was run over 25 frames from this dataset and results for frames 10, 15 and 20 are shown in Figure 5.8. It shows that even using this empirical weighting scheme, results in smoother surfaces having preserved details.



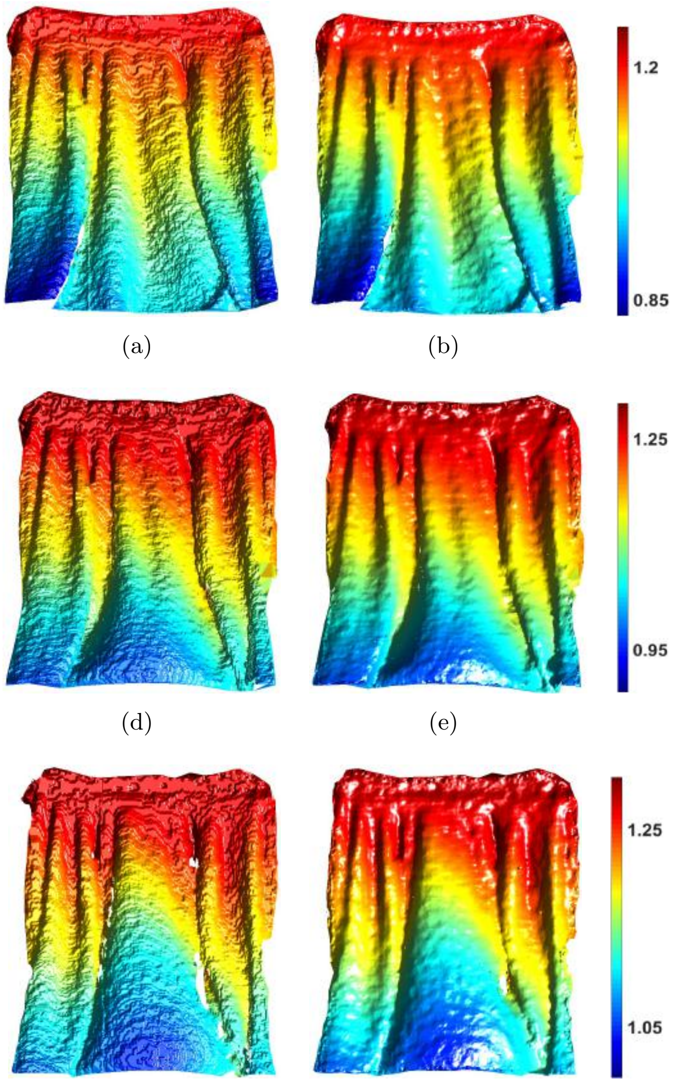
(a) (b) (c) (d)

(f) (g) (h) (i)

FIGURE 5.7: “Facecap” dataset. **Top row:** Frame 5 (a) Ground truth, (b) raw data, (c) result of KinectDeform, (d) result of KinectDeform after deblurring. **Bottom row:** Frame 15 (e) Ground truth, (f) raw data, (g) result of KinectDeform, (h) result of KinectDeform after deblurring. Display color-scale is based on the depth values of the 3D points and is in the units of meters.

5.4 Conclusion

We have presented KinectDeform, a novel method for enhanced 3D reconstruction based on tracking of dynamic non-rigid objects. It has two main components, first is the use of an efficient and effective pair-wise non-rigid tracking which allows for tracking of non-rigid objects without any constraints and without using a template. Second is the use of a recursive filtering mechanism derived from KinectFusion but with a change in the reference being used and a weighting scheme which takes into account different sources of noise present in the input data. We have carried out both quantitative and qualitative evaluation of our method and we show that this algorithm is successfully able to filter noisy depth data to give smoother and feature preserving reconstructions over time. KinectDeform has been designed keeping in mind its planned extension to a completely automated real-time system which should enable us to analyze its performance over longer sequences constituting hundreds of data frames. It should also enable us to study further the domain of filtering based on non-rigid tracking for data acquired from



(a) (b)

(d) (e)

(g) (h)

FIGURE 5.8: Real moving cloth dataset. **Left Column:** Raw acquisitions. **Right Column:** Results of KinectDeform. **Top row:** Frame 10. **Middle row:** Frame 15. **Bottom row:** Frame 20. Display color-scale is based on the depth values of the 3D points and is in the units of meters.

consumer depth cameras both in mono-view and multi-view systems which is discussed in the next chapters.

Chapter 6

View-Independent Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects

In this Chapter, we target enhanced 3D reconstruction of non-rigid objects undergoing local deformations based on a view-independent surface representation with an automated recursive filtering scheme. This work improves upon the KinectDeform algorithm which was presented in Chapter 5. KinectDeform uses an implicit view-dependent volumetric TSDF based surface representation. The view-dependence makes its pipeline complex by requiring surface prediction and extraction steps based on camera's FOV. In this Chapter we propose to use an explicit projection-based Moving Least Squares (MLS) surface representation from point-sets. Moreover, the empirical weighted filtering scheme in KinectDeform is replaced by an automated fusion scheme based on a Kalman filter. Qualitative and quantitative performance analysis shows that the proposed technique is able to produce enhanced and feature preserving 3D reconstructions.

6.1 Introduction

Data acquired by commodity 3D sensing technologies is noisy and of limited resolution. This limits its direct use in various applications ranging from environment mapping for mobile autonomous systems and preservation of historical sites, to human activity and gesture recognition for virtual communications, assistive robotics, security and surveillance.

Research has been carried out to build online template-free and recursive filtering techniques, such as KinectFusion, around commodity 3D sensing technologies to accurately reconstruct captured 3D rigid objects or static scenes [1, 2, 47, 48]. Recently, researchers have focused on tracking highly non-rigid behaviors of deforming objects without the knowledge of any prior shape or reference [97, 110], for the purposes of, for example, depth video enhancement [111–113].

In our previous work in Chapter 5, known as KinectDeform, we showed that a non-rigid registration method can be used in a recursive pipeline similar to KinectFusion to produce enhanced 3D reconstructions of deforming objects [114]. The non-rigid registration step in the pipeline is followed by surface filtering or fusion using volumetric TSDF based implicit surface representation. This surface representation scheme is view-dependent and requires organized point clouds as input. Since non-rigid registration deforms, and hence destroys the organization of input point clouds, an expensive data-reorganization step in the form of meshing and ray-casting is required before surface fusion [114]. Moreover, for fusion, a weighted average scheme is used for which parameters are chosen empirically for each iteration. Ray-casting is used again to extract the resulting point-based surface from fused TSDF volumes after every iteration.

In this work, we propose a method called View-Independent KinectDeform or VI-KinectDeform which improves upon the KinectDeform algorithm by replacing the volumetric TSDF based view-dependent surface representation with an octree-based view-independent and explicit surface representation using Point Set Surfaces (PSS) based on the method of Moving Least Squares (MLS) [115]. This results in a simplified version of KinectDeform with the removal of an expensive data reorganization step. Moreover, we improve upon the fusion mechanism by proposing an automated recursive filtering scheme using a simple Kalman filter [60]. Due to our explicit surface representation, surface prediction step at the end of each iteration is also not required resulting in a simpler algorithm. We compare the results of VI-KinectDeform with those of KinectDeform using non-rigidly deforming objects and show that for the same number of iterations VI-KinectDeform produces stable and more accurate 3D reconstructions.

6.2 Background and Problem Formulation

6.2.1 Background

The online and template-free recursive filtering problem for getting enhanced 3D reconstructions via commodity depth cameras has been formulated in Section 2.4.1. In this work we redefine the problem by assuming that an input 3D point cloud acquired with a camera at time t may be unorganized and hence, can be represented by a 3D point-set $\mathcal{V}_t$, of size M, with corresponding measurement error $\mathcal{E}_t$. The point-set $\{\mathbf{p}_t^j\}$ in $\mathcal{V}_t$, where $\mathbf{p}_t^j \in \mathbb{R}^3$ and $j \in \{1, \ldots, M\}$, approximates the underlying surface of deformable objects in camera's field of view. The problem at hand is therefore to reduce $\mathcal{E}_t$ for $t > 0$, to recover an enhanced sequence $\{\mathcal{V}_0^{f'}, \mathcal{V}_1^{f'}, \ldots, \mathcal{V}_{N-1}^{f'}\}$ starting from the input sequence $\{\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_{N-1}\}$. This leads to redefining the required recursive filtering function $filt(\cdot, \cdot)$ in (2.10) such that:

$$\mathcal{V}_t^{f'} = \begin{cases} \mathcal{V}_t & \text{for } t = 0, \\ filt(\mathcal{V}_{t-1}^{f'}, \mathcal{V}_t) & t > 0. \end{cases} \tag{6.1}$$

As mentioned before a major shortcoming of the KinectDeform scheme lies in the 3D surface representation based on the view-dependent TSDF volume for data fusion and filtering [114]. Construction of a TSDF volume for a point cloud requires projecting each centroid of the TSDF volume to the corresponding camera's image plane which, in turn, requires the points in the point cloud to be organized with respect to the image plane. Therefore, after the non-rigid registration which destroys the data organization of the input point cloud, an expensive data reorganization step based on meshing and ray-casting is required for computation of a TSDF. After that, the TSDF volumes, created for current measurement and the deformed result of previous iteration, are fused together using an empirical weighting scheme whereby the weighting parameters are chosen heuristically [114]. This is followed by another surface prediction step via ray-casting to extract the final filtered surface from the fused volume.

6.2.2 Point Set Surfaces

Keeping in view the key limitations of the KinectDeform method explained in Section 6.2.1, a simpler approach would be to replace the view-dependent TSDF volume-based surface representation for fusion and filtering with a view-independent surface representation. This would result in avoiding data reorganization and surface prediction steps. As mentioned before the input points $\{\mathbf{p}^j\}$, ignoring subscript t for simplicity, approximate the underlying surface of objects in the scene. In [115], Alexa et al. built upon Levin's work [116], and proposed a view-independent point-based surface reconstruction method based on MLS. This method projects a point $\mathbf{r} \in \mathbb{R}^3$ lying near $\{\mathbf{p}^j\}$ on the underlying surface approximated by the local neighborhood of $\mathbf{r}$. Apart from facilitating the computation of the differential geometric properties of the surface such as normals and curvatures, this method is able to handle noisy data and provides smooth reconstructions. Moreover, the local nature of projection procedure improves the efficiency of the algorithm [117].

The projection procedure as proposed by Alexa et al. is divided into two steps [115]. In the first step a local reference domain, i.e., a plane $\mathcal{H}_{\mathbf{r}} = \{\mathbf{p} \in \mathbb{R}^3 : \vec{\mathbf{u}}^T\mathbf{p} = \vec{\mathbf{u}}^T\mathbf{v}, \mathbf{v} \in \mathbb{R}^3, \vec{\mathbf{u}} \in \mathbb{R}^3, \|\vec{\mathbf{u}}\| = 1\}$, is computed by minimizing the following non-linear energy function [117]:

$$e_{MLS}(\mathbf{v}, \vec{\mathbf{u}}) = \sum_{\mathbf{q_r} \in \Omega_{\mathbf{r}}} w(d, \|\mathbf{q_r} - \mathbf{v}\|)\langle \vec{\mathbf{u}}, \mathbf{q_r} - \mathbf{v}\rangle^2, \tag{6.2}$$

where $\Omega_{\mathbf{r}}$ is the neighborhood of $\mathbf{r}$. Also $\vec{\mathbf{u}} = (\mathbf{r} - \mathbf{v})/\|\mathbf{r} - \mathbf{v}\|$, $\langle ., . \rangle$ is the dot product and $w(d, e) = \exp(-e^2 d^{-2})$ is the Gaussian weight function where d represents the anticipated spacing between neighboring points [115]. The surface features of size less than d are smoothed out due to the MLS projection. Replacing $\mathbf{v}$ by $\mathbf{r} + t\vec{\mathbf{u}}$ where $t \in \mathbb{R}$ in (6.2) we have:

$$e_{MLS}(\mathbf{r}, \vec{\mathbf{u}}) = \sum_{\mathbf{q_r} \in \Omega_{\mathbf{r}}} w(d, \|\mathbf{q_r} - \mathbf{r} - t\vec{\mathbf{u}}\|)\langle \vec{\mathbf{u}}, \mathbf{q_r} - \mathbf{r} - t\vec{\mathbf{u}}\rangle^2. \tag{6.3}$$

The minimum of (6.3) is found with the smallest t and the local tangent plane $\mathcal{H}_{\mathbf{r}}$ near $\mathbf{r}$ [115]. The local reference domain is then defined by an orthonormal coordinate system in $\mathcal{H}_{\mathbf{r}}$ with $\mathbf{v}$ as its origin [117].

In the next step, we find the orthogonal projections of points $\mathbf{q_v} \in \Omega_{\mathbf{v}}$, lying in the local neighborhood of $\mathbf{v}$ to get their corresponding 2D representations $(x_{\mathbf{q_v}}, y_{\mathbf{q_v}})$ in the local coordinate system in $\mathcal{H}_{\mathbf{r}}$. The height of $\mathbf{q_v}$ over $\mathcal{H}_{\mathbf{r}}$ is found via:

$$h_{\mathbf{q_v}} = \langle \vec{\mathbf{u}}, \mathbf{q_v} - \mathbf{r} - t\vec{\mathbf{u}}\rangle. \tag{6.4}$$

Using the local 2D projections and the height map, a local bivariate polynomial approximation $g : \mathbb{R}^2 \to \mathbb{R}$ is computed by minimizing the weighted least squares error:

$$\sum_{\mathbf{q_v} \in \Omega_{\mathbf{v}}} w(d, \|\mathbf{q_v} - \mathbf{r} - t\vec{\mathbf{u}}\|)(g(x_{\mathbf{q_v}}, y_{\mathbf{q_v}}) - h_{\mathbf{q_v}})^2. \tag{6.5}$$

The degree of the polynomial to be computed is fixed beforehand. At the end, the projection $\mathbf{r} \in \mathbb{R}^3$ onto the underlying surface, denoted by $P(\mathbf{r})$, is defined by the polynomial value at the origin, i.e.:

$$P(\mathbf{r}) = \mathbf{v} + g(0, 0)\vec{\mathbf{u}} = \mathbf{r} + (t + g(0, 0))\vec{\mathbf{u}}. \tag{6.6}$$

The projected point is considered to be the resulting filtered point lying on the approximated surface. These two steps are repeated for all points which need to be sampled to sufficiently represent the surfaces of objects in camera's FOV to get enhanced 3D reconstructions.

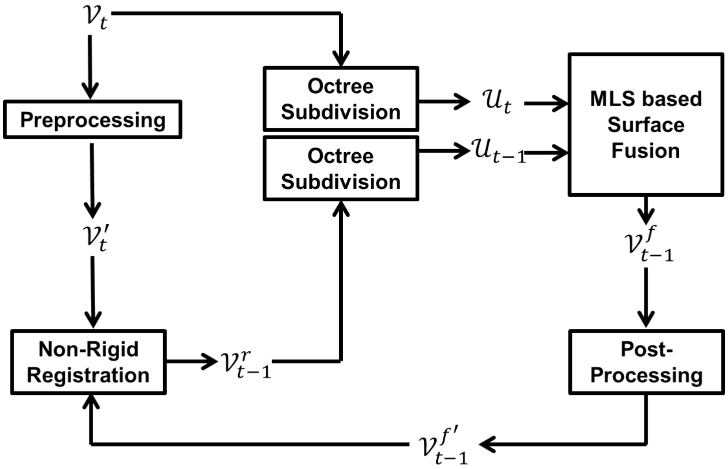


FIGURE 6.1: Detailed pipeline of VI-KinectDeform. $\mathcal{V}_t$: input point cloud at time t. $\mathcal{V}_t^{'}$: result of pre-processing on $\mathcal{V}_t$. $\mathcal{V}_t^r$: result of non-rigid registration of $\mathcal{V}_{t-1}^{f'}$ to $\mathcal{V}_i^{'}$. $\mathcal{U}_t$ and $\mathcal{U}_{t-1}$: resulting voxel sets based on octree sub-division corresponding to $\mathcal{V}_t$ and $\mathcal{V}_{t-1}^r$ respectively. $\mathcal{V}_{t-1}^f$: the result of projection-based MLS surface computation and Kalman filtering-based fusion. $\mathcal{V}_{t-1}^{f'}$: the final result after post-processing. For more details please read Section 6.2 and Section 6.3.

6.3 Proposed Technique

Figure 6.1 shows the pipeline of VI-KinectDeform which is an improved/simplified version of KinectDeform. After the non-rigid registration step which deforms $\mathcal{V}_{t-1}^{f'}$ to produce $\mathcal{V}_{t-1}^r$ to be registered to $\mathcal{V}_t$, the data reorganization step is removed. Instead, a view-independent surface representation and filtering based on the MLS method is proposed. Since the MLS method works on the local neighborhoods of sampled points, voxelizing/sub-dividing the space of input 3D point clouds not only provides us with sampling information but also helps in accelerating the search for local neighborhoods of the sampled points. After that, the sampled points are projected onto the underlying surfaces of both point clouds based on the MLS method. The resulting projections are then fused together via an automatic Kalman filtering based scheme to give enhanced 3D reconstructions. These steps are explained as follows:

6.3.1 Sampling and MLS Based Projection

We use octree data structure to sample the space occupied by $\mathcal{V}_t$ and $\mathcal{V}_{t-1}^r$ resulting in two voxel sets $\mathcal{U}_t$ and $\mathcal{U}_{t-1}$ with a pre-defined depth $k \in \mathbb{N}$. At depth level k, $\mathcal{U}_t$ and $\mathcal{U}_{t-1}$ contain the non-empty voxels o_t^k and o_{t-1}^k, respectively. It is to be noted that since $\mathcal{V}_t$ and $\mathcal{V}_{t-1}^r$ are mapped, the corresponding voxels in $\mathcal{U}_t$ and $\mathcal{U}_{t-1}$ occupy the same space. Each voxel $u_{t,a}^k \in \mathcal{U}_t$ where $a \in \{1, \ldots, o_t^k\}$ (or similarly each voxel $u_{t-1,b}^k \in \mathcal{U}_{t-1}$) is represented by its geometric center $\mathbf{c}_{t,a}^k$ (or $\mathbf{c}_{t-1,b}^k$), the points contained in the voxel and information about its immediate neighbors. These centroids lying near input points provide us with suitable sampling points to be projected onto the underlying surface based on the procedure explained in Section 6.2. Therefore, in the next step the centroid of each non-empty leaf voxel in $\mathcal{U}_t \cup \mathcal{U}_{t-1}$ lying in the vicinity of points from both $\mathcal{V}_t$ and $\mathcal{V}^r_{t-1}$ is projected on the approximated underlying surfaces using its corresponding neighborhood points in $\mathcal{V}_t$ and $\mathcal{V}^r_{t-1}$, respectively, via the MLS method to get:

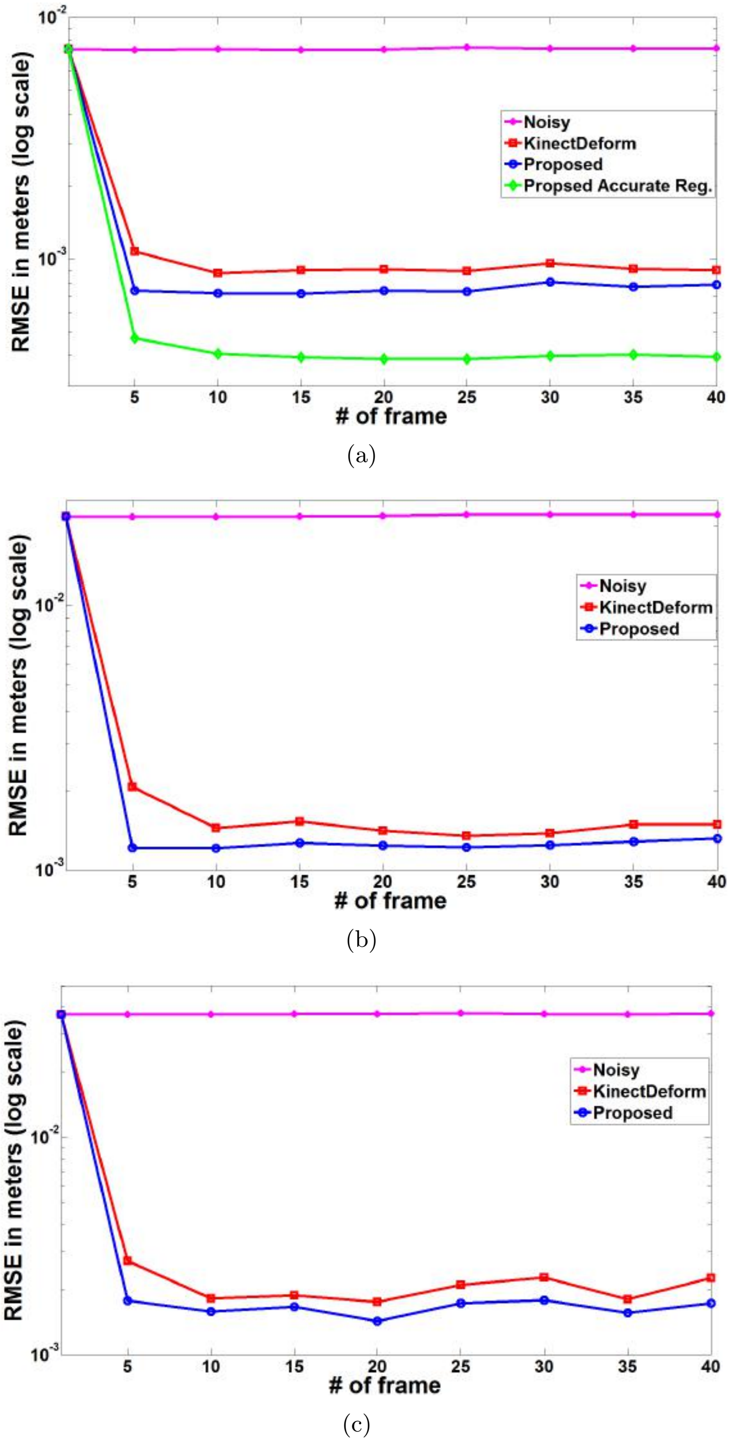
$$\begin{aligned} \mathbf{p}^c_t = P_t(\mathbf{c}^k_{t,a}), \mathbf{p}^c_{t-1} = P_{t-1}(\mathbf{c}^k_{t,a}),\ \mathit{or} \\ \mathbf{p}^c_t = P_t(\mathbf{c}^k_{t-1,b}), \mathbf{p}^c_{t-1} = P_{t-1}(\mathbf{c}^k_{t-1,b}). \end{aligned} \tag{6.7}$$

where $1 \leq c \leq (o^k_t + o^k_{t-1})$, and $P_t(.)$ and $P_{t-1}(.)$ are the MLS based projections function, defined in (6.6), corresponding to $\mathcal{V}_t$ and $\mathcal{V}^r_{t-1}$, respectively. The degree of the bivariate polynomial approximating the underlying surface computed for each centroid is kept variable (maximum 3 for our experiments) depending on the number of points found in the neighborhood. Hence as a result of the MLS-based projection procedure, two sets of corresponding filtered points, $\{\mathbf{p}^c_t\}$ and $\{\mathbf{p}^c_{t-1}\}$, are generated.

6.3.2 Fusion

It is clear that under ideal conditions, i.e., noise free sensor and with perfectly registered inputs $\mathcal{V}_t$ and $\mathcal{V}^r_{t-1}$, the point sets $\{\mathbf{p}^c_t\}$ and $\{\mathbf{p}^c_{t-1}\}$ should be the same however, the noise factors affecting the sensor measurements and the non-rigid data registration have to be taken into account. Therefore, in this step we propose a methodology to fuse the corresponding projected points $\{\mathbf{p}^c_t\}$ and $\{\mathbf{p}^c_{t-1}\}$, taking into account noise factors affecting them to produce a filtered 3D reconstruction $\mathcal{V}^f_t$. The main noise factor affecting the current measurement $\mathcal{V}_t$, and hence $\{\mathbf{p}^c_t\}$, is the sensor noise while on the other hand for $\mathcal{V}^r_{t-1}$ it is assumed that, due to pre-processing, some amount of this sensor noise is mitigated with a loss of few details and hence the main noise factor is error due to non-rigid registration [114]. This should be coupled with iterative effects of filtering as $\mathcal{V}^r_{t-1}$ is indeed a deformed state of the filtered $\mathcal{V}^{f'}_{t-1}$.

In KinectDeform, we tackle these factors by performing a surface fusion/filtering using a weighted average of TSDF values of corresponding voxels [114]. The weights are chosen empirically based on an analysis of noise factors affecting the two input voxel sets per iteration. In this work, we propose an automatic filtering approach by point tracking with a Kalman filter [60]. The observation model is based on the current measurements $\{\mathbf{p}^c_t\}$, and the associated sensor noise ϵ^s_t is assumed to follow a Gaussian distribution $n^s_t \sim \mathcal{N}(0, \sigma^2_{s,t})$. Similarly, the motion model assumes as its output the result of non-rigid registration ,i.e., $\{\mathbf{p}^c_{t-1}\}$, and the associated process noise n^r_{t-1} is assumed to follow



(a)

(b)

(c)

Figure 6.2: "Facecap" dataset. Quantitative analysis on data with different levels of Gaussian noise. Each figure contains RMSE in log scale of: noisy data, result of KinectDeform and result of VI-KinectDeform. (a) Results for Gaussian noise with standard deviation of 0.01 m. It also contains RMSE in log scale of VI-KinectDeform with registration based on noise free data. (b) Results for Gaussian noise with standard deviation of 0.03 m. (c) Results for Gaussian noise with standard deviation of 0.05 m.

a Gaussian distribution $n^r_{t-1} \sim \mathcal{N}(0, \sigma^2_{r,t-1})$. Therefore the prediction step is:

$$\begin{cases} \mathbf{p}^k_{t|t-1} = \mathbf{p}^k_{t-1}, \\ \sigma^2_{t|t-1} = \sigma^2_{t-1|t-1} + \sigma^2_{r,t-1}, \end{cases} \tag{6.8}$$

and measurement update is given as:

$$\begin{cases} \mathbf{p}^k_{t|t} = \mathbf{p}^k_{t|t-1} + G_t(\mathbf{p}^k_t - \mathbf{p}^k_{t|t-1}), \\ \sigma^2_{t|t} = \sigma^2_{t|t-1} - G_t \sigma^2_{t|t-1}, \end{cases} \tag{6.9}$$

where:

$$G_t = \frac{\sigma^2_{t|t-1}}{\sigma^2_{t|t-1} + \sigma^2_{s,t}}. \tag{6.10}$$

This results in the filtered set of points $\{\mathbf{p}^k_{t|t}\}$ which constitutes $\mathcal{V}^f_t$.

6.4 Experiments and Results

The quality of VI-KinectDeform is analyzed both quantitatively and qualitatively. We use the "Facecap" dataset which captures a person's face deforming non-rigidly due to changing expressions in different scenes [107]. The selected scene includes 40 frames. We simulate a depth camera in V-Rep [80], placed approximately at 0.45 m − 0.55 m away from the object and add Gaussian noise in depth measurements with zero mean and standard deviations of 0.01 m, 0.03 m and 0.05 m, respectively. Experiments are carried out using these datasets for both VI-KinectDeform and KinectDeform. A bilateral filter is used in the pre-processing step to obtain improved registration for both methods [118]. We use the algorithm proposed by Destelle et al. [97] for non-rigid registration in both methods. We use the proposed automated fusion scheme in both VI-KinectDeform and KinectDeform by replacing the empirical fusion scheme used previously. Post-processing is based on the bilateral mesh de-noising with very small parameters for the neighborhood size and the projection distance for both VI-KinectDeform and KinectDeform [119]. The quantitative evaluation of VI-KinectDeform as compared to KinectDeform is reported in Figure 6.2. It shows the RMSE of the data enhanced with VI-KinectDeform, and the data enhanced with KinectDeform with respect to the ground truth data for different noise levels. These results show superior performance of VI-KinectDeform in terms of overall accuracy of 3D reconstructions as compared to KinectDeform. It is noted that the accuracy of the proposed technique is restricted by the accuracy of the considered non-rigid registration algorithm. We have tested the proposed VI-KinectDeform by using non-rigid registration parameters obtained from noise free data. The post-processing

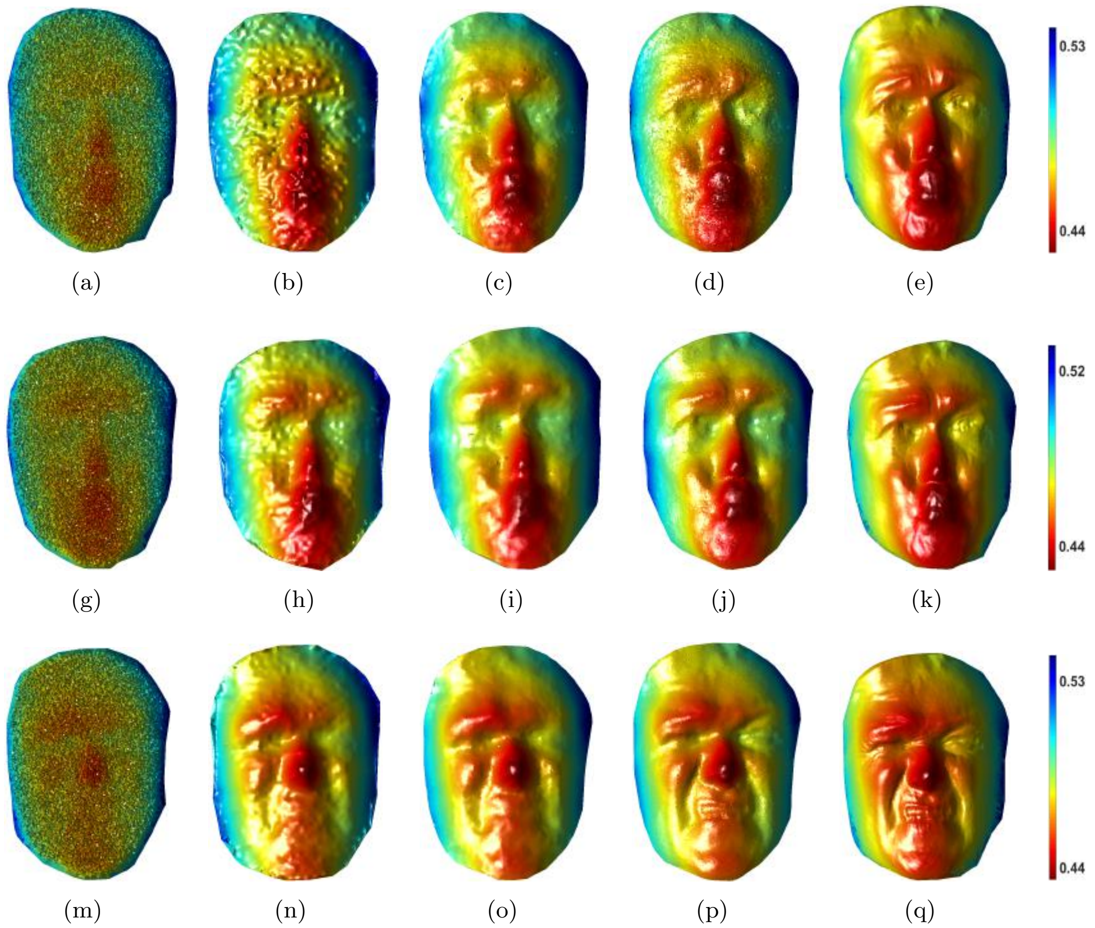


FIGURE 6.3: “Facecap” dataset. **First row:** Frame #5, **Second row:** Frame #15, **Third row:** Frame #35. Each row contains noisy data with Gaussian noise of standard deviation 0.01 m, result of KinectDeform, result of VI-KinectDeform, result of VI-KinectDeform with registration based on noise free data and ground truth respectively. Display color-scale is based on the depth values of the 3D points and is in the units of meters.

step is skipped in this case. The resulting curve in Fig. 6.2(a) shows a significant decrease in error when using VI-KinectDeform as compared to its earlier version. This is observed through all frames. The qualitative analysis presented in Figure. 6.3, corresponding to the noise level and results in Figure 6.2(a), shows superior quality of 3D reconstructions obtained via VI-KinectDeform in terms of feature preservation and smoothness when compared to the results obtained via KinectDeform.

For further analysis of the performance of the proposed technique, we use the “Swing” dataset [120]. We, again, simulate a depth camera in V-Rep placed approximately at 1.5 m away from the object and add *Gaussian* noise with zero mean and standard deviation of 0.0075 m. We use 20 frames for this experiment. We analyze the performance of the proposed VI-KinectDeform with 3 other view-independent surface representation

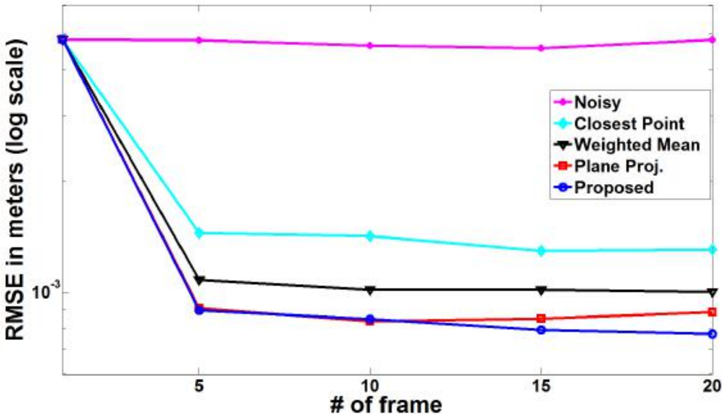
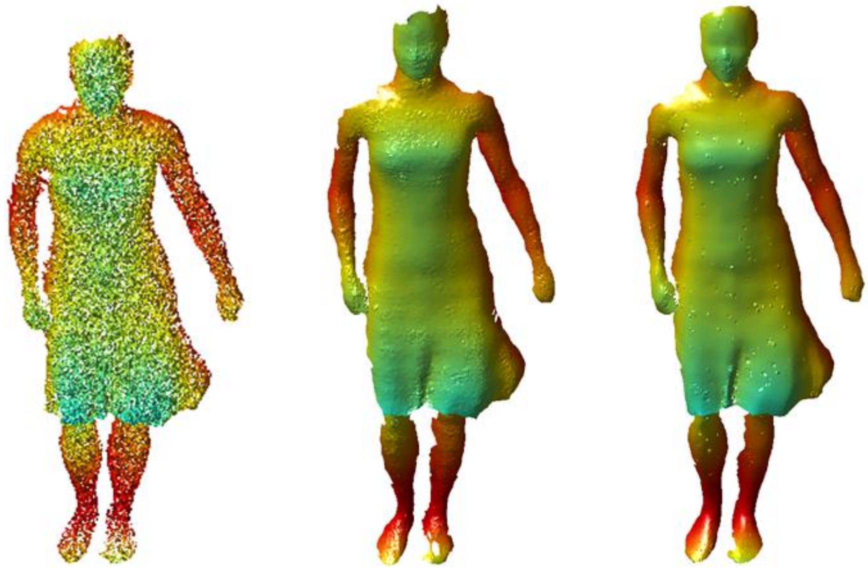


Figure 6.4: "Swing" dataset. RMSE in log scale of: noisy data with *Gaussian* noise of standard deviation 0.0075 m, result of closest point-based surface representation, result of weighted-mean based surface representation, result of local plane projection-based surface representation and result of the proposed projection-based MLS surface representation. Please read Section 3.4 for more details.

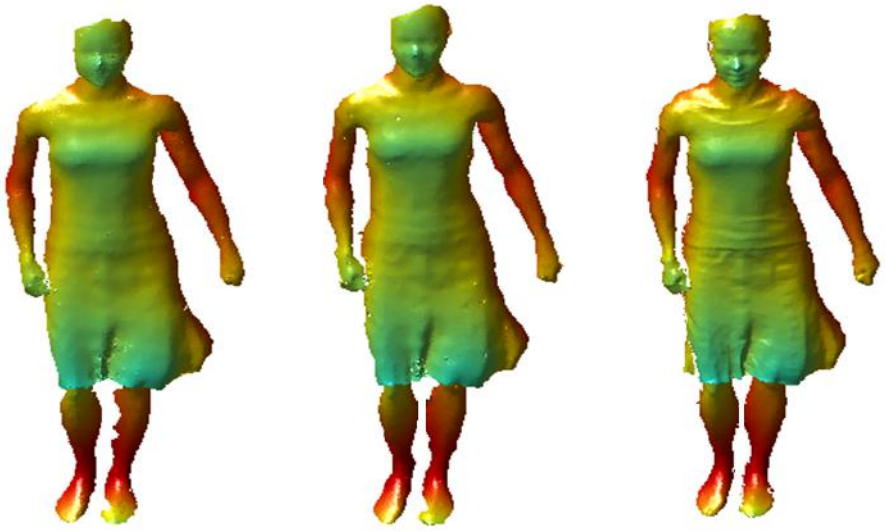
schemes. These representation schemes are based on finding the surface approximation with respect to each centroid belonging to the leaf nodes of $\mathcal{U}_t$ and $\mathcal{U}_{t-1}$ lying close to $\mathcal{V}_t$ and $\mathcal{V}^r_{t-1}$.

The first scheme is based on finding the closest points in local neighborhoods of the centroids. The second scheme is based on finding the weighted mean of all points lying in local neighborhoods of each centroid using the weighting scheme similar to the one used in (6.2). The third scheme fits tangent planes to points in local neighborhoods and finds the projections of the centroid on them. It is similar to the proposed scheme wherein the degree of the polynomial is fixed to one.

Quantitative and qualitative results are shown in Figure 6.4 and Figure 6.5, respectively. As expected, Figure 6.4 shows that the closest point-based method is least accurate followed by the weighted mean-based method, the plane projection-based method, and the proposed projection-based MLS method in terms of overall accuracy. Similar results are obtained via quantitative analysis as shown in Figure 6.5 wherein the proposed method produces the most accurate and feature preserving reconstruction. The plane projection-based method also gives good results but small features such as nose and foldings on clothing are not well preserved. This experiment shows that the proposed pipeline is generic enough such that any view-independent point-based surface representation scheme using local neighborhoods can replace the proposed MLS-based scheme.



(a) Noisy (b) Closest Point (c) Weighted Mean



(d) Plane Proj. (e) Proposed Tech. (f) Ground Truth

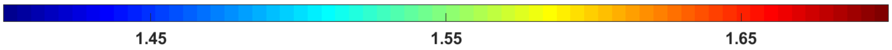


FIGURE 6.5: "Swing" dataset. **First row:** *Left:* noisy data with Gaussian noise of standard deviation 0.0075 m, *Center:* result of closest point-based surface representation, *Right:* result of weighted mean-based surface representation. **Second row:** *Left:* result of local plane projection-based surface representation, *Center:* result of the proposed projection-based MLS surface representation, *Right:* ground truth. Display color-scale is based on the depth values of the 3D points and is in the units of meters.

6.5 Conclusion and Future Work

In this work we have proposed VI-KinectDeform, an automated recursive filtering scheme for producing enhanced 3D reconstructions of non-rigidly deforming objects. It improves upon our previous work, i.e., KinectDeform [114], by replacing the implicit view-dependent TSDF based surface representation scheme with an explicit MLS-based view-independent surface representation scheme [115]. This simplifies the pipeline by removing surface prediction and extraction steps. Moreover we improve upon the data fusion scheme by proposing an automated point tracking with a Kalman filter [60], The quantitative and qualitative evaluation of our method shows that it is able to produce smooth and feature preserving 3D reconstructions with an improved accuracy when compared to KinectDeform. We also show that the proposed pipeline is generic, and can use any view-independent point-based surface representation scheme. The generic and view-independent nature of this algorithm allows for the extension to a multi-view system to produce enhanced full 360° 3D reconstructions of scenes containing non-rigid objects.

Chapter 7

Full 3D Reconstruction of Non-Rigidly Deforming Objects

In this chapter, we discuss enhanced full 360° 3D reconstruction of dynamic scenes containing non-rigidly deforming objects using data acquired from commodity RGB-D or 3D cameras. In the Chapters 3 and 4, we have explored the domain of setting up a multi-view system around commodity RGB-D cameras. Our proposed method accurately aligns the partial measurements, acquired by each camera, to obtain full 360° 3D reconstructions of dynamic scenes instantaneously. Moreover, we have extended state-of-art by proposing template-free recursive data filtering methods to remove noise and produce enhanced 3D reconstructions of non-rigildy deforming objects using data acquired from mainly mono-view systems. In this part we target to enhance the quality of noisy and low-resolution (LR) full 3D reconstructions acquired with a fully calibrated multi-view system. For this purpose, we propose a recursive dynamic multi-frame 3D super-resolution (SR) scheme for noise removal and resolution enhancement of 3D measurements, of non-rigidly deforming objects, acquired by 3D sensors in a multi-view system. The proposed approach is template-free and works directly on 3D points, thus giving it flexibility to the types of objects being reconstructed, and the ability to capture their characteristics, i.e., position and motion in the 3D world more accurately. To tackle the affects of system blur we use an efficient multi-level 3D bilateral total variation (BTV) regularization. Quantitative and qualitative performance evaluation of the proposed technique using both simulated and real data shows that it outperforms state-of-art methods and produces smooth, high-quality and feature preserving full 3D reconstructions.

7.1 Introduction

Acquiring high quality and full 360° 3D reconstructions of dynamic scenes containing non-rigidly deforming objects is one of the fundamental goals of research in computer vision and robotics.

Compared to photometric cameras, commodity 3D cameras based reconstruction approaches, although aided by 3D acquisitions, have to overcome problems related to noise and limited resolution. After the advent of commodity RGB-D or 3D cameras based enhanced 3D reconstruction techniques for rigid objects [1, 2, 47, 48], researchers have moved towards handling non-rigid deformations by proposing to construct complete and enhanced 3D models of mainly human subjects by fusing information from multiple views. This requires handling quasi-rigid motions between different views for which a global non-rigid registration is performed [49, 50], or a model-to-part registration based on deformation graph [121] or Shape Completion and Animation of People (SCAPE) model [122] is used to avoid error accumulation [51, 52]. The works of Cui et al. [123] and Shapiro et al. [124] are interesting in this regard as they try to tackle the limited-resolution of the data acquired commodity 3D cameras as well. Before data fusion, a resolution enhancement step, called super-resolution (SR), is performed on data from individual views with the help of either high-resolution (HR) RGB images [123] or mono-view filtering under rigidity constraints [1, 124], to get enhanced HR 3D reconstructions.

To efficiently achieve enhanced 3D reconstructions of non-rigid objects, undergoing relatively large local motions, template based methods have been proposed in which a high quality template is built as a first step. Li et al. [50, 61] and Zollhöfer et al. [29] propose to pre-build high quality complete templates of the target objects, which are then used to track non-rigid deformations before being fused with current measurements to produce enhanced 3D reconstructions. These methods are restricted to the class of objects which can stay static or undergo controlled rigid motions for a sufficient period of time for accurate template reconstruction.

On the other hand, methods based on different 3D non-rigid registration algorithms, using compact deformable parameterizations based on, e.g., Deformation Graphs [121, 125], Thin Plate Splines [126, 127], and skeleton extraction [10], consensus and matching under articulated motion assumptions [128], have been proposed [129, 130]. Ye et al., propose a performance capture method for complete human bodies based on skeleton fitting with three hand-held Kinect v1 cameras by making use of RGB information to aid in the registration process [131]. Li et al. [50] employ a visual hull prior, with pair-wise non-rigid scan registration based on deformation graphs [125] for hole-filling and shape completion based on relatively noise-free data.

Another class of template-free methods for complete reconstruction of 3D objects is based on spatio-temporal refinement and tracking of input data to build 4D models offline [53, 54]. Wand et al. use a topology-aware adaptive sub-space deformation technique to reduce the drift, together with as-rigid-as-possible and temporally coherent constraints on motion, to establish correspondences between acquisitions in 3D videos [55, 56]. The computed deformation field is used to construct a noise-free template from partial acquisitions. Sharf et al. relax the motion and spatial coherence constraints by using a bounded volume [57]. Their method suffers from flickering effects while still not being able to capture large deformations [50]. A recent work by Xu et al. is interesting wherein a complete 3D model, and ultimately a 4D reconstruction, is iteratively built by fusing the non-rigidly deforming partial and low resolution observations and parameters of deformation subspace with the help of the Coherent Point Drift (CPD) algorithm [132]. CPD is a probabilistic non-rigid registration algorithm which is shown to handle arbitrary motions and arbitrary topologies accurately. The method of Xu et al. also has a tendency to suffer from drift due to large deformations.

Similar to Xu et al. [58], a recent body of work in this domain uses a recursive approach for temporal fusion and incremental construction of high quality 3D reference models without the need to build complete 4D reconstructions. In this vain, Dou and Fuchs, have proposed a recursive template-free scheme, using a multi-view system composed of ten Kinect v1 cameras, which tracks the motion of dynamic human subjects using deformation graphs [3]. After motion estimation, partial measurements and the reference frame are fused together using a directional distance function to produce enhanced 3D reconstructions [3, 133]. This method is restricted by the limitations of having open gesture topology for the reference frame. Moreover, the results lack quantitative analysis, and the technique has not been tested in setups with fewer cameras or with low-resolution acquisitions. DynamicFusion is a similar work which targets real-time enhancement and incremental surface completion of non-rigidly deforming objects using a mono-view system, but suffers from similar limitations as the work by Dou and Fuchs [3, 62].

To tackle the above mentioned challenges of recursive surface enhancement techniques, we have proposed mono-view techniques such as KinectDeform and VI-KinectDeform in Chapter 5 and Chapter 6. They are able to handle large local motions and do not require a reference model with a fixed topology. KinectDeform is a view-dependent method and hence can only produce partial reconstructions. VI-KinectDeform, on the other hand, is a view-independent moving least squares (MLS) and Kalman filter based, 3D video enhancement scheme which could directly be used in 3D multi-view systems. It has duly been tested for mono-view systems but has not been tested for and may not perform well on LR data [134].

To tackle LR and noisy non-rigidly deforming data we look into image-based SR techniques [111–113, 135–137]. It is important to mention the work of Al Ismaeil et al. in this regard which, though restricted to enhancement of mono-view dynamic depth videos, proposes to tackle the problem of LR sensing systems via recursive dynamic multi-frame depth SR algorithm [111, 135]. This algorithm recursively estimates an HR and enhanced depth map at each time-step, by taking as input the current upsampled LR measurement and the result of previous time-step to track and correct the depth and radial displacement values of each 3D point, associated with a pixel, using a Kalman filter [60]. This method performs well on various non-rigid scenes but cannot be used for full 3D reconstructions. Moreover, due to range flow approximation this method can face difficulties to track fast and abrupt motions.

This overview of the state-of-art suggests that although several approaches for enhanced and complete 3D reconstructions of non-rigid objects, undergoing local motions, have been proposed, they suffer from several limitations. These limitations are due to the requirements for template generation, inability to tackle large deformations, inability to tackle highly noisy and low-resolution data, and inability to produce online results.

To tackle these limitations, we propose a template-free and recursive SR approach capable of handling highly noisy and low-resolution data acquired from a multi-view system. The pipeline of the proposed algorithm is shown in Figure 7.1. Following image-based SR approaches [113, 135], at every time-step, it upsamples the acquired measurement and uses it together with the result of previous time-step to track and correct the position and motion of each 3D point. It, therefore, avoids error accumulation or drift caused by large deformations. Furthermore, regularization of positions and correction of motion is carried out, at each time-step, with the help of a novel 3D BTV regularization. We validate the proposed approach via quantitative and qualitative analysis on simulated and real data.

7.2 Problem Formulation

A fully calibrated 3D multi-view system captures an LR 3D video $\{\mathcal{L}_t\}$ of a scene containing non-rigidly deforming objects, with each unorganized point cloud represented as an ordered point-set $\mathcal{L}_t$, acquired at time t, and containing M 3D points, where $M \in \mathbb{N}^*$. The acquired points in $\mathcal{L}_t$ approximate the underlying surface of objects in the scene. The objective is to reconstruct an enhanced HR 3D video $\{\mathcal{H}_t\}$ where each

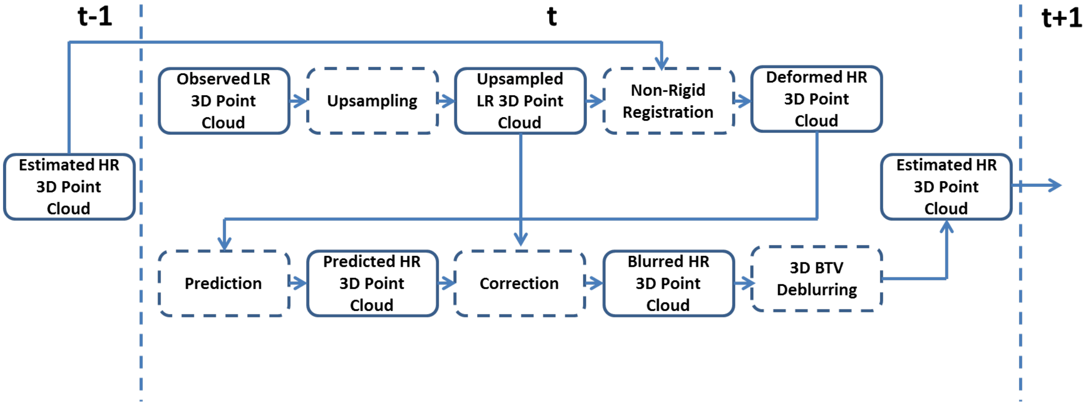

t-1
t
t+1
Estimated HR 3D Point Cloud
Observed LR 3D Point Cloud
Upsampling
Upsampled LR 3D Point Cloud
Non-Rigid Registration
Deformed HR 3D Point Cloud
Estimated HR 3D Point Cloud
Prediction
Predicted HR 3D Point Cloud
Correction
Blurred HR 3D Point Cloud
3D BTV Deblurring

FIGURE 7.1: Detailed pipeline of the proposed recursive dynamic multi-frame 3D super-resolution algorithm. For more details please see Section 7.3.

point-set $\mathcal{H}_t = [\mathbf{p}_t^1, \cdots, \mathbf{p}_t^M]$, with points $\mathbf{p}_t^i = (x_t^i, y_t^i, z_t^i)^\intercal$ where x_t^i, y_t^i and $z_t^i \in \mathbb{R}$, $\intercal$ is the transpose, $i \in \{1, \cdots, U\}$, and also, $U = o \times M$, where $o \in \mathbb{N}^*$ is factor by which the resolution of the input data is enhanced. It is also known as the SR factor.

Let us assume that each LR acquired point cloud $\mathcal{L}_t$ is related to the corresponding HR cloud $\mathcal{H}_t$ via the sensor model:

$$\mathcal{L}_t = r(\mathcal{H}_t) + \mathcal{W}_t, \tag{7.1}$$

where $r(.)$ is the measurement function which incorporates system blur and downsampling operators, and $\mathcal{W}_t$ represents additive white noise at time t and has same size as $\mathcal{H}_t$. We can perform dense upsampling on the acquired LR point clouds as a pre-processing step which eliminates the resolution difference between the measured data and the desired $\hat{\mathcal{H}}_t$ that we are to estimate, and helps in decreasing the registration error [112, 113]. Considering a dense upsampling operator $\uparrow$ which performs an increase or enhancement in resolution, with a factor o, (7.1) becomes:

$$\tilde{\mathcal{H}}_t = \mathcal{L}_t \uparrow = [r(\mathcal{H}_t)] \uparrow + \mathcal{W}_t \uparrow, \tag{7.2}$$

Moreover, each HR point cloud $\mathcal{H}_{t-1}$ undergoes a dynamic deformation at time t to give HR point cloud $\mathcal{H}_t$ via:

$$\mathcal{H}_t = h_t\left(\mathcal{H}_{t-1}\right) + \mathcal{F}_t, \tag{7.3}$$

where $h_t(\cdot)$ is the local deformation function which deforms $\mathcal{H}_{t-1}$ to $\mathcal{H}_t$, and $\mathcal{F}_t$ is the innovation containing information about new and disappearing points [111, 135].

The objective of this paper is to devise a dynamic multi-frame SR algorithm which recursively estimates $\mathcal{H}_t$, by taking into account the current upsampled input point cloud $\tilde{\mathcal{H}}_t$, the previous result $\hat{\mathcal{H}}_{t-1}$ and the estimated 3D non-rigid deformation relating them, such that:

$$\hat{\mathcal{H}}_t = \begin{cases} \tilde{\mathcal{H}}_t & \text{for } t = 0, \\ filt(\hat{\mathcal{H}}_{t-1}, \tilde{\mathcal{H}}_t) & t > 0. \end{cases} \tag{7.4}$$

where $filt(\cdot, \cdot)$ is a filtering function which is redefined from (2.10). It takes into account the local deformations between $\hat{\mathcal{H}}_{t-1}$ and $\tilde{\mathcal{H}}_t$ to mitigate the effects of cameras' measurement limitations which result in noisy measurements with limited resolution and system blur.

7.3 Proposed Approach

7.3.1 Overview

In this chapter, we propose a recursive dynamic multi-frame 3D SR algorithm. It tackles the limitations imposed on the multi-view systems based on commodity 3D cameras, which lead to acquiring noisy and LR measurements. It produces enhanced HR and full 3D reconstructions of dynamic scenes. Figure 7.1 gives an overview of this algorithm. After upsampling the acquired LR point cloud $\mathcal{L}_t$ to get $\tilde{\mathcal{H}}_t$, using (7.2), we estimate the non-rigid deformations which register the enhanced HR result of previous iteration $\hat{\mathcal{H}}_{t-1}$ with $\tilde{\mathcal{H}}_t$. This registration is used to establish point-to-point correspondences between $\tilde{\mathcal{H}}_t$ and $\hat{\mathcal{H}}_{t-1}$, which allows to track and filter the position and motion of each point in $\tilde{\mathcal{H}}_t$. For this purpose, we use the CPD algorithm [132] which is a probabilistic method, wherein the matching of two point clouds is considered a probability density estimation problem [132]. It provides better registration accuracy than the registration algorithm, proposed in [97], which is used in KinectDeform and VI-KinectDeform. The CPD algorithm non-rigidly registers $\hat{\mathcal{H}}_{t-1}$ to $\tilde{\mathcal{H}}_t$, which is followed by a nearest neighbor search for establishing point-to-point correspondences. For per-point refinement via tracking, in this work, we use a Kalman filter [60], which performs prediction and correction for each 3D point's motion and position using the point-to-point correspondence information. This results in a noise-free but blurred estimate of $\mathcal{H}_t$ [31]. We use a novel 3D BTV regularization to perform deblurring and produce a noise-free HR estimate $\hat{\mathcal{H}}_t$. After that a motion correction step using updated point positions in $\hat{\mathcal{H}}_t$ is also carried out. These steps are repeated for every measurement $\mathcal{L}_t$, and hence result in a recursive process which enhances the resolution and quality of $\mathcal{L}_t$ using the previous result.

In what follows, we describe the method for per-point tracking using the correspondence information provided by the non-rigid registration algorithm.

7.3.2 Per-point Refinement via Tracking

For simplification of notation, in what follows we remove the point indices i, i.e., $\mathbf{r}_t^i \equiv \mathbf{r}_t$, $\forall \mathbf{r}_t^i \in \mathbb{R}^3$. We assume that the non-rigid registration step, in Figure 7.1, establishes point-to-point correspondences between the points $\tilde{\mathbf{p}}_t$ and $\hat{\mathbf{p}}_{t-1}$. Now the measurement model for each point follows from (7.2) such that:

$$\tilde{\mathbf{p}}_t = \mathbf{p}_t + \mathbf{n}_t, \tag{7.5}$$

where $\mathbf{n}_t = (n_{(x,t)}, n_{(y,t)}, n_{(z,t)})^\intercal$ represents per coordinate independent Gaussian noise which affects each measured point $\tilde{\mathbf{p}}_t$ such that $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}_3, \mathbf{C})$ is a 3-dimensional noise vector where $\mathbf{0}_3$ is a 3D null vector, and $\mathbf{C} = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix}$ is the covariance matrix. The per-point dynamic model follows from (7.3) such that:

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \mathbf{w}_t, \tag{7.6}$$

where $\mathbf{w}_t$ is the noisy version of the innovation. We propose to treat each 3D point $\mathbf{p}_t$ in motion as an independent dynamic system decorrelated from other 3D points in the scene. The state $\mathbf{s}_t$ of this dynamic system is defined by the position $\mathbf{p}_t = (x_t, y_t, z_t)^\intercal$ and the velocity $\mathbf{v}_t = (v_{(x,t)}, v_{(y,t)}, v_{(z,t)})^\intercal$ of the corresponding 3D point such that $\mathbf{s}_t = (x_t, v_{(x,t)}, y_t, v_{(y,t)}, z_t, v_{(z,t)})^\intercal$. We propose to use the per point correspondence together with the measurement and dynamic models, and their corresponding measurement and motion uncertainties, to update and filter the system state using a Kalman filter [60].

Following from (7.5), the measurement model for state $\mathbf{s}_t$ is defined as:

$$\tilde{\mathbf{p}}_t = \mathbf{B}.\mathbf{s}_t + \mathbf{n}_t, \text{where } \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{7.7}$$

In this work we assume a constant velocity model, where the acceleration $\mathbf{a}_t$ of the point $\mathbf{p}_t$ is a random vector such that $\mathbf{a}_t \sim \mathcal{N}(\mathbf{0}_3, \mathbf{C}_\mathbf{a})$ where $\mathbf{C}_\mathbf{a} = \begin{pmatrix} \sigma_{a_x}^2 & 0 & 0 \\ 0 & \sigma_{a_y}^2 & 0 \\ 0 & 0 & \sigma_{a_z}^2 \end{pmatrix}$. Considering a time step Δt the dynamic model in (7.6) can be written as:

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \mathbf{v}_{t-1}\Delta t + \frac{1}{2}\mathbf{a}_t \Delta t^2, \tag{7.8}$$

and the corresponding velocity is:

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \mathbf{a}_t \Delta t, \tag{7.9}$$

which can, in turn, be written in the following matrix form:

$$\mathbf{s}_t = \mathbf{D}\mathbf{s}_t + \boldsymbol{\alpha}_t, \text{such that } \mathbf{D} = \begin{pmatrix} \mathbf{D}_x & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{D}_y & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \mathbf{D}_z \end{pmatrix}, \tag{7.10}$$

where $\mathbf{D}_x = \mathbf{D}_y = \mathbf{D}_z = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$. Moreover, $\boldsymbol{\alpha}_t$ represents the process error, such that $\boldsymbol{\alpha}_t \sim \mathcal{N}(\mathbf{0}_6, \mathbf{Q})$ where $\mathbf{0}_6$ is a 6 dimensional null vector and $\mathbf{Q} = \begin{pmatrix} \sigma^2_{a_x}A & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \sigma^2_{a_y}A & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} & \sigma^2_{a_z}A \end{pmatrix}$, where $\mathbf{A} = \Delta t^2 \begin{pmatrix} \Delta t^2/4 & \Delta t/2 \\ \Delta t/2 & 1 \end{pmatrix}$. Now using the standard Kalman equations, the prediction of the next state is given as:

$$\begin{cases} \hat{\mathbf{s}}_{t|t-1} = \mathbf{D}\mathbf{s}_{t-1|t-1}, \\ \hat{\mathbf{P}}_{t|t-1} = \mathbf{D}\mathbf{P}_{t-1|t-1}\mathbf{D}^\intercal + \mathbf{Q}, \end{cases} \tag{7.11}$$

where $\mathbf{P}_{t-1|t-1}$ is the covariance matrix corresponding to the previous state $\mathbf{s}_{t-1|t-1}$ and $\hat{\mathbf{P}}_{t|t-1}$ is the covariance matrix corresponding to the predicted state $\hat{\mathbf{s}}_{t|t-1}$. The error in the predicted state $\hat{\mathbf{s}}_{t|t-1}$ is corrected by comparing it with the observed measurement $\tilde{\mathbf{p}}_t$ based on the Kalman gain matrix $\mathbf{G}_{t|t}$ which is computed as follows:

$$\mathbf{G}_{t|t} = \hat{\mathbf{P}}_{t|t-1}\mathbf{B}^\intercal \left(\mathbf{B}\hat{\mathbf{P}}_{t|t-1}\mathbf{B}^\intercal + \mathbf{C}\right)^{-1}, \tag{7.12}$$

using this gain $\mathbf{G}_{t|t}$, the corrected state vector and covariance matrix are obtained via:

$$\begin{cases} \mathbf{s}_{t|t} = \hat{\mathbf{s}}_{t|t-1} + \mathbf{G}_{t|t}(\tilde{\mathbf{p}}_t - \mathbf{B}\hat{\mathbf{s}}_{t|t-1}), \\ \mathbf{P}_{t|t} = \hat{\mathbf{P}}_{t|t-1} - \mathbf{G}_{t|t}\mathbf{B}\hat{\mathbf{P}}_{t|t-1}. \end{cases} \tag{7.13}$$

This per-point filtering is performed for each $\tilde{\mathbf{p}}_t$ to obtain the filtered, but blurred, estimate of $\mathcal{H}_t$, i.e., $\hat{\mathcal{H}}^f_t$. Similarly, we get the filtered 3D velocity estimates for all points i.e., $\hat{\mathcal{V}}^f_t$, where $\hat{\mathcal{V}}^f_t$ contains U velocity vectors. It is to be noticed that since the measurement noise and the process noise affect each coordinate of the 3D point independently, the per point Kalman filtering can be split into per coordinate Kalman filtering. This decreases the complexity of computation of the Kalman gain matrix $\mathbf{G}_{t|t}$ for each point.

7.3.2.1 Depth Dependent Measurement Noise

The measurement model in (7.5) assumes per coordinate independent Gaussian noise affecting each 3D point $\mathbf{p}_t$. In reality and as explained in Section 2.1.1 the 3D points are computed from depth images acquired via commodity 3D cameras built on structured-light or time-of-flight principles [18–20]. The acquired per-point depth measurement, i.e., $\tilde{\mathbf{q}}_t = (\tilde{u}_t, \tilde{v}_t, \tilde{z}_t)^\intercal$ is defined by the approximated pixel position $(\tilde{u}_t, \tilde{v}_t)$, in the depth image, and the measured depth value $\tilde{z}_t$ where:

$$\tilde{\mathbf{q}}_t = \mathbf{q}_t + \dot{\mathbf{n}}_t, \tag{7.14}$$

where $\dot{\mathbf{n}}_t = (n_{(u,t)}, n_{(v,t)}, n_{(z,t)})^\intercal$ represents noise in the measured pixel position and depth value. Let us consider a structured-light depth camera [19], for which the depth measurement $\tilde{z}_t$ suffers due to noise $n_{(d,t)}$ in disparity d as explained in Section 2.1.1, which is the distance (in pixels) between locations of a point in observed and projected pattern, via the relation [84, 138]:

$$n_{(z,t)} = -\frac{z_t^2}{f.b} n_{(d,t)}, \tag{7.15}$$

where f is camera's horizontal focal length, b the baseline distance between the camera and the projector, and $n_{(d,t)}$ is the noise in the corresponding disparity measurement $\tilde{d}_t$. The main factor affecting both the pixel and disparity measurements is the noise due to quantization [84], therefore we can assume it to be drawn from independent Gaussian distributions such that $n_{(u,t)} \sim \mathcal{N}(0, \sigma_u^2)$, $n_{(v,t)} \sim \mathcal{N}(0, \sigma_v^2)$ and $n_{(d,t)} \sim \mathcal{N}(0, \sigma_d^2)$. This allows us to model the noise in depth measurement i.e., $n_{(z,t)} \sim \mathcal{N}(0, \sigma_{(z,t)}^2)$ where $\sigma_{(z,t)}^2 = (-\frac{z_t^2}{f.b})^2 \sigma_d^2$.

To convert the depth measurement $\tilde{\mathbf{q}}_t$ to the corresponding 3D position $\tilde{\mathbf{p}}_t$, the intrinsic matrix $\mathbf{K} = \begin{pmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix}$, where (f_u, f_v) represent the focal lengths (where $f = f_u$), and (c_u, c_v) represent center of camera's imager such that:

$$\tilde{\mathbf{p}}_t = \mathbf{K}^{-1}\tilde{\mathbf{Z}}_t\tilde{\mathbf{q}}_t = \mathbf{K}^{-1}\tilde{\mathbf{Z}}_t(\mathbf{q}_t + \dot{\mathbf{n}}_t), \tag{7.16}$$

where $\tilde{\mathbf{Z}}_t = \begin{pmatrix} \tilde{z}_t & 0 & 0 \\ 0 & \tilde{z}_t & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $\tilde{z} = z + n_{(z,t)}$. Therefore the measurement model for each 3D point can now be defined as:

$$\tilde{\mathbf{p}}_t = \mathbf{p}_t + \mathbf{n}'_t, \tag{7.17}$$

where:

$$\mathbf{n}'_t = \begin{pmatrix} \frac{z_t n_{(u,t)} + (u_t - c_u) n_{(z,t)} + n_{(z,t)} n_{(u,t)}}{f_u} \\ \frac{z_t n_{(v,t)} + (v_t - c_v) n_{(z,t)} + n_{(z,t)} n_{(v,t)}}{f_v} \\ n_{(z,t)} \end{pmatrix}. \tag{7.18}$$

Here $\mathbf{n}'_t \sim \mathcal{N}(\mathbf{0}_3, \mathbf{C}'_t)$ where the entries of covariance matrix $\mathbf{C}'_t$ are defined as:

$$\begin{cases} cov(n_{(x,t)}, n_{(x,t)}) = \left(\frac{z_t^2\sigma_u^2 + (u_t - c_u)^2\sigma_{(z,t)}^2 + \sigma_u^2\sigma_{(z,t)}^2}{f_u^2}\right), \\ cov(n_{(y,t)}, n_{(y,t)}) = \left(\frac{z_t^2\sigma_v^2 + (v_t - c_v)^2\sigma_{(z,t)}^2 + \sigma_v^2\sigma_{(z,t)}^2}{f_v^2}\right), \\ cov(n_{(z,t)}, n_{(z,t)}) = \sigma_{(z,t)}^2 \\ cov(n_{(x,t)}, n_{(y,t)}) = \frac{(u_t - c_u)(v_t - c_v)}{f_u f_v}\sigma_{(z,t)}^2, \\ cov(n_{(x,t)}, n_{(z,t)}) = \frac{(u_t - c_u)}{f_u}\sigma_{(z,t)}^2, \\ cov(n_{(y,t)}, n_{(z,t)}) = \frac{(v_t - c_v)}{f_v}\sigma_{(z,t)}^2, \end{cases} \tag{7.19}$$

where $cov(.,.)$ computes the covariance between two random variables. This covariance matrix, specific to each point, can therefore be replaced in (7.12) when dealing with data acquired from depth cameras. To compute this covariance matrix, the noise-free pixel and depth values are required, but are not available in practice. Therefore, we propose to use the measured pixel and depth values instead, which are the closest approximation of the noise-free values we can get. Using the $\mathbf{C}'_t$ increases complexity of the proposed approach as now we have to deploy a Kalman filter per point, instead of per coordinate which was the case previously, but it captures the noise characteristics of depth cameras more accurately.

7.3.3 Proposed 3D BTV Deblurring

Per-point refinement via tracking discussed in Section 7.3.2 does not explicitly cater for blurring in the measurement model in (7.2) [31]. Furthermore, blurring artifacts are introduced due to treating each point separately which affects the global smoothness property of point clouds [135]. This results in filtered but blurred estimates of 3D point positions in $\mathcal{H}_t$, i.e., $\hat{\mathcal{H}}_t^f$, together with the corresponding velocity estimates, i.e., $\hat{\mathcal{V}}_t^f$. Therefore after per-point tracking, at every time-step Δt, it is necessary to carry out deblurring and regularization of position and motion estimates at hand to produce deblurred and globally smooth estimates [135]. We tackle the problem of deblurring by using a novel method for 3D BTV regularization of point positions. The regularized position estimates are used to correct the motion estimates. Please refer to Chapter 8 for details of this method. We carry out the 3D BTV regularization of position estimates $\hat{\mathcal{H}}_t^f$ to get a deblurred point cloud $\hat{\mathcal{H}}_t$.

In the next step we want to use the deblurred point cloud $\hat{\mathcal{H}}_t$ to correct the per point constant velocities estimates in $\hat{\mathcal{V}}_t^f$ to get $\hat{\mathcal{V}}_t$. For this purpose we use $\hat{\mathcal{H}}_t$ and the previous

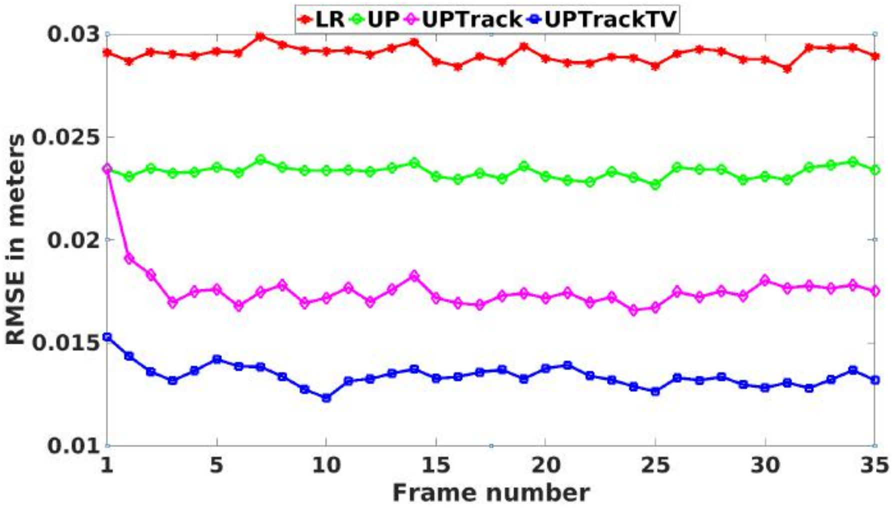


FIGURE 7.2: Comparison of results of different steps of the proposed dynamic filtering pipeline, as shown in Figure 7.1, on 35 LR frames of the "Samba" dataset [120] with zero-mean Gaussian noise of standard deviation 3cm added to each coordinate of 3D points independently. The steps include dense upsampling (UP), UP with per-point tracking using a Kalman filter (UPTrack), and UP with per-point tracking and 3D BTV deblurring (UPTrackTV). Per-point tracking alone is not able to handle system blur, therefore the proposed method of per-point tracking together with 3D BTV deblurring produces the best results. The SR factor is $o = 4$.

result $\hat{\mathcal{H}}_{t-1}$ to compute the per point corrected velocities estimates via:

$$\hat{\mathbf{v}}_t^i = (\hat{\mathbf{p}}_t^i - \hat{\mathbf{p}}_{t-1}^i)/\Delta t. \tag{7.20}$$

These corrected velocity estimates are used to get the per-point corrected state estimates which are then used in the next iteration.

7.4 Experiments and Results

In this section we present the results of the quantitative and qualitative analysis of performance of the proposed recursive dynamic 3D SR method using both synthetic and real experimental data. The data is in the form of 3D videos and contains non-rigid objects undergoing local motions of various complexities. We start by analyzing the results of our experiments on synthetic data which includes evaluation of different steps of the proposed method and its comparison with the state-of-art methods. This is followed by an analysis of results of the proposed method using real data acquired by cameras in a multi-view system. We show the ability of the proposed 3D SR method to enhance LR and noisy 3D reconstructions of non-rigid objects undergoing local motions as well as significant topology changes.

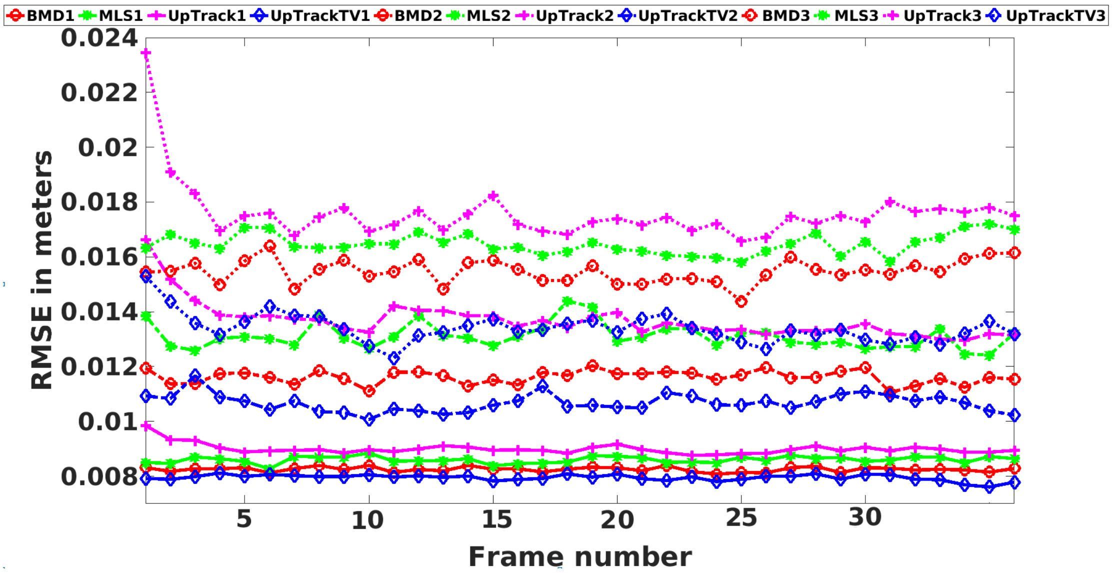


FIGURE 7.3: Comparison of the proposed technique with the state-of-art methods for enhancement of 3D measurements, corresponding to non-rigid objects, affected by noise of varying magnitude. 35 LR frames of the "Samba" dataset [120], with zero-mean Gaussian noise of standard deviations $1cm$, $2cm$ and $3cm$ added to each coordinate of 3D points independently, are used respectively. The SR factor is $o = 4$. Two static filtering methods namely Bilateral Mesh Denoising (BMD) [139] and Moving Least Squares (MLS) [140] are compared with the proposed recursive and dynamic SR method with (UPTrackTV) and without (UPTrack) the 3D BTV deblurring. BMD1 is the result of BMD on data affected by Gaussian noise of standard deviation $1cm$, and so on. Results show that UPTrackTV provides the best performance, as compared to the other methods, across all noise levels with its comparative performance improvement increasing with increasing data noise. This is due to its ability to tackle noisy artifacts locally as well as globally, in contrast with other methods which are mainly local in nature and hence, are unable to tackle high magnitude of noise in the data.

7.4.1 Evaluation on Synthetic Data

In this section we analyze the performance of the proposed method, using synthetic data with available ground truth, both qualitatively and quantitatively. This performance analysis includes analyzing the affects of different steps of the proposed pipeline followed by a comparison with the state-of-art filtering methods under varying noise and SR levels.

We use the "Samba" dataset [120] which contains high quality meshes from which HR 3D point clouds, representing full 3D reconstructions of real scenes of a non-rigid human body undergoing smooth and non-smooth local motions over time which we call the ground truth (GT), are extracted. We use 35 frames for our experiments.

We start by analyzing the effects of different steps of the proposed SR pipeline as shown in Figure 7.1. For this purpose, the GT point clouds are first downsampled by a SR factor $o = 4$, then zero-mean Gaussian noise is added independently to each coordinate of 3D points, of the downsampled GT clouds, with standard deviations $\sigma_x = \sigma_y = \sigma_z =$

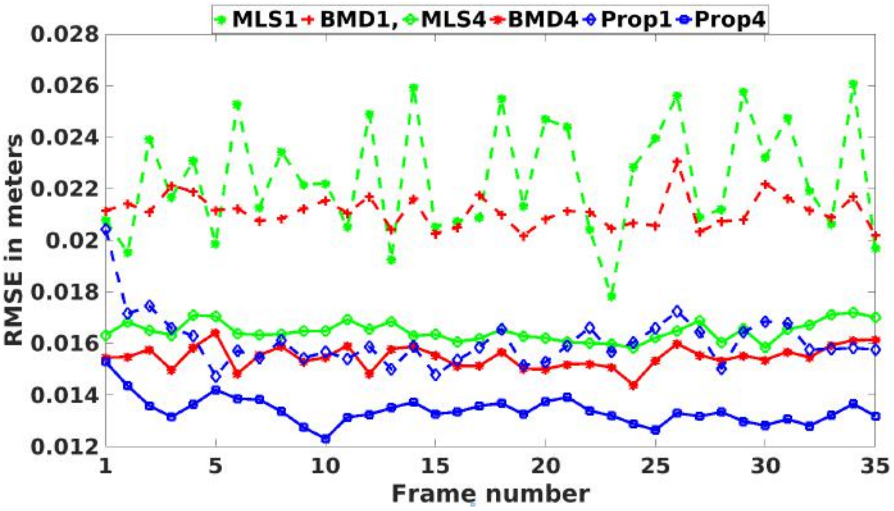


FIGURE 7.4: Comparison of the proposed technique with the state-of-art methods for 3D point cloud enhancement for different SR factors. 35 LR frames (downsampled by a factor $o = 4$) of the "Samba" dataset [120], with zero-mean Gaussian noise of standard deviation 3cm added to each coordinate of 3D points independently, are used. The filtering is performed on the input data upsampled by a factor $o = 1$ and $o = 4$, respectively. Two static filtering methods namely BMD [139] and MLS [140] are compared with the proposed recursive and dynamic SR method. BMD1 is the result of BMD on input LR and noisy data upsampled by a factor $o = 1$, and so on. Although the proposed method has comparative performance at $o = 1$ with respect to the performance of the state-of-art methods at $o = 4$, it achieves best results at $o = 4$.

$3cm$. These LR noisy point clouds are given as input and SR results of upsampling based on mesh edge division using GT mesh information with o=4, upsampling and per-point tracking using a Kalman filter, and upsampling, per-point tracking together with multi–Level iterative 3D BTV deblurring, are obtained. Root Mean Squared Error (RMSE) for the result of each method is computed with respect to the HR GT data. Figure 7.2 shows the RMSE per frame for each of the steps mentioned before. Although per-point tracking using a Kalman filter recursively enhances the 3D point clouds and requires only 3-4 frames to converge, its performance is limited by its inability to handle system blur and its ability to introduce noisy artifacts. Adding a deblurring step based on 3D BTV regularization solves this problem and produces the best results.

In the next experiment, we perform a comparison of the state-of-art static 3D point cloud enhancement methods with the proposed dynamic SR scheme using the data affected by noise of varying magnitude. The GT point clouds are downsampled and upsampled by a factor $o = 4$ as explained above. Zero-mean Gaussian noise of standard deviations $\sigma_x = \sigma_y = \sigma_z = 1cm$, $2cm$ and $3cm$, is added to the downsampled GT point clouds, respectively. In addition to the proposed method, we use static filtering schemes based on Bilateral Mesh Denoising (BMD) [139] and Moving Least Squares (MLS) [140] to enhance the upsampled point clouds. RMSE per frame for results of BMD, MLS, proposed method with per-point tracking only and proposed method with per-point

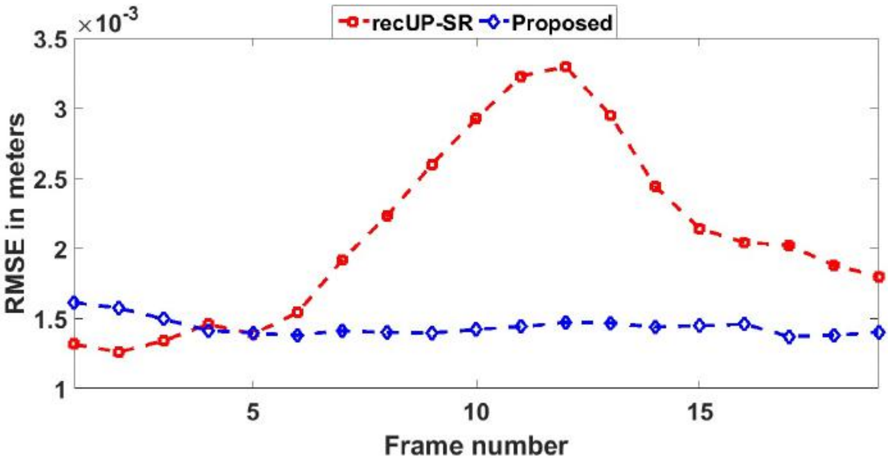


Figure 7.5: Comparison of the proposed technique with the dynamic state-of-art method, namely recUP-SR [135], for enhancement of 3D/depth videos generated by simulating a mono-view depth system using the “Samba” dataset [120]. 19 LR depth frames with zero-mean Gaussian noise of standard deviation 3cm added to the depth measurements, are used [135]. The results show improved accuracy of the proposed method as compared to recUP-SR.

tracking and 3D BTV deblurring, are plotted in Figure 7.3. Although the proposed method, with per-point tracking only, is able converge more quickly as the noise level decreases, its performance remains worse than the other methods due to introduction of blurring artifacts. The performance of BMD and MLS starts to get worse with the increase in noise magnitude due to their local nature and their inability to handle highly noisy artifacts. The proposed method with per-point tracking and 3D BTV blurring provides the best performance at all noise levels and can produce globally smooth and feature preserving point clouds even at high noise levels.

In Figure 7.6, we plot mesh reconstructions of an example frame (number 33) which are obtained as a result of; adding independent Gaussian noise to each coordinate of the downsampled GT data with standard deviation of $1cm$, dense upsampling of LR noisy data with $o = 4$ only, upsampling and BMD, upsampling and MLS, proposed pipeline with $o = 4$, together with HR ground truth meshes. The meshing of point clouds is carried out by using the mesh information available for GT. The results clearly show that the proposed technique produces enhanced, smoother and feature preserving reconstruction as compared to other methods. BMD and MLS fail to preserve smaller features such as hands, arm, nose, etc. To investigate further the quality of reconstructions obtained via the methods mentioned above we calculate the RMSE for different body parts for the reconstructed example Frame#33. Table 7.1 shows these results from which it is clear that even for separate body parts the conclusions drawn above hold.

Figure. 7.6 shows plots of 3D mesh reconstruction of an example Frame#33, from different views, obtained as result of the proposed method. The proposed method tackles

TABLE 7.1: 3D RMSE in mm for different body parts, of Frame#33 of the "Samba" dataset [120], using different methods as shown in Figure 7.6.

	Arm	Leg	Torso	Full body
LR	11.31	11.61	11.03	11.48
UP	9.43	10.23	9.55	9.84
BMD	9.22	9.03	7.46	8.23
MLS	10.07	8.83	7.75	8.69
Proposed	**8.05**	**7.55**	**7.26**	**7.83**

non-rigidities and recursively filters the noisy measurements to produce super-resolved and enhanced complete 3D reconstructions of dynamic objects/scenes.

In the next experiment, we perform a comparison of the state-of-art static 3D point cloud enhancement methods, i.e., BMD and MLS, with the proposed dynamic SR scheme for different SR factors. This means that GT point clouds are first downsampled by a SR factor $o = 4$, then zero-mean Gaussian noise is added independently to each coordinate of 3D points, of the downsampled GT clouds, with standard deviations $\sigma_x = \sigma_y = \sigma_z = 3cm$. Filtering is carried on this data with upsamling factors of $o = 1$ and $o = 4$, respectively. RMSE per frame is plotted in Figure 7.4. Results show that proposed method clearly outperforms both BMD and MLS when used on same data. The results also show that even at upsampling factor $o = 1$ the proposed dynamic scheme gives comparative performance with respect to both BMD and MLS used on upsampled noisy data with $o = 4$. This is outperformed by applying the proposed dynamic filtering scheme at $o = 4$. The reason for this is that at $o = 1$, the method recursively denoises the noisy input. On the other hand, at $o = 4$, the method applies the full recursive dynamic super-resolution pipeline which together with denoising, enhances the quality of data by preserving useful features.

Lastly, we perform a comparison of the proposed dynamic 3D SR method with the state-of-art dynamic depth SR method proposed by Al Ismaeil et al. [135], called recUP-SR. We again make use of the "Samba" dataset [120], and simulate a depth camera, placed at a distance of approx. 2 meters, in V-Rep [80] to generate a mono-view synthetic depth sequence [135]. This GT depth sequence is downsampled by a factor $o = 4$, and zero mean Gaussian noise of variance $\sigma_z = 3cm$ is added to the depth measurements. This LR noisy depth sequence is given as input to recUP-SR, and is converted to a 3D sequence via the known camera parameters and given as input to the proposed method. To compare the super-resolved (by a factor $o = 4$) results of both methods the resulting depth sequence from recUP-SR and the GT depth sequence are converted to 3D sequences as explained before. After that per frame RMSE for the result of each method with respect to the 3D GT is computed. The results are reported in the Figure 7.5. The results show the robustness and improved accuracy of the proposed method as compared to recUP-SR.

7.4.2 Evaluation on Real Data

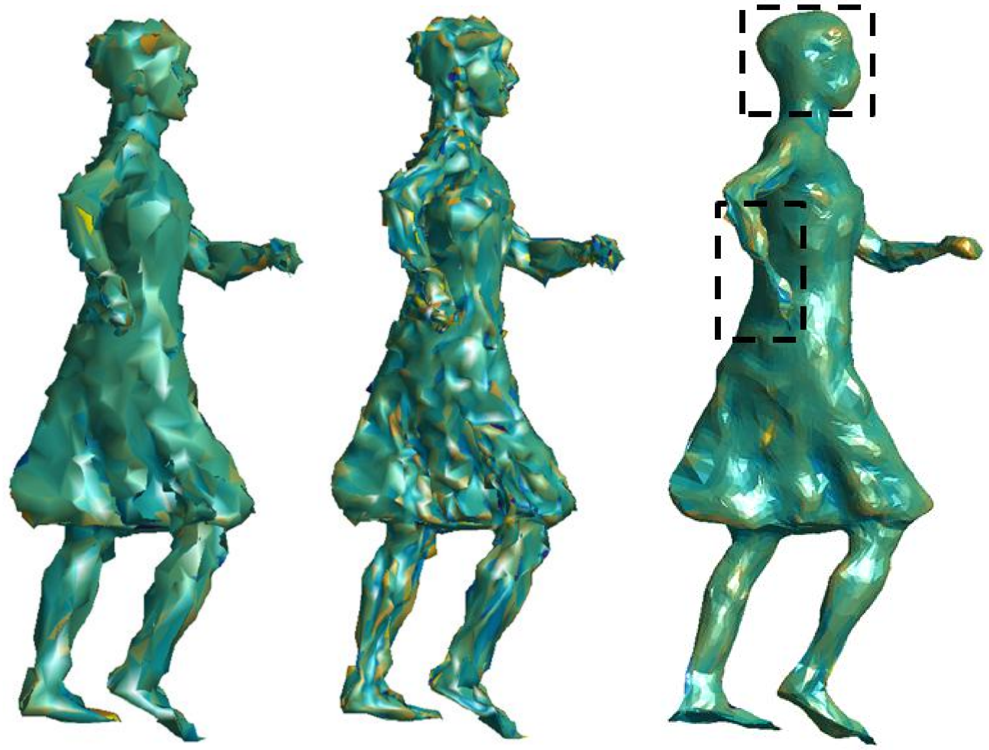
In this section we analyze the performance of the proposed method using real data acquired via multi-view systems composed of photometric and commodity depth cameras, respectively. In addition to showcasing the ability of the proposed method to enhance the quality of LR and noisy data to produce smooth and feature preserving full 3D reconstructions of non-rigid objects, this experimental analysis also demonstrates the capabilities of the proposed method to produce accurate and enhanced 3D reconstructions of objects with changing topologies.

In the first experiment we use full 3D point-clouds extracted from meshes of the "adult child ball" scene from the Inria4D dataset [141]. This dataset is acquired via a fully calibrated multi-view system based on photometric RGB cameras. This dataset has two characteristics; the resolution of data is quite low (approx. 10000 points per scene) resulting in non-smooth surfaces, and it contains an object, i.e. a ball, with changing topology as shown in Figure 7.8. Due to these characteristics this dataset is very challenging for the class of methods to which belong the works by Dou and Fuchs [3, 133] and DynamicFusion [62]. These methods do not explicitly target LR data and are very sensitive to objects with changing topologies due to their design of always fusing the current measurement with the first frame which is considered to be the reference. The proposed method, on the other hand, explicitly targets LR 3D data and produces HR, smooth and feature preserving 3D reconstructions as shown in Figure 7.8. Moreover, it works by recursively fusing the current measurement and the result of the previous iteration/time-step and hence, can accurately reconstruct objects, in this case a ball, with changing topologies.

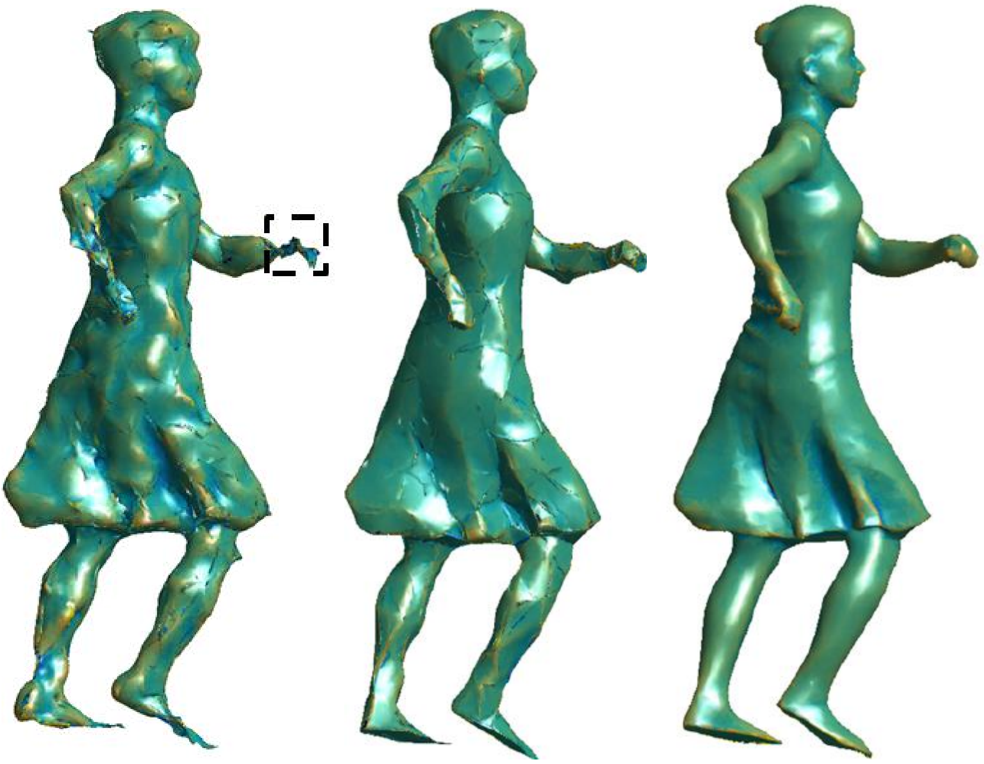
In the next experiment we use point clouds from the full 3D video of the "jumping in place" action performed by a human subject from the Berkeley Multimodal Human Action Database (MHAD) [142]. This dataset is acquired via a fully calibrated multi-view system composed of two Kinect version1 cameras placed at opposite corners of the acquisition space. As explained in Section 2.1.1 and Section 2.1.2, the depth acquisition system of Kinect version1 is based on structured-light principle and suffers from depth dependent measurement noise. The distance of Kinect cameras from the subjects in MHAD's multi-view setup is approximately $3.5 - 4$ meters. This results in highly noisy 3D measurements with non-smooth surfaces and diminished features as shown in Figure 7.9. Figure 7.9 also shows the point clouds which are received as the output of the proposed algorithm. The input data is upsampled by a factor $o = 1.5$. Moreover, to tackle the depth dependent measurement noise specific to Kinect version1 cameras the measurement model presented in Section 7.3.2.1 is used during the per-point tracking step. The resolution enhancement together with per-point tracking and 3D BTV deblurring results in point clouds which are relatively noise-free, have smoother surfaces with less holes/gaps and better preserved features/details.

7.5 Conclusion

In this chapter we have presented a framework for acquiring high quality and full 360° 3D reconstructions of dynamic scenes containing non-rigid objects undergoing large local motions/deformations. We target noisy and LR data acquired from commodity 3D cameras in a multi-view system. This framework is based on a recursive dynamic multi-frame 3D SR algorithm which is capable of filtering out the noise as well as enhancing the resolution of the raw measurements obtained from multi-view systems. The proposed algorithm tracks and filters the position and motion of every 3D point recursively, hence making use of complete 3D characteristics of the input data. It is able to handle generic 3D as well as structured-light sensing based depth specific noise in 3D measurements. Moreover, it uses a 3D BTV regularization for deblurring and smoothing of the point clouds after per point tracking. Quantitative and qualitative evaluation of the proposed framework on both simulated and real data shows its improved performance as compared to state-of-art methods, and its ability to tackle highly noisy and LR data in order to produce noise-free, smooth and feature preserving full 3D reconstructions.

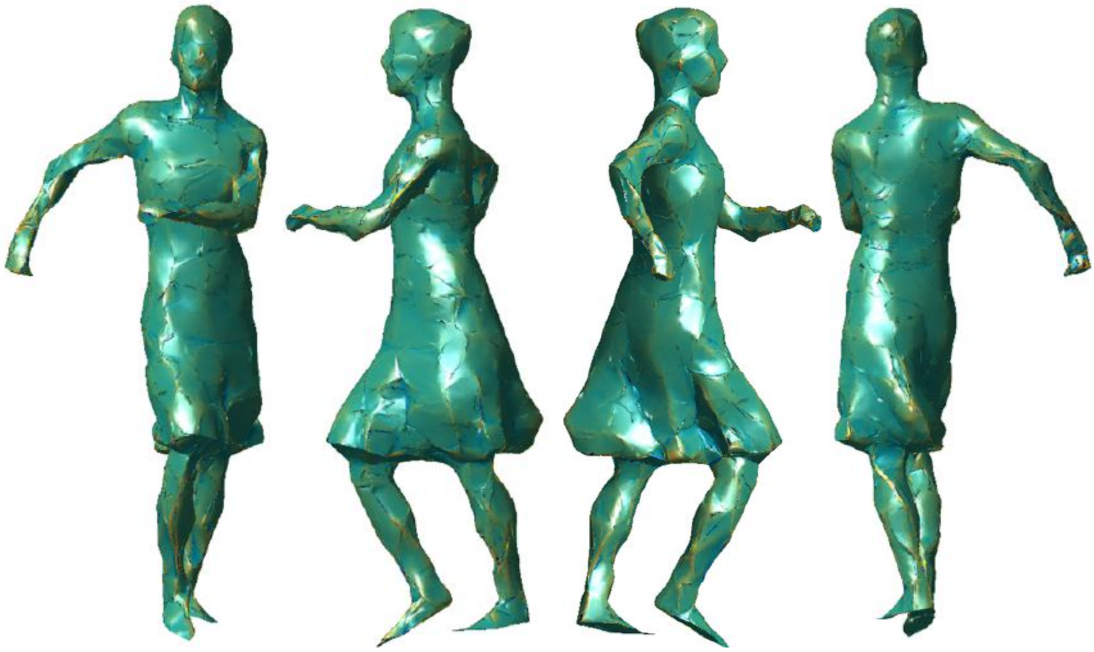


(a) LR (b) UP (c) BMD



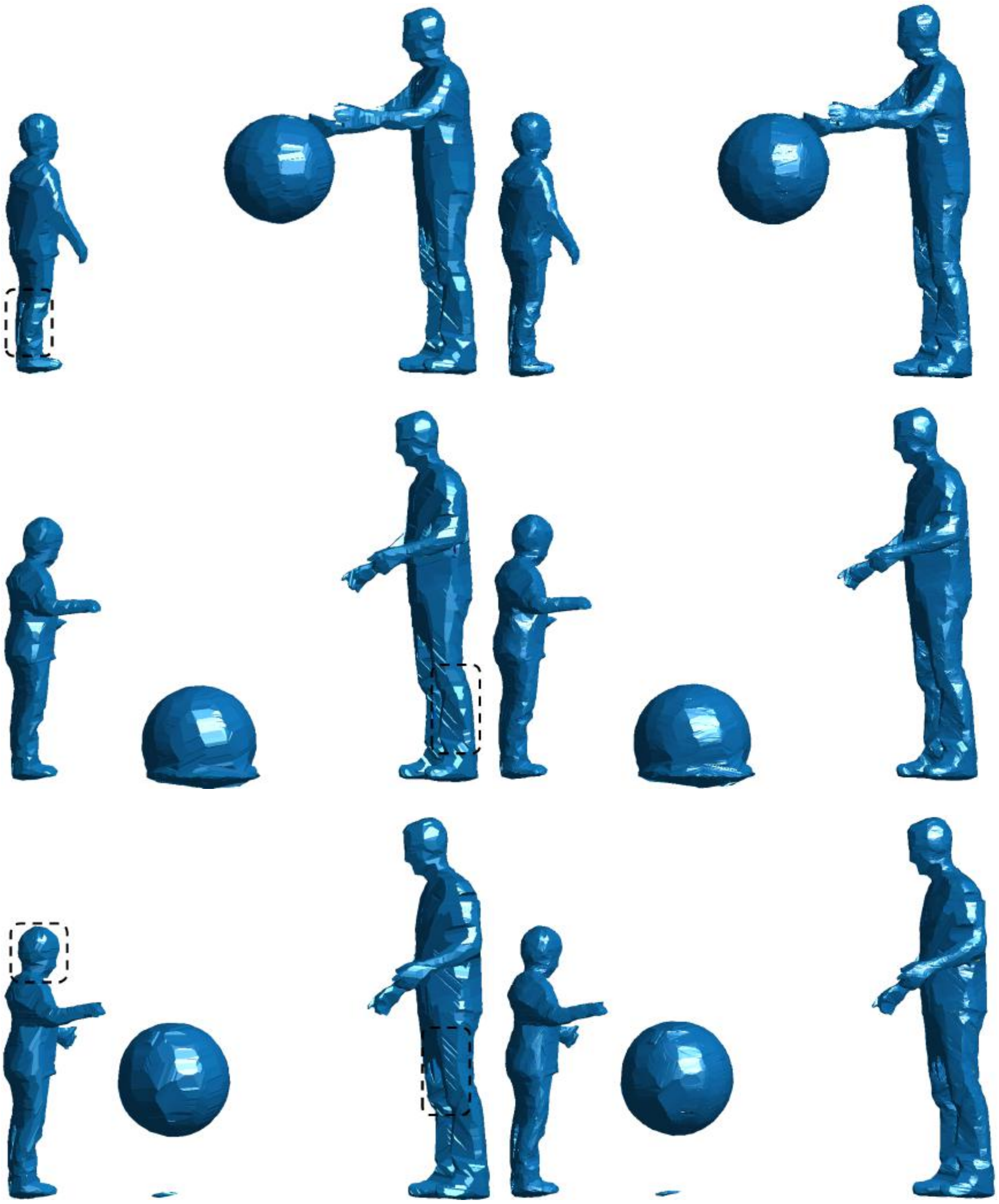
(d) MLS (e) Proposed (f) GT

Figure 7.6: 3D mesh plots of a super-resolved resultant Frame#33 from the ”Samba” dataset [120] after: b) dense upsampling (UP), c) Bilateral Mesh Denoising (BMD), d) Moving Least Squares (MLS) and e) Proposed recursive and dynamic SR scheme. a) is the 3D plot of LR noisy data, and e) is the GT HR mesh respectively. Proposed technique produces smooth, enhanced and feature preserving reconstruction as compared to the rest. The SR factor is $o = 4$. Display color-scale is based on mean surface curvature.



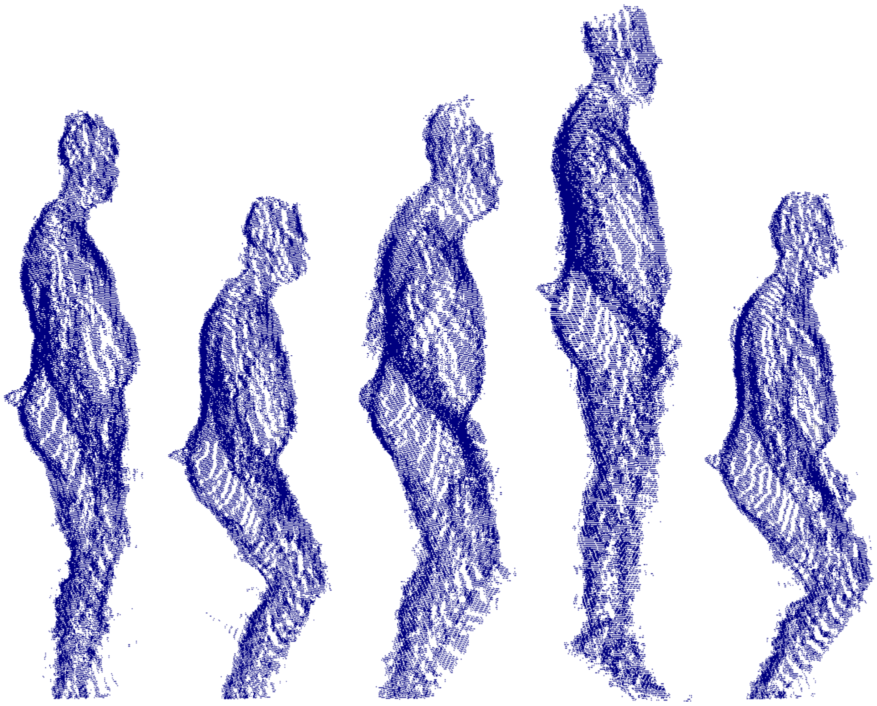
(a) Front (b) Left (c) Right (d) Back

FIGURE 7.7: 3D mesh plot of from different views of a super-resolved resultant full 3D point cloud (Frame#33) from the "Samba" dataset [120]. Display color-scale is based on mean surface curvature.

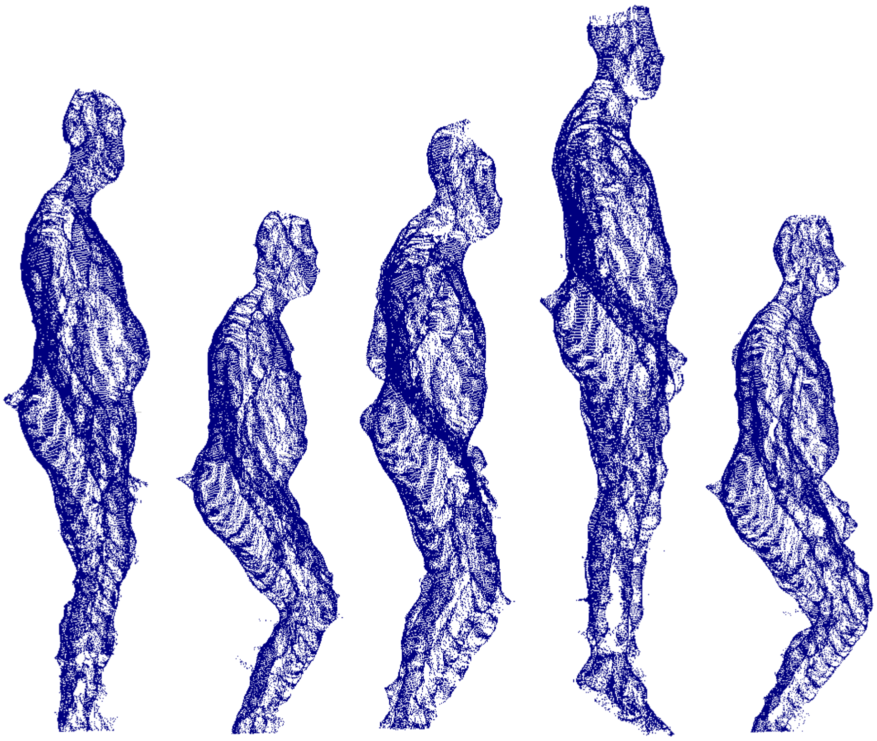


(a) LR (b) Proposed

Figure 7.8: 3D mesh plots of three LR frames (#5, #12 and #15) from the "Inria4D" dataset [141], i.e. the first column, and the corresponding super-resolved (using SR factor $o = 4$) results of the proposed algorithm, i.e. the second column. The input data has low-resolution which results in non-smooth surfaces, thick edges and loss of details. The results show super-resolved, smooth, and feature preserving 3D reconstructions of non-rigid objects. They also show ability of the proposed method to produce enhanced reconstructions of objects with changing topologies e.g., the ball in the above plots. Display color-scale is based on mean surface curvature.



(a) LR



(b) Proposed

FIGURE 7.9: Plots of LR 3D point-clouds of five frames (#6, #10, #18, #22 and #35) from the "Berkeley MHAD" Kinect dataset [142], i.e. the first row, and the corresponding super-resolved (using SR factor $o = 1.5$) results of the proposed algorithm, i.e. the second row. The input data suffers from high magnitude of noisy artifacts, in the form of non-smooth surface and jagged edges, due to large distance of the object from the cameras. The results show super-resolved, smooth, and feature preserving full 3D reconstructions of the human subject.

Chapter 8

Point Cloud Denoising via 3D Bilateral Total Variation Regularization

In this chapter, we discuss the problem of noise removal from unorganized 3D point clouds while preserving finer details to produce accurate 3D reconstructions of objects or scenes. The state-of-art unorganized 3D point cloud denoising methods are usually local in nature. The local nature of theses methods restricts their ability to tackle noisy artifacts and produce globally smooth 3D reconstructions while preserving salient features. Total variation regularization based global methods, such as BTV regularization, have been successful in denoising and deblurring color or depth images to produce globally smooth and edge preserving results [143]. These methods make use of organization of the data on the image grid to compute the total variation of all pixel values. In this work we propose to extend the image based BTV regularization to unorganized 3D point clouds. We tackle the challenge of computation of BTV of 3D points without the availability of local neighborhood or structure information by extracting local point patches and making use of elements from the local Bilateral Mesh Denoising (BMD) framework [139]. Quantitative and qualitative performance evaluation of the proposed algorithm shows that it outperforms state-of-art local point cloud denoising methods and produces accurate, globally smooth and features preserving 3D reconstructions.

8.1 Introduction

Modeling the 3D world around us accurately is a fundamental task in computer vision. It can be achieved via 3D sensors acquiring the geometric information in the form of 3D point clouds. In most cases the quality of this information declines due to various systematic and non-systematic factors. This challenge has been amplified by the availability of commodity depth cameras, such as Microsoft Kinect [18], Asus Xtion Pro Live [19] and PMD camboard nano [20] etc, which though easily available and widely in use produce low quality and limited resolution geometric measurements. Therefore, there is a need for such algorithms which can filter the acquired 3D point clouds provided by such sensors to mitigate the noise affecting them, while preserving the feature information, in order to help produce as accurate 3D reconstructions of the scenes/objects as possible.

State-of-art methods suggest two main classes of 3D point cloud denoising techniques i.e., neighborhood based methods and projection based methods [144]. Neighborhood based methods find their roots in image based denoising methods and have been extended to 3D point clouds such as Bilateral Mesh Denoising (BMD) by Fleishman et al [145]. For each point, its neighbors are assigned weights using similarity measures based on distance and other geometric properties and a new value is computed via a weighted mean [146]. A non-local denoising based extension of this method where a weight based on patch based similarity measure approximated via Radial Basis Functions (RBFs) is added as proposed by Yoshizawa et al [147].

Projection based approaches, on the other hand, include the well-known Moving Least Squares method which was introduced by Levin [148]. This method works by finding the projection of each point on the underlying surface of the points in its local neighborhood [140]. Since it was first proposed, it has been a topic of interests for various researchers who have presented several insights and improvements to this method for preserving sharp geometric features [149–154].

The above mentioned methods, although able to produce relatively smooth and feature preserving 3D reconstructions, still face challenges when they encounter highly noisy data due to their local nature. This causes noisy artifacts in the resulting reconstructions. Total variation (TV) regularization is a class of global methods which has been proven to be very successful in denoising and deblurring noisy images, mainly due to their global smoothing and edge preserving abilities [155, 156]. It works by maximizing the data fidelity while minmizing the total variation amongst the pixel values in the image at the same time. Farisu et al. proposed to incorporate the concept of the bilateral filter [157] for images in TV regularization to propose Bilateral Total Variation (BTV) regularization. The proposed BTV regularization provides a priori knowledge to stabilize the ill-posed super-resolution (SR) image reconstruction problem [143]. It combines image restoration and edge preserving properties of TV and bilateral filter, while putting a check on local artifacts and over-smoothing, and preserves finer details [143]. It has since been used in various color and depth images based SR approaches [111, 113, 158–161].

In this work, we propose to extend the image based BTV regularization to unorganized 3D point clouds to produce globally smooth but feature preserving 3D reconstructions by mitigating the noise affecting the acquired point clouds. The color or depth image based BTV regularization approaches make use of data organization on the image grid, which provides the local neighborhood or structure information, to compute the BTV of color or depth values in the image [111, 143]. In the case of unorganized point clouds there is no information available about local structures or neighborhood which makes the task of computation of total variation of 3D points very challenging. In this work, we solve this problem by making use of the elements from BMD framework to extract the shape and geometric properties in local point patches for computation of BTV. Our experiments show that the proposed approach is able to produce globally smooth and feature preserving 3D reconstructions as compared to state-of-art methods.

8.2 Background and Problem Formulation

Given a noisy 2D image $\tilde{\mathbf{X}}$, the BTV regularization minimizes the following cost in order to produce an enhanced image:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \mu|\nabla\mathbf{X}| + \frac{1}{2}\|\mathbf{X} - \tilde{\mathbf{X}}\|_2^2, \tag{8.1}$$

which defines an L_2-optimization with an L_1-BTV regularization $|\nabla\mathbf{X}|$. $\nabla\mathbf{X}$ represents the discrete gradient of $\mathbf{X}$, $|.|$ denotes the L1-norm and μ is the regularization parameter. Using the structure and neighborhood information embedded in images, the BTV regulazier is easy to compute and is given by [143]:

$$|\nabla\mathbf{X}| = \sum_{n=-l}^{l}\sum_{m=0}^{l} a^{|n|+|m|}|\mathbf{X} - \mathbf{S}_u^p\mathbf{S}_v^q\mathbf{X}|, \tag{8.2}$$

where $\mathbf{S}_u^p$ and $\mathbf{S}_v^q$ are operators which shift the image $\mathbf{X}$ by p and q pixels in horizontal and vertical directions, respectively. The scalar a, $0 < a < 1$, controls the speed of decay. Our goal is to denoise an unorganized noisy 3D point cloud represented by an ordered point set $\tilde{\mathcal{H}}$ of size $U \in \mathbb{N}^*$, which is the noisy version of $\mathcal{H} = [\mathbf{p}^1, \cdots, \mathbf{p}^n]$, such that each $\mathbf{p}^i \in \mathbb{R}^3$, and $i \in \{1, \cdots, U\}$, via BTV regularization. There is however, no strucutre or neighborhood information which allows us to easily compute the gradient or BTV as in (8.2).

The BMD algorithm filters a 3D point by estimating the local surface normal and moving the point in the direction of that normal. The distance by which the query point is moved is computed as the weighted mean of the shortest distances of all points in its neighborhood to the plane which is tangent to the local surface approximated by the neighborhood. The weighted mean of the shortest distances $\hat{d^i}$ for a point $\mathbf{p}^i$ with a normal $\vec{\mathbf{u}}^i$ and neighborhood defined by Ω^i is computed as:

$$\hat{d^i} = \sum_{\mathbf{p}^j \in \Omega^i} \frac{w(\sigma_d, d^{ij}) w(\sigma_c, c^{ij}).d^{ij}}{\sum_{\mathbf{p}^j \in \Omega^i} w_d^{ij} w_c^{ij}}, \tag{8.3}$$

where $d^{ij} = (\vec{\mathbf{u}}^i)^\intercal (\mathbf{p}^i - \mathbf{p}^j)$ and the weight:

$$w(\sigma_d, d^{ij}) = \exp(-(d^{ij})^2 / 2\sigma_d^2), \tag{8.4}$$

where $d^{ij} = (\vec{\mathbf{u}}^i)^\intercal (\mathbf{p}^i - \mathbf{p}^j)$ is the shortest distance of $\mathbf{p}^j$ to the plane which is tangent, at $\mathbf{p}^i$, to the underlying surface sampled by the local patch of $\mathbf{p}^i$. The parameter σ_d is a constant thresholding factor. The weight $w(\sigma_d, d^{ij})$ serves to detect outliers and preserve the edge information by taking into account the change in curvature in the local patch of $\mathbf{p}^i$. The weight $w(\sigma_c, c^{ij})$, on the other hand, is defined as:

$$w(\sigma_c, c^{ij}) = \exp(-(c^{ij})^2 / 2\sigma_c^2), \tag{8.5}$$

where $c^{ij} = \|\mathbf{p}^i - \mathbf{p}^j\|$ is the Euclidean distance between $\mathbf{p}^i$ and $\mathbf{p}^j$, and σ_c is a constant thresholding factor. The weight $w(\sigma_c, c^{ij})$ serves to give more importance points which lie closer to $\mathbf{p}^i$. Once $\hat{d^i}$ is computed we get the filtered point $\hat{\mathbf{p}}^i$ via:

$$\hat{\mathbf{p}}^i = \mathbf{p}^i + \vec{\mathbf{u}}^i.\hat{d^i}. \tag{8.6}$$

Therefore, the problem at hand is to use the elements of BMD to compute BTV for unorganized 3D point clouds and use it to formulate the 3D BTV regularization.

8.3 3D Bilateral Total Variation Regularization

In this work we propose to tackle the problem of point cloud denoising by introducing a novel method for 3D BTV regularization. Given a noisy measurement $\tilde{\mathcal{H}}$, the proposed method is based on the following minimization framework:

$$\hat{\mathcal{H}} = \arg\min_{\mathcal{H}} \mu |\nabla \mathcal{H}| + \frac{1}{2} \|\mathcal{H} - \hat{\mathcal{H}}^f\|_2^2, \tag{8.7}$$

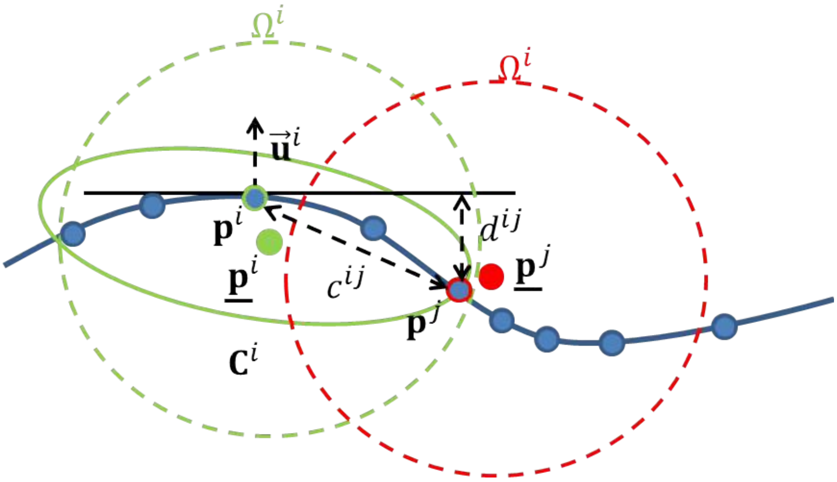


FIGURE 8.1: Illustration of main components for per point gradient computation on a 2D surface. $\mathbf{p}^i$ is the query point and $\mathbf{p}^j$ lies in its neighborhood. Their corresponding neighborhoods are represented by Ω^i and Ω^j and the local point patches corresponding to these neighborhoods are classified by the mean and covariance of points in them i.e., $(\underline{\mathbf{p}}^i, \mathbf{C}^i)$ and $(\underline{\mathbf{p}}^j, \mathbf{C}^j)$, respectively. c^{ij} is the Euclidean distance between $\mathbf{p}^i$ and $\mathbf{p}^j$ and d^{ij} is the shortest distance of $\mathbf{p}^j$ to the plane, tangent at $\mathbf{p}^i$ to the local patch of $\mathbf{p}^i$, defined via the normal vector $\vec{\mathbf{u}}^i$.

BTV regularization/denoising has been a topic of interest for researchers but most of the research has been restricted to organized color and depth images [111, 159–162], where the neighborhoods are well defined and the gradient, based on intensity or depth values, is easy to compute, e.g., via shift operators [31, 111]. In the current problem, $\hat{\mathcal{H}}^f$ is a set of unorganized 3D points without any connectivity or neighborhood information, therefore the extension of BTV regularization to 3D point clouds is not a straightforward task. We are interested in finding a gradient operator ∇, which computes the gradient per 3D point by taking into account the spatial and geometric properties of the local point patches in its neighborhood. Therefore, we choose ∇ such that it exploits the properties of local point patches based on their unique locations, geometry and curvature, as illustrated in Figure 8.1. We formulate the 3D BTV regularization such that:

$$
\begin{aligned}
|\nabla\mathcal{H}| &= \sum_{i,j} \|\nabla\mathbf{p}^{ij}\| \\
&= \sum_{\mathbf{p}^i} \sum_{\mathbf{p}^j \in \Omega^i} \frac{w(\sigma_d, d^{ij}) w(\sigma_c, c^{ij}) \|((\mathbf{p}^i - \underline{\mathbf{p}}^i) - (\mathbf{p}^j - \underline{\mathbf{p}}^j))\|}{\sum_{\mathbf{p}^j \in \Omega^i} w(\sigma_d, d^{ij}) w(\sigma_c, c^{ij})},
\end{aligned}
\tag{8.8}
$$

where Ω^i is the pre-computed neighborhood of $\mathbf{p}^i$. Each local patch corresponding to the neighborhood Ω^i of the query point $\mathbf{p}^i$ is characterized by the mean and covariance i.e., $(\underline{\mathbf{p}}^i, \mathbf{C}^i)$, of the points in it. Similarly the patch corresponding to $\mathbf{p}^j$ is characterized by $(\underline{\mathbf{p}}^j, \mathbf{C}^j)$. We assume that all points in $\mathcal{H}^f$ are equally distributed therefore we have $\mathbf{C}^i = \mathbf{C}^j$. Now we localize $\mathbf{p}^i$ and $\mathbf{p}^j$ by subtracting from them the corresponding means and then finding the difference between their local positions. This difference is then weighted by $w(\sigma_d, d^{ij})$ and $w(\sigma_c, c^{ij})$ as defined in (8.4) and (8.5), respectively.

The L_2-norm in (8.7) is convex and differentiable whereas the L_1-norm is convex and non-differentiable (non-smooth). Such type of problems cannot be solved using simple gradient-decent methods [162]. Therefore we use the Foward-Backward Splitting (FBS) method (also known as proximal gradient solver), which relies on computing a *proximal* operator for the non-smooth part of the problem, which is implemented using Fast Adaptive Shrinkage/Thresholding Algorithm (FASTA) [162]. $|\nabla\mathcal{H}|$ is first reformulated to a simpler form which is differentiable, by defining a vector $\mathbf{r}^{ij} \in \mathbb{R}^3$ and using Cauchy-Swartz inequality to write [162]:

$$\max_{\|\mathbf{r}^{ij}\|\leq 1} \langle \mathbf{r}^{ij}, \nabla\mathbf{p}^{ij}\rangle = \|\nabla\mathbf{p}^{ij}\|, \tag{8.9}$$

where $\mathbf{r}^{ij}$ is assumed to be parallel to $\nabla\mathbf{p}^{ij}$, having a unit norm. Using this definition of $\|\nabla\mathbf{p}^{ij}\|$ in (8.8) and (8.9) receptively, solving (8.7) is equivalent to finding:

$$\max_{\|\mathbf{r}^{ij}\|\leq 1} \arg\min_{\mathcal{H}} \mu\langle \mathcal{R}, \nabla\mathcal{H}\rangle + \frac{1}{2}\|\mathcal{H} - \hat{\mathcal{H}}^f\|_2^2 \tag{8.10}$$

where $\mathcal{R} = \{\mathbf{r}^{ij}\}$, and the inner minimization is now differentiable. The minimal value of $\mathcal{H}$ for a given value of $\mathcal{R}$ should satisfy $\mathcal{H} = \hat{\mathcal{H}}^f + \mu\nabla\cdot\mathcal{R}$ where $\nabla\cdot$ is the discrete divergence operator and can be computed by taking transpose of the gradient operator. We can reformulate (8.10) using the optimal value of $\mathcal{H}$ to get dual form of (8.7) such that:

$$\hat{\mathcal{R}} = \arg\min_{\|\mathcal{R}\|_\infty\leq 1} \frac{1}{2}\|\nabla\cdot\mathcal{R} - \frac{1}{\mu}\hat{\mathcal{H}}^f\|^2. \tag{8.11}$$

This problem is solved via the FBS method as explained in [162], and the final deblurred result is obtained via:

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}^f + \mu\nabla\cdot\hat{\mathcal{R}}. \tag{8.12}$$

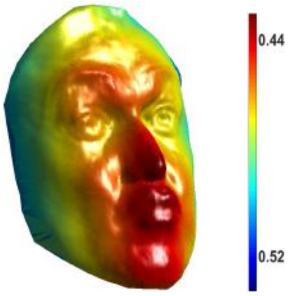
In the case ∇ is a linear operator it can be represented as a sparse matrix for which the corresponding discrete divergence operator can be computed by taking the transpose of this sparse matrix. This makes the solution of this problem very efficient. Therefore, for making ∇ linear we use the input $\hat{\mathcal{H}}^f$ to pre-compute the neighborhoods Ω^i and Ω^j, the weights $w(\sigma_d, d^{ij})$ and $w(\sigma_c, c^{ij})$, and the normals $\vec{\mathbf{u}}^i$, for all points. This method, although effective, is sensitive to the choice of parameters and can result in oversmoothing of the output. Therefore, similar to the work done in the image domain [111, 159–161], we propose to use iterative regularization with the minimization in (8.7) carried out multiple times, whereby in each iteration the regularization parameter μ is decreased in a dyadic way. This produces enhanced and feature preserving point clouds as shown in the results.

8.4 Experiments and Results

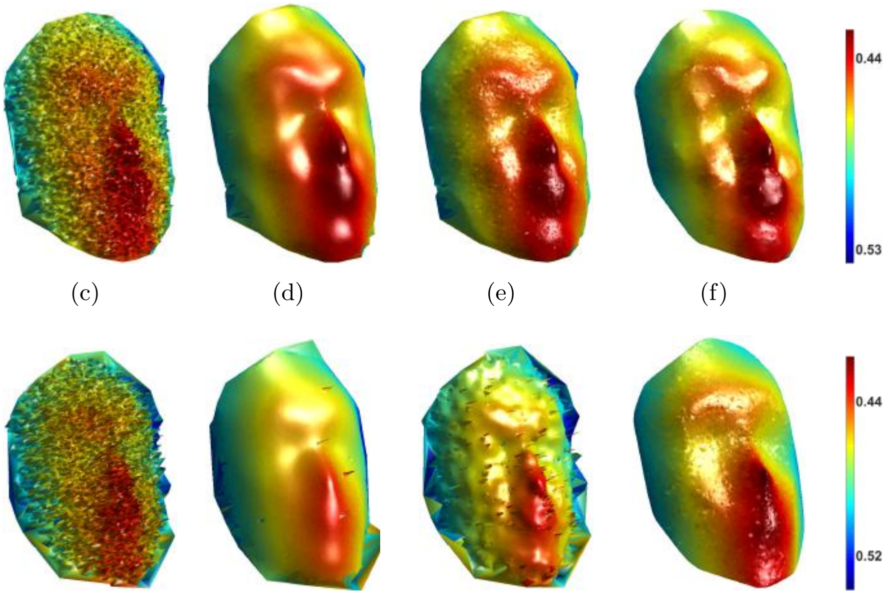
In this section we present the results of the quantitative and qualitative analysis of performance of the proposed 3D BTV regularization using both simulated and real data.

We start by presenting a performance comparison of the proposed algorithm with the state-of-art methods. For this purpose, we simulate a 3D camera acquisition of a human face obtained from the "Facecap" dataset [107]. We call this acquisition, containing 13710 3D points, the ground truth (GT) and it is shown in Figure 8.2(a). We add independent Gaussian noise in each coordinate of the 3D points in the GT with zero mean and standard deviations, i.e., σ of $2.5mm$ and $5mm$, respectively. The noisy 3D data is shown in Figure 8.2(c) and Figure 8.2(h), respectively. The noisy point clouds are then given as input to the state-of-art filtering algorithms namely, BMD and MLS, and also to the proposed 3D BTV regularization. For the sake of fair comparison, the number of iterations for each method are selected such that they either achieve a smooth reconstruction and/or the processing time is comparative to other methods.

The resulting filtered point clouds are plotted in Figure 8.2. A qualitative analysis of the results show that although BMD is able to remove most of the noisy artifacts from the input point clouds, the resulting point clouds suffer from loss of valuable local and global feature information due to over-smoothing, as shown in Figure 8.2(d) and Figure 8.2(i), respectively. The effects of over-smoothing become worse as the noise level increases. The results of MLS, on the other hand, show its inability to tackle the noisy artifacts in the data. This behavior becomes severe as the noise in the data increases, as shown in Figure 8.2(e) and Figure 8.2(j), respectively. In comparison, the proposed method is able to successfully remove noise from the data and does not suffer from the problem of over-smoothing. This method is able to preserve salient facial features in the output point clouds, even at higher noise levels, as shown in Figure 8.2(f) and Figure 8.2(k), respectively. This is due to using the combination of the TV regularization, which is global in nature, and elements of the BMD framework which exploit the local shape and geometric properties of point patches. Moreover, the use of TV regularization equips the proposed method to successfully preserve the edge information as compared to the state-of-art methods, as shown in the results in Figure 8.2.



(a)



(c) (d) (e) (f)

(h) (i) (j) (k)

FIGURE 8.2: 3D plotting of the simulated frame from the "Facecap" dataset. The GT data is plotted in the **First row**. This is followed by the plots of results of the state-of-art methods and the proposed technique, which are used to filter the GT data affected by zeros mean independent Gaussian noise with the standard deviations, i.e., σ of **Second row:** $2.5mm$ and **Third row:** $5mm$, respectively. Each row contains, from left to right, the noisy input point cloud, the result of BMD, the result of MLS, and the result of the proposed 3D BTV regularization. Display color-scale is based on the depth values of the 3D points and is in the units of meters.

For quantitative analysis, we compute the RMSE of the filtered point cloud resulting from each method with respect to the GT. A comparison of the RMSE, and the processing times, for each method is shown in Table 8.1. BMD is the most efficient of all three methods but is also the less accurate due to loss of valuable feature information as explained before. On the other hand, MLS consumes more processing time but is unable to tackle artifacts due to high magnitude of noise. The proposed method consumes approximately the same amount of processing time as MLS and is the most accurate of all the tested methods. It is able to successfully filter the data affected by varying amount of noise.

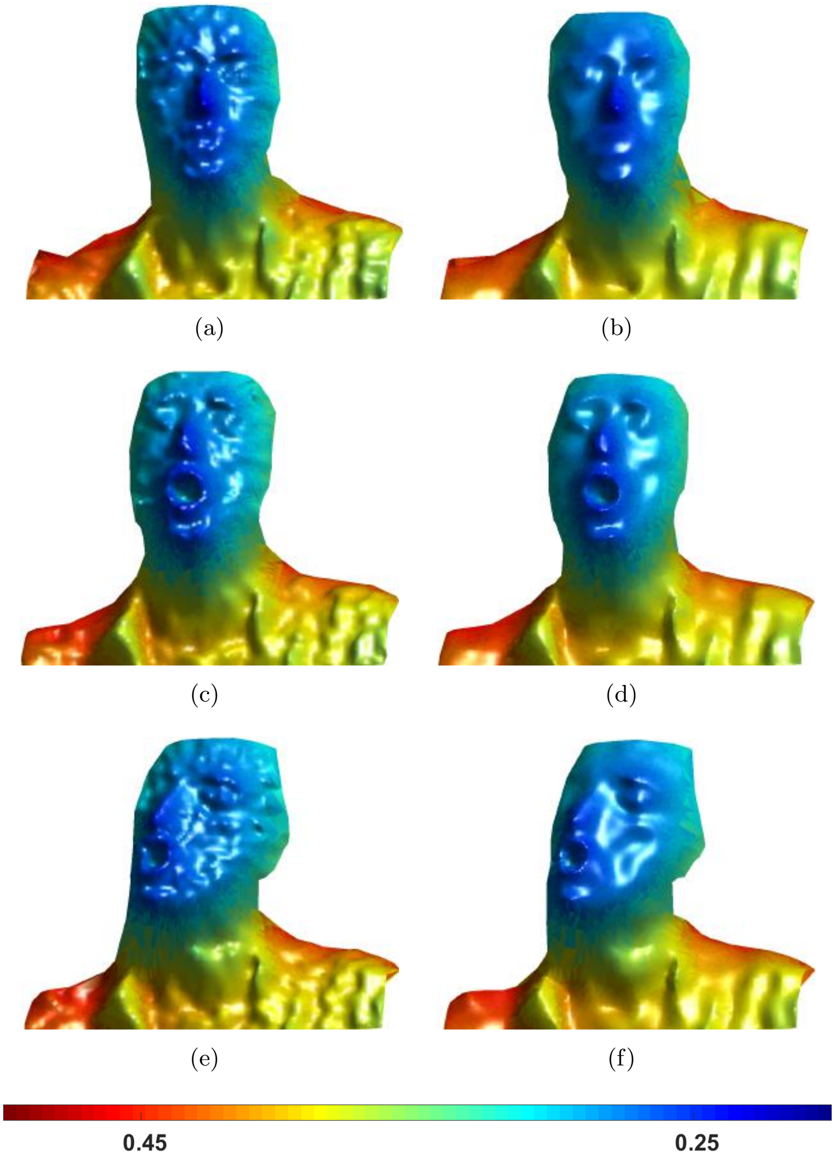
Table 8.1: 3D RMSE in mm/processing time in sec for the results obtained from the state-of-art methods, namely BMD and MLS, as well as the proposed method, i.e., 3DBTV on a single simulated 3D frame obtained from the "Facecap" dataset. Independent Gaussian noise is added in each coordinate of the 3D points with zero mean and standard deviations, i.e, σ of 2.5mm and 5mm, respectively.

	$\sigma = 2.5mm$	$\sigma = 5mm$
BMD	1.26mm/10sec	2.42mm/23sec
MLS	1.10mm/47sec	2.22mm/117sec
3DBTV	**0.98mm**/46sec	**1.63mm**/114sec

In the next step, we analyze the qualitative performance of the proposed algorithm using real data. For this purpose, we use a video sequence containing a non-rigidly deforming human face acquired with a PMD camboard nano ToF camera [20, 111]. We select 3 frames from this sequence, as shown in Figure 8.3(a), Figure 8.3(c) and Figure 8.3(e), respectively. The PMD camera has a resolution of 120 × 160 and, each selected frame contains approximately 4400 valid 3D points. We run the proposed 3D BTV regularization on each of the selected 3 frames. The processing time per frame is 4.5sec, approximately. The resulting 3D frames or point clouds are plotted in Figure 8.3(b), Figure 8.3(d) andFigure 8.3(f), respectively. These results show the ability of the proposed algorithm to filter out noise from the camera acquisitions. Moreover, the results show that the proposed algorithm is able to preserve features such as nose, mouth, collar etc., and produce globally smooth 3D reconstructions.

8.5 Conclusion

In this chapter, we have presented a framework for denoising unorganized 3D point clouds via 3D BTV regularization. BTV regularization has been shown to perform well for noise removal and feature preservation in 2D color and depth images. The computation of BTV of image pixels is made easier by the data organization on the image grid, which provides local structure and neighborhood information. In the case of unorganized 3D point clouds, computation of BTV of 3D points becomes a challenging task as no information about local structure or neighborhood is available. We propose to use the elements from BMD framework which exploit the shape and geometric properties in local point patches to compute the 3D BTV. We perform experiments on both simulated and real data and show that the proposed method is more accurate as compare to the state-of-art local filtering methods, and is able to produce globally smooth and feature preserving 3D reconstructions.



(a) (b)

(c) (d)

(e) (f)

Figure 8.3: Results of applying the proposed technique on real data acquired with a PMD camboard nano ToF camera [20]. **First column** contains a 3D plotting of the 3 selected frames from the data acquired with the camera. **Second column** contains a 3D plotting of the corresponding filtered frames obtained via the proposed 3D BTV regularization. Display color-scale is based on the depth values of the 3D points and is in the units of meters.

Chapter 9

Conclusions and Future Work

Recently available 3D sensing technologies in commodity RGB-D or 3D cameras have enabled us to acquire 3D reconstructions of the world in an affordable and flexible manner. With affordability and flexibility, come the limitations of limited field-of-view (FOV), high magnitude of noise and limited resolution which affect the 3D measurements acquired with these cameras. These limitations inhibit the usage of commodity cameras in applications which require accurate, detailed and full 3D reconstructions of the environment. In this thesis, our goal has been to extend state-of-art via research and development of algorithms which overcome the above mentioned limitations of commodity RGB-D or 3D cameras to build accurate, feature preserving and full 360° 3D reconstructions of the environment instantaneously. These algorithms are able to reconstruct dynamic scenes containing rigid as well as non-rigid objects undergoing relatively large local motions.

Indeed, a single commodity RGB-D camera can only capture partial views of a dynamic scene and hence fails to provide full 360° 3D reconstructions instantaneously. For this purpose, multiple cameras with overlapping FOVs can be used but the partial 3D reconstructions obtained from them cannot be accurately aligned unless the knowledge of their relative poses is available. Estimation of relative camera poses is carried out by a process known as extrinsic calibration. State-of-art methods for extrinsic calibration of RGB-D multi-view systems use classical calibration algorithms which make use of 2D photometric and 3D geometric information, separately. A technique tailored for RGB-D cameras based multi-view systems was missing. Such technique should use both types of available complementary information to achieve more accurate results. For this purpose we have proposed BAICP+ which combines two sate-of-art algorithms namely Bundle Adjustment (BA) [41], which makes use of 2D photometric information, and Iterative Closest Point (ICP) [44] algorithm, which makes use of 3D geometric information in a single weighted bi-objective optimization. By manually varying the weight which decides the relative importance given to either 2D or 3D information, we have shown that the proposed framework achieves more accurate result as compared to state-of-art methods using 2D and 3D information separately.

Building upon BAICP+, in the next step we have analytically analyzed the relationship of the weight with noise present in the 2D and 3D measurements. As a result, we have proposed a completely automated weighted bi-objective optimization scheme which optimally combines both sources of information to achieve accurate results. In the absence of parameters of noise model affecting the 2D and 3D measurements, we have proposed an iterative method to estimate these parameters together with relative camera poses in parallel. In our experiments, we have shown improved calibration performance as compared to state-of-art methods on both simulated and real data. In this work we have considered a noise model which assumes independent and identically distributed noise in both 2D and 3D measurements. Although, this method has shown to perform well on real data but the noise affecting specifically the 3D measurements is in reality more complex and depends on several factors as explained in Section 2.1.2. Therefore, as future work we would like to look into incorporating these factors and more complex noise models in our framework.

In the second part of this thesis, we have targeted the other two limitations of commodity RGB-D or 3D cameras, namely high magnitude of measurement noise and limited resolution. These limitation prevents these cameras from acquiring accurate 3D reconstructions with complex and fine-scale features. Our focus has been on online methods which recursively fuse and filter the acquired information to improve its resolution and quality. The state-of-art recursive 3D data fusion and filtering algorithms, built around commodity 3D sensing technologies, such as KinectFusion are restricted to reconstructing dynamic scenes containing rigid objects undergoing global deformations only [1]. Therefore we have proposed KinectDeform, which extends KinectFusion to target enhanced 3D reconstruction of scenes containing non-rigid objects undergoing local deformations. KinectDeform uses mono-view systems and combines an efficient non-rigid registration algorithm with a view-dependent implicit TSDF based surface representation. Experiments showed that KinectDeform is able to produce noise-free 3D reconstructions of non-rigid objects and can handle large local deformations.

Moving beyond KinectDeform and towards handling data from multi-view systems, we have proposed an algorithm called VI-KinectDeform. VI-KinectDeform simplifies the KinectDeform pipeline by replacing its view-dependent implicit TSDF based surface representation, which required an expensive data reorganization step at every iteration, with a view-independent explicit MLS based surface representation. Comparative experimental evaluation of VI-KinectDeform and KinectDeform also showed VI-KinectDeform's improved performance in terms of both complexity and accuracy. Although both KinectDeform and VI-KinectDeform produce noise-free results and VI-KinectDeform is easily extendable to multi-view systems, they can only handle HR data and might not work well with LR data. Therefore, we have proposed a multi-view framework based on LR data acquired via commodity cameras. The proposed framework filters out noise and enhances the resolution of data to recover and preserve features and, produce full 360° 3D reconstructions of non-rigid objects. This framework makes use of a novel recursive dynamic multi-frame 3D super-resolution algorithm to track and filter the 3D motion and position of every point. To recover global smoothness property of 3D data after per-point tracking and to remove system blur, we have also proposed a novel 3D bilateral total variation (BTV) regularization. The proposed 3D BTV regularization exploits surface properties of local point patches to construct a surface gradient operator for the computation of the regularization term. A detailed qualitative and quantitative evaluation of the proposed framework shows it to be able to handle highly noisy and LR data to produce accurate, smooth, feature preserving and full 3D reconstruction of dynamic scenes. The per-point tracking algorithm used in the proposed framework allows for handling large local motions but it uses a constant velocity model which limits the ability to track abrupt changes in motion of 3D points. Therefore, as future work it would be interesting to incorporate a more complex motion model such as the constant acceleration model in the proposed framework.

Appendix A

Non-linear Optimization for Proposed Bi-Objective Framework

Due to the non-linear dependence of cost function in (4.5) on parameters in $\mathbf{S}$, the MLE $\hat{\mathbf{S}}$ is to be computed via a numerical scheme based on non-linear optimization. In this scheme at every iteration a small change is introduced in the current set of parameters leading to comparatively improved performance or lower residual [86]. First step in this scheme is to linearize $\mathbf{b}^j_{l,m}(\mathbf{T}_l, \mathbf{T}_m)$ and $\mathbf{a}^h_l(\mathbf{S}^h_l)$ about current estimate $\hat{\mathbf{S}}$ assuming very small error $\Delta\mathbf{S}$ using Taylor expansion to get:

$$\mathbf{b}^j_{l,m}(\mathbf{T}_l, \mathbf{T}_m) \approx \mathbf{b}^j_{l,m}(\hat{\mathbf{T}}_l, \hat{\mathbf{T}}_m) + \mathbf{J}_{\mathbf{b}^j_{l,m}}\Delta\mathbf{S}, \tag{A.1}$$

and:

$$\mathbf{a}^h_l(\mathbf{S}^h_l) \approx \mathbf{a}^h_l(\hat{\mathbf{S}}^h_l) + \mathbf{J}_{\mathbf{a}^h_l}\Delta\mathbf{S}, \tag{A.2}$$

where $\mathbf{J}_{\mathbf{b}^j_{l,m}}$ and $\mathbf{J}_{\mathbf{a}^h_l}$ are Jacobians of $\mathbf{b}^j_{l,m}(\mathbf{T}_l, \mathbf{T}_m)$ and $\mathbf{a}^h_l(\mathbf{S}^h_l)$, with respect to $\mathbf{S}$, respectively. Replacing (A.1) and (A.2) in (4.5) and concatenating $\mathbf{b}^j_{l,m}$, $\mathbf{a}^h_l$, $\Delta\mathbf{S}$ and corresponding Jacobians we have:

$$\begin{aligned} V(\mathbf{S}) \approx\ & (\mathbf{B} + \mathbf{J}_\mathbf{B}\boldsymbol{\Delta}\mathbf{S})^T(\mathbf{B} + \mathbf{J}_\mathbf{B}\boldsymbol{\Delta}\mathbf{S}) + \\ & w(\mathbf{A} + \mathbf{J}_\mathbf{A}\boldsymbol{\Delta}\mathbf{S})^T(\mathbf{A} + \mathbf{J}_\mathbf{A}\boldsymbol{\Delta}\mathbf{S}) \\ =\ & (\mathbf{B}^T\mathbf{B} + 2\Delta\mathbf{S}^T\mathbf{J}^T_\mathbf{B}\mathbf{B} + \Delta\mathbf{S}^T\mathbf{J}^T_\mathbf{B}\mathbf{J}_\mathbf{B}\Delta\mathbf{S}) \\ & + w(\mathbf{A}^T\mathbf{A} + 2\Delta\mathbf{S}^T\mathbf{J}^T_\mathbf{A}\mathbf{A} + \Delta\mathbf{S}^T\mathbf{J}^T_\mathbf{A}\mathbf{J}_\mathbf{A}\Delta\mathbf{S}). \end{aligned} \tag{A.3}$$

After that we take the derivative of $V(\mathbf{S})$ with respect to $\mathbf{S}$ and equate it to zero to get:

$$\frac{\partial V(\mathbf{S})}{\partial \mathbf{S}} \approx \mathbf{J}_{\mathbf{B}}^{T}\mathbf{B} + \mathbf{J}_{\mathbf{B}}^{T}\mathbf{J}_{\mathbf{B}}\mathbf{\Delta S} + w\mathbf{J}_{\mathbf{A}}^{T}\mathbf{A} + w\mathbf{J}_{\mathbf{A}}^{T}\mathbf{J}_{\mathbf{A}}\mathbf{\Delta S} = 0. \tag{A.4}$$

Rearranging (A.4), we get the parameter update rule as:

$$\mathbf{\Delta S} = -(\mathbf{J}_{\mathbf{B}}^{T}\mathbf{J}_{\mathbf{B}} + w\mathbf{J}_{\mathbf{A}}^{T}\mathbf{J}_{\mathbf{A}})^{-1}(\mathbf{J}_{\mathbf{B}}^{T}\mathbf{B} + w\mathbf{J}_{\mathbf{A}}^{T}\mathbf{A}). \tag{A.5}$$

We can also rearrange (A.5) according to Levenberg-Marquardt LM [91] algorithm get the parameter update rule as:

$$\begin{split}((\frac{1}{2\sigma_{3D}^{2}}\mathbf{J}_{\mathbf{B}}^{T}\mathbf{J}_{\mathbf{B}} + \frac{1}{\sigma_{2D}^{2}}\mathbf{J}_{\mathbf{A}}^{T}\mathbf{J}_{\mathbf{A}})+ \\ \lambda\mathtt{diag}(\frac{1}{2\sigma_{3D}^{2}}\mathbf{J}_{\mathbf{B}}^{T}\mathbf{J}_{\mathbf{B}} + \frac{1}{\sigma_{2D}^{2}}\mathbf{J}_{\mathbf{A}}^{T}\mathbf{J}_{\mathbf{A}}))\mathbf{\Delta S} \\ = -(\frac{1}{2\sigma_{3D}^{2}}\mathbf{J}_{\mathbf{B}}^{T}\mathbf{B} + \frac{1}{\sigma_{2D}^{2}}\mathbf{J}_{\mathbf{A}}^{T}\mathbf{A}),\end{split} \tag{A.6}$$

where λ is the damping factor.

Bibliography

[1] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-2183-0. doi: 10.1109/ISMAR.2011.6092378. URL `http://dx.doi.org/10.1109/ISMAR.2011.6092378`.

[2] Bernhard Kainz, Stefan Hauswiesner, Gerhard Reitmayr, Markus Steinberger, Raphael Grasset, Lukas Gruber, Eduardo Veas, Denis Kalkofen, Hartmut Seichter, and Dieter Schmalstieg. OmniKinect: real-time dense volumetric data acquisition and applications. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, VRST '12, pages 25–32, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1469-5. doi: 10.1145/2407336.2407342. URL `http://doi.acm.org/10.1145/2407336.2407342`.

[3] Mingsong Dou and H. Fuchs. Temporally enhanced 3D capture of room-sized dynamic scenes with commodity depth cameras. In *Virtual Reality (VR), 2014 iEEE*, pages 39–44, March 2014. doi: 10.1109/VR.2014.6802048.

[4] J. Sturm, E. Bylow, F. Kahl, and D. Cremers. CopyMe3D: Scanning and printing persons in 3D. In *German Conference on Pattern Recognition (GCPR)*, Saarbrücken, Germany, September 2013.

[5] A. Maimone and H. Fuchs. Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 137–146, Oct. doi: 10.1109/ISMAR.2011.6092379.

[6] C. Kuster, T. Popa, C. Zach, C. Gotsman, and M. Gross. FreeCam: A Hybrid Camera System for Interactive Free-Viewpoint Video. In *Proceedings of Vision, Modeling, and Visualization (VMV)*, 2011.

[7] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research (IJRR)*, 31(5):647–663, April 2012.

[8] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 28(5), December 2009.

[9] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Probabilistic Deformable Surface Tracking From Multiple Videos. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV 2010 - 11th European Conference on Computer Vision*, volume 6314, pages 326–339, Heraklion, Greece, September 2010. Springer. doi: 10.1007/978-3-642-15561-1_24. URL https://hal.inria.fr/inria-00568912.

[10] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27(3):97:1–97:9, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360696. URL http://doi.acm.org/10.1145/1360612.1360696.

[11] Christian Theobalt, Naveed Ahmed, Hendrik Lensch, Marcus Magnor, and Hans-Peter Seidel. Seeing people in different light-joint shape, motion, and reflectance capture. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):663–674, 2007. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.1006.

[12] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph.*, 22(3):569–577, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882309. URL http://doi.acm.org/10.1145/882262.882309.

[13] Yasutaka Furukawa and Jean Ponce. Dense 3d motion capture from synchronized video streams. In *Image and Geometry Processing for 3-D Cinematography*, pages 193–211. 2010. doi: 10.1007/978-3-642-12392-4_9. URL http://dx.doi.org/10.1007/978-3-642-12392-4_9.

[14] Rui Yu, Chris Russell, Neill D. F. Campbell, and Lourdes Agapito. Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from RGB video. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 918–926, 2015. doi: 10.1109/ICCV.2015.111. URL http://dx.doi.org/10.1109/ICCV.2015.111.

[15] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27(3):97:1–97:9, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360696. URL `http://doi.acm.org/10.1145/1360612.1360696`.

[16] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):98:1–98:10, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360697. URL `http://doi.acm.org/10.1145/1360612.1360697`.

[17] Edilson de Aguiar, Christian Theobalt, Carsten Stoll, and Hans-Peter Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages XX–XX, Minneapolis, USA, June 2007. IEEE, IEEE.

[18] Kinect V2. URL `https://dev.windows.com/en-us/kinect//`.

[19] Xtion PRO LIVE . URL `https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/`.

[20] PMD Technologies. Camboard Nano, 2012. URL `http://www.pmdtec.com`.

[21] Jan Smisek, Michal Jancosek, and Tomás Pajdla. 3D with Kinect. In *ICCV Workshops*, pages 1154–1160, 2011.

[22] P. Palasek, Heng Yang, Zongyi Xu, N. Hajimirza, E. Izquierdo, and I. Patras. A flexible calibration method of multiple Kinects for 3D human reconstruction. In *Multimedia Expo Workshops (ICMEW), 2015 IEEE International Conference on*, pages 1–4, June 2015. doi: 10.1109/ICMEW.2015.7169829.

[23] Kai Berger, Kai Ruhl, Christian Brümmer, Yannic Schröder, Alexander Scholz, and Marcus Magnor. Markerless Motion Capture using multiple Color-Depth Sensors. In *Proc. Vision, Modeling and Visualization (VMV) 2011*, pages 317–324, October 2011.

[24] K. Berger, K. Ruhl, M. Albers, Y. Schroder, A. Scholz, J. Kokemuller, S. Guthe, and M. Magnor. The capturing of turbulent gas flows using multiple Kinects. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1108–1113, Nov 2011. doi: 10.1109/ICCVW.2011.6130374.

[25] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[26] B. Penelle, A. Schenkel, and N. Warzee. Geometrical 3D reconstruction using real-time RGB-D cameras. In *3D Imaging (IC3D), 2011 International Conference on*, pages 1–8, Dec 2011. doi: 10.1109/IC3D.2011.6584368.

[27] R.S. Yang, Yuk Hin Chan, Rui Gong, Minh Nguyen, A.G. Strozzi, P. Delmas, G. Gimel'farb, and R. Ababou. Multi-Kinect scene reconstruction: Calibration and depth inconsistencies. In *Image and Vision Computing New Zealand (IVCNZ), 2013 28th International Conference of*, pages 47–52, Nov 2013. doi: 10.1109/IVCNZ.2013.6726991.

[28] Oisin Mac Aodha, Neill D.F. Campbell, Arun Nair, and Gabriel J. Brostow. Patch Based Synthesis for Single Depth Image Super-Resolution. In *ECCV (3)*, pages 71–84, 2012.

[29] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. Real-time Non-rigid Reconstruction using an RGB-D Camera. *ACM Transactions on Graphics (TOG)*, 2014.

[30] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust Single-view Geometry and Motion Reconstruction. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 175:1–175:10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-858-2. doi: 10.1145/1661412.1618521. URL http://doi.acm.org/10.1145/1661412.1618521.

[31] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. LidarBoost: Depth Superresolution for ToF 3D Shape Scanning. *In Proc. of IEEE CVPR 2009*, 2009.

[32] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers. Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions. In *Robotics: Science and Systems Conference (RSS)*, June 2013.

[33] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D Reconstruction at Scale using Voxel Hashing. *ACM Transactions on Graphics (TOG)*, 2013.

[34] Cerqueira Márcio, A. L. Apolinario Jr., and A. C. S. Souza. KinectFusion for Faces: Real-Time 3D Face Tracking and Modeling Using a Kinect Camera for a Markerless AR System. *SBC Journal on 3D Interactive Systems*, 4:2–7, 2013.

[35] Yan Cui, Will Chang, Tobias Nöll, and Didier Stricker. KinectAvatar: Fully Automatic Body Capture Using a single Kinect. In *ACCV Workshop on Color Depth fusion in computer*. ACCV, 2012.

[36] J. Sturm, E. Bylow, F. Kahl, and D. Cremers. CopyMe3D: Scanning and printing persons in 3D. In *German Conference on Pattern Recognition (GCPR)*, Saarbrücken, Germany, September 2013.

[37] D.S. Alexiadis, G. Kordelas, K.C. Apostolakis, J.D. Agapito, J.M. Vegas, E. Izquierdo, and P. Daras. Reconstruction for 3D immersive virtual environments. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2012 13th International Workshop on*, pages 1–4, May 2012. doi: 10.1109/WIAMIS.2012.6226760.

[38] S. Miller, A. Teichman, and S. Thrun. Unsupervised extrinsic calibration of depth sensors in dynamic scenes. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2695–2702, Nov 2013. doi: 10.1109/IROS.2013.6696737.

[39] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673 vol.1, 1999. doi: 10.1109/ICCV.1999.791289.

[40] J. Bougouet. http://www.vision.caltech.edu/bouguetj/calibdoc/, 2007.

[41] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle Adjustment – A Modern Synthesis. In *VISION ALGORITHMS: THEORY AND PRACTICE, LNCS*, pages 298–375. Springer Verlag, 2000.

[42] K. Amplianitis, M. Adduci, and R. Reulke. Calibration of a Multiple Stereo and Rgb-D Camera System for 3d Human Tracking. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, (1):7–14, March 2014. doi: 10.5194/isprsarchives-XL-3-W1-7-2014.

[43] M. Nakazawa, I. Mitsugami, Y. Makihara, H. Nakajima, H. Habe, H. Yamazoe, and Y. Yagi. Dynamic scene reconstruction using asynchronous multiple Kinects. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 469–472, Nov 2012.

[44] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992. ISSN 0162-8828. doi: 10.1109/34.121791. URL http://dx.doi.org/10.1109/34.121791.

[45] Mingsong Dou, Li Guan, Jan-Michael Frahm, and Henry Fuchs. Exploring High-Level Plane Primitives for Indoor 3D Reconstruction with a Hand-held RGB-D Camera. In *Computer Vision - ACCV 2012 Workshops, ACCV 2012 International Workshops, Daejeon, Korea, November 5-6, 2012, Revised Selected Papers, Part II*, pages 94–108, 2012. doi: 10.1007/978-3-642-37484-5_9. URL http://dx.doi.org/10.1007/978-3-642-37484-5_9.

[46] T. Tykkala, C. Audras, and A.I. Comport. Direct Iterative Closest Point for real-time visual odometry. In *Computer Vision Workshops (ICCV Workshops), 2011*

IEEE International Conference on, pages 2050–2056, Nov 2011. doi: 10.1109/ICCVW.2011.6130500.

[47] Henry Roth and Marsette Vona. Moving Volume KinectFusion. In *Proceedings of the British Machine Vision Conference*, pages 112.1–112.11. BMVA Press, 2012. ISBN 1-901725-46-4. doi: http://dx.doi.org/10.5244/C.26.112.

[48] T. Whelan, M. Kaess, M.F. Fallon, H. Johannsson, J.J. Leonard, and J. McDonald. Kintinuous: Spatially Extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012.

[49] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3D Full Human Bodies Using Kinects. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4):643–650, April 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.56.

[50] Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T. Barron, and Gleb Gusev. 3d self-portraits. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2013)*, 32(6), November 2013.

[51] Ming Zeng, Jiaxiang Zheng, Xuan Cheng, and Xinguo Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[52] A. Weiss, D. Hirshberg, and M. J. Black. Home 3d body scans from noisy image and range data. In *2011 International Conference on Computer Vision*, pages 1951–1958, Nov 2011. doi: 10.1109/ICCV.2011.6126465.

[53] Niloy J. Mitra, Simon Flory, Maks Ovsjanikov, Natasha Gelfand, Leonidas Guibas, and Helmut Pottmann. Dynamic geometry registration. In *Symposium on Geometry Processing*, pages 173–182, 2007.

[54] Jochen Sussmuth, Marco Winter, and Gunther Greiner. Reconstructing animated meshes from time-varying point clouds. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1469–1476, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. URL `http://dl.acm.org/citation.cfm?id=1731309.1731332`.

[55] Michael Wand, Philipp Jenke, Qixing Huang, Martin Bokeloh, Leonidas Guibas, and Andreas Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 49–58, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. ISBN 978-3-905673-46-3. URL `http://dl.acm.org/citation.cfm?id=1281991.1281998`.

[56] Michael Wand, Bart Adams, Maksim Ovsjanikov, Alexander Berner, Martin Bokeloh, Philipp Jenke, Leonidas Guibas, Hans-Peter Seidel, and Andreas Schilling. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph.*, 28(2):15:1–15:15, May 2009. ISSN 0730-0301. doi: 10.1145/1516522.1516526. URL http://doi.acm.org/10.1145/1516522.1516526.

[57] Andrei Sharf, Dan A. Alcantara, Thomas Lewiner, Chen Greif, Alla Sheffer, Nina Amenta, and Daniel Cohen-Or. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph.*, 27(5):110:1–110:10, December 2008. ISSN 0730-0301. doi: 10.1145/1409060.1409063. URL http://doi.acm.org/10.1145/1409060.1409063.

[58] Weipeng Xu, Mathieu Salzmann, Yongtian Wang, and Yue Liu. Deformable 3d fusion: From partial dynamic 3d observations to complete 4d models. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 2183–2191, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.252. URL http://dx.doi.org/10.1109/ICCV.2015.252.

[59] Andreas Nüchter, Jan Elseberg, Peter Schneider, and Dietrich Paulus. Study of parameterizations for the rigid body transformations of the scan registration problem. *Computer Vision and Image Understanding*, 114(8):963–980, 2010.

[60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D): 35–45, 1960.

[61] Hao Li, Bart Adams, Leonidas J. Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 175:1–175:10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-858-2. doi: 10.1145/1661412.1618521. URL http://doi.acm.org/10.1145/1661412.1618521.

[62] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[63] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer Vision and Image Understanding*, 139:1 – 20, 2015. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2015.05.006. URL http://www.sciencedirect.com/science/article/pii/S1077314215001071.

[64] F. Brunet. *Contributions to Parametric Image Registration and 3D Surface Reconstruction*. PhD thesis, Université d'Auvergne, Technische Universität Müunchen, 2010.

[65] Kourosh Khoshelham and Er Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. In *Sensors 2012, 12, 1437–1454. 2013*, page 8238.

[66] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Horaud. *Time of Flight Cameras: Principles, Methods, and Applications*. SpringerBriefs in Computer Science. Springer, October 2012. doi: 10.1007/978-1-4471-4658-2. URL https://hal.inria.fr/hal-00725654.

[67] Otmar Hilliges Dave Molyneaux Steve Hodges David Kim Alex Butler, Shahram Izadi. Shake'n'sense: reducing interference for overlapping structured light depth cameras. January 2012. URL https://www.microsoft.com/en-us/research/publication/shakensense-reducing-interference-for-overlapping-structured-light-depth-cameras

[68] Y. M. Kim, D. Chan, Christian Theobalt, and S. Thrun. Design and calibration of a multi-view tof sensor fusion system. In *IEEE CVPR Workshop on Time-of-flight Computer Vision*, pages 1–7, Anchorage, USA, 2008. IEEE.

[69] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9. URL http://dl.acm.org/citation.cfm?id=938978.939190.

[70] Szymon Rusinkiewicz, Olaf A. Hall-Holt, and Marc Levoy. Real-time 3D model acquisition. In *SIGGRAPH*, pages 438–446, 2002.

[71] Brian Curless and Marc Levoy. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237269. URL http://doi.acm.org/10.1145/237170.237269.

[72] Steven Parker, Peter Shirley, Yarden Livnat, Charles Hansen, and Peter-Pike Sloan. Interactive Ray Tracing for Isosurface Rendering. In *Proceedings of the Conference on Visualization '98*, VIS '98, pages 233–238, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. ISBN 1-58113-106-2. URL http://dl.acm.org/citation.cfm?id=288216.288266.

[73] PrimeSense http://www.primesense.com/. URL http://www.primesense.com/.

[74] R. Furuakwa, K. Inose, and H. Kawasaki. Multi-view reconstruction for projector camera systems based on bundle adjustment. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 69–76, 2009. doi: 10.1109/CVPRW.2009.5204318.

[75] Andrew Johnson and Sing Bing Kang. Registration and Integration of Textured 3-D Data. In *InternationalConference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, pages 234 – 241, May 1997.

[76] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the RGB-D SLAM system. In *ICRA*, pages 1691–1696. IEEE, 2012. ISBN 978-1-4673-1403-9. URL http://dblp.uni-trier.de/db/conf/icra/icra2012.html#EndresHESCB12.

[77] Dominik Neumann, Felix Lugauer, Sebastian Bauer, Jakob Wasza, and Joachim Hornegger. Real-time RGB-D mapping and 3-D modeling on the GPU using the random ball cover data structure. In *ICCV Workshops*, pages 1161–1167. IEEE, 2011. ISBN 978-1-4673-0062-9. URL http://dblp.uni-trier.de/db/conf/iccvw/iccvw2011.html#NeumannLBWH11.

[78] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers. Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions. In *Robotics: Science and Systems Conference (RSS)*, June 2013.

[79] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.

[80] V-REP. URL http://www.coppeliarobotics.com/.

[81] Jan Smisek, Michal Jancosek, and Tomás Pajdla. 3D with Kinect. In *ICCV Workshops*, pages 1154–1160. IEEE, 2011. ISBN 978-1-4673-0062-9. URL http://dblp.uni-trier.de/db/conf/iccvw/iccvw2011.html#SmisekJP11.

[82] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *International Conference on Robotics and Automation*, Shanghai, China, 2011 2011.

[83] J Michot, A Bartoli, and F Gaspard. Bi-Objective Bundle Adjustment With Application to Multi-Sensor SLAM. In *3DPVT'10 – Int'l Symp. on 3D Data Processing, Visualization and Transmission*, Paris, France, may 2010.

[84] H. Afzal, D. Aouada, D. Fofi, B. Mirbach, and B. Ottersten. RGB-D Multi-view System Calibration for Full 3D Scene Reconstruction. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2459–2464, Aug 2014. doi: 10.1109/ICPR.2014.425.

[85] Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, New York, NY, USA, 1st edition, 2012. ISBN 1107011795, 9781107011793.

[86] I. J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100, 2003.

[87] V-REP `http://www.coppeliarobotics.com/`. URL `http://www.primesense.com/`.

[88] Qilong Zhang. Extrinsic calibration of a camera and laser range finder. In *In IEEE International Conference on Intelligent Robots and Systems (IROS*, page 2004, 2004.

[89] Péter Fankhauser, Michael Bloesch, Diego Rodriguez, , Ralf Kaestner, Marco Hutter, and Roland Siegwart. Kinect v2 for Mobile Robot Navigation: Evaluation and Modeling. In *IEEE International Conference on Advanced Robotics (ICAR)*, 2015.

[90] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[91] Jorge Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, chapter 10, pages 105–116–116. Springer Berlin / Heidelberg, 1978. ISBN 978-3-540-08538-6. doi: 10.1007/bfb0067700. URL `http://dx.doi.org/10.1007/bfb0067700`.

[92] Vincent Rabaud. Vincent's Structure from Motion Toolbox. `http://github.com/vrabaud/sfm_toolbox`.

[93] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 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*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047270. URL `http://doi.acm.org/10.1145/2047196.2047270`.

[94] Frank Steinbrücker, Christian Kerl, and Daniel Cremers. Large-Scale Multi-resolution Surface Reconstruction from RGB-D Sequences. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 3264–3271, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.405. URL `http://dx.doi.org/10.1109/ICCV.2013.405`.

[95] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu. Octree-based fusion for realtime 3d reconstruction. *Graphical Models*, 75(3):126 – 136, 2013. ISSN 1524-0703. doi: http://dx.doi.org/10.1016/j.gmod.2012.09.002. Computational Visual Media Conference 2012.

[96] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A Kolb. Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion. In *3D Vision - 3DV 2013, 2013 International Conference on*, pages 1–8, June 2013. doi: 10.1109/3DV.2013.9.

[97] François Destelle, Céline Roudet, Marc Neveu, and Albert Dipanda. Towards a real-time tracking of dense point-sampled geometry. *International Conference on Image Processing*, pages 381–384, 2012.

[98] Tali Basha, Yael Moses, and Nahum Kiryati. Multi-view Scene Flow Estimation: A View Centered Variational Approach. In *International Journal of Computer Vision*, pages 1–16, 2011.

[99] Jan Cech, Jordi Sanchez-Riera, and Radu P. Horaud. Scene Flow Estimation by Growing Correspondence Seeds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011. URL `http://perception.inrialpes.fr/Publications/2011/CSH11`.

[100] Simon Hadfield and Richard Bowden. Kinecting the dots: Particle Based Scene Flow From Depth Sensors. In *Proceedings, International Conference on Computer Vision*, pages 2290 – 2295, Barcelona, Spain, 6-13 Nov 2011. doi: 10.1109/ICCV.2011.6126509. URL `http://personal.ee.surrey.ac.uk/Personal/S.Hadfield/papers/Kinecting%20the%20dots%20Particle%20Based%20Scene%20Flow%20From%20Depth%20Sensors.pdf`.

[101] Hao Li, Robert W. Sumner, and Mark Pauly. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. *Computer Graphics Forum (Proc. SGP'08)*, 27(5), July 2008.

[102] Zeng, Ming and Zheng, Jiaxiang and Cheng, Xuan and Liu, Xinguo. Templateless quasi-rigid shape modeling with implicit loop-closure. In *CVPR*, pages 145–152.

IEEE, 2013. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#ZengZCL13.

[103] Frédéric Cazals and Joachim Giesen. Delaunay triangulation based surface reconstruction: Ideas and algorithms. In *Effective Computational Geometry for Curves and Surfaces*, pages 231–273. Springer, 2006.

[104] Michael Gschwandtner, Roland Kwitt, Andreas Uhl, and Wolfgang Pree. BlenSor: Blender Sensor Simulation Toolbox Advances in Visual Computing. volume 6939 of *Lecture Notes in Computer Science*, chapter 20, pages 199–208. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-24030-0. doi: 10.1007/978-3-642-24031-7_20. URL http://dx.doi.org/10.1007/978-3-642-24031-7_20.

[105] Rahul Narain, Tobias Pfaff, and James F. O'Brien. Folding and crumpling adaptive sheets. *ACM Transactions on Graphics*, 32(4):51:1–8, July 2013. URL http://graphics.berkeley.edu/papers/Narain-FCA-2013-07/. Proceedings of ACM SIGGRAPH 2013, Anaheim.

[106] Rahul Narain, Armin Samii, and James F. O'Brien. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Transactions on Graphics*, 31(6): 147:1–10, November 2012. URL http://graphics.berkeley.edu/papers/Narain-AAR-2012-11/. Proceedings of ACM SIGGRAPH Asia 2012, Singapore.

[107] Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. Lightweight Binocular Facial Performance Capture Under Uncontrolled Lighting. *ACM Trans. Graph.*

[108] CloudCompare. URL http://www.cloudcompare.org/.

[109] Kassem Al Ismaeil, Djamila Aouada, Bruno Mirbach, and Björn E. Ottersten. Depth Super-Resolution by Enhanced Shift and Add. In *Computer Analysis of Images and Patterns - 15th International Conference, CAIP 2013, York, UK, August 27-29, 2013, Proceedings, Part II*, pages 100–107, 2013.

[110] Mohammad Rouhani and AngelD. Sappa. Non-rigid shape registration: A single linear least squares framework. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, volume 7578 of *Lecture Notes in Computer Science*, pages 264–277. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33785-7.

[111] K. Al Ismaeil, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten. Real-time non-rigid multi-frame depth video super-resolution. In *2015 IEEE Conference*

on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 8–16, June 2015. doi: 10.1109/CVPRW.2015.7301389.

[112] K. A. Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Dynamic super resolution of depth sequences with non-rigid motions. In *2013 IEEE International Conference on Image Processing*, pages 660–664, Sept 2013. doi: 10.1109/ICIP.2013.6738136.

[113] Kassem Al Ismaeil, Djamila Aouada, Bruno Mirbach, and Björn Ottersten. Enhancement of dynamic depth scenes by upsampling for precise super-resolution (up-sr). *Computer Vision and Image Understanding*, 2016. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2016.04.006. URL `http://www.sciencedirect.com/science/article/pii/S1077314216300303`.

[114] H. Afzal, K. Al Ismaeil, D. Aouada, F. Destelle, B. Mirbach, and B. Ottersten. KinectDeform: Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects. In *The 3DV Workshop on Dynamic Shape Measurement and Analysis*, December 2014.

[115] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 9(1):3–15, Jan 2003. ISSN 1077-2626. doi: 10.1109/TVCG.2003.1175093.

[116] David Levin. Mesh-independent surface interpolation. In Hamann Brunnett and Mueller, editors, *Geometric Modeling for Scientific Visualization*, pages 37–49. Springer-Verlag, 2003.

[117] Z.-Q. Cheng, Y.-Z. Wang, B. Li, K. Xu, G. Dang, and S.-Y. Jin. A survey of methods for moving least squares surfaces. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'08, pages 9–23, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. ISBN 978-3-905674-12-5.

[118] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9.

[119] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 950–953, New York, NY, USA, 2003. ACM. ISBN 1-58113-709-5.

[120] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 Papers*,

SIGGRAPH '08, pages 97:1–97:9, New York, NY, USA, 2008. ACM. ISBN 978-1-4503-0112-1.

[121] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26(3), July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276478. URL http://doi.acm.org/10.1145/1276377.1276478.

[122] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073207. URL http://doi.acm.org/10.1145/1073204.1073207.

[123] Yan Cui, Will Chang, Tobias Nöll, and Didier Stricker. Kinectavatar: Fully automatic body capture using a single kinect. In *Proceedings of the 11th International Conference on Computer Vision - Volume 2*, ACCV'12, pages 133–147, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 978-3-642-37483-8. doi: 10.1007/978-3-642-37484-5_12. URL http://dx.doi.org/10.1007/978-3-642-37484-5_12.

[124] Andrew Feng, Ari Shapiro, Wang Ruizhe, Mark Bolas, Gerard Medioni, and Evan Suma. Rapid avatar capture and simulation using commodity depth sensors. In *ACM SIGGRAPH 2014 Talks*, SIGGRAPH '14, pages 16:1–16:1, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2960-6. doi: 10.1145/2614106.2614182. URL http://doi.acm.org/10.1145/2614106.2614182.

[125] Hao Li, Robert W. Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proc. SGP'08)*, 27(5), July 2008.

[126] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, June 1989. ISSN 0162-8828. doi: 10.1109/34.24792. URL http://dx.doi.org/10.1109/34.24792.

[127] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3), July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276467. URL http://doi.acm.org/10.1145/1276377.1276467.

[128] Qian Zheng, Andrei Sharf, Andrea Tagliasacchi, Baoquan Chen, Hao Zhang, Alla Sheffer, and Daniel Cohen-Or. Consensus skeleton for non-rigid space-time registration. *Computer Graphcis Forum (Special Issue of Eurographics)*, 29(2):635–644, 2010.

[129] Will Chang and Matthias Zwicker. Range scan registration using reduced deformable models. *Computer Graphics Forum (Proceedings of Eurographics 2009), to appear.*

[130] C. Malleson, M. Klaudiny, A. Hilton, and J. Y. Guillemaut. Single-view rgbd-based reconstruction of dynamic human geometry. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 307–314, Dec 2013. doi: 10.1109/ICCVW.2013.48.

[131] Genzhi Ye, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*, chapter Performance Capture of Interacting Characters with Handheld Kinects, pages 828–841. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-33709-3. doi: 10.1007/978-3-642-33709-3_59. URL http://dx.doi.org/10.1007/978-3-642-33709-3_59.

[132] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, Dec 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.46.

[133] Mingsong Dou, Henry Fuchs, and Jan Michael Frahm. *Scanning and tracking dynamic objects with commodity depth cameras*, pages 99–106. 2013. ISBN 9781479928699. doi: 10.1109/ISMAR.2013.6671769.

[134] Hassan Afzal, Djamila Aouada, François Destelle, Bruno Mirbach, and Björn Ottersten. *Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015, Proceedings, Part II*, chapter View-Independent Enhanced 3D Reconstruction of Non-rigidly Deforming Objects, pages 712–724. Springer International Publishing, Cham, 2015. ISBN 978-3-319-23117-4. doi: 10.1007/978-3-319-23117-4_61. URL http://dx.doi.org/10.1007/978-3-319-23117-4_61.

[135] K. Al Ismaeil, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten. Real-time enhancement of dynamic depth videos with non-rigid deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[136] Djamila Aouada, Kassem Al Ismaeil, and Björn E. Ottersten. Patch-based statistical performance analysis of upsampling for precise super-resolution. In *VISAPP 2015 - Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Volume 1, Berlin, Germany, 11-14 March, 2015.*, pages 186–193, 2015. doi: 10.5220/0005316001860193. URL http://dx.doi.org/10.5220/0005316001860193.

[137] Djamila Aouada, Kassem Al Ismaeil, Kedija Kedir Idris, and Björn E. Ottersten. Surface UP-SR for an improved face recognition using low resolution depth cameras. In *11th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2014, Seoul, South Korea, August 26-29, 2014*, pages 107–112, 2014. doi: 10.1109/AVSS.2014.6918652. URL `http://dx.doi.org/10.1109/AVSS.2014.6918652`.

[138] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. ISBN 1848829345, 9781848829343.

[139] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 950–953, New York, NY, USA, 2003. ACM. ISBN 1-58113-709-5.

[140] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 9(1):3–15, Jan 2003. ISSN 1077-2626. doi: 10.1109/TVCG.2003.1175093.

[141] Inria4D. URL `http://4drepository.inrialpes.fr/pages/home`.

[142] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 53–60, Jan 2013. doi: 10.1109/WACV.2013.6474999.

[143] Sina Farsiu, Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multi-frame super-resolution. *IEEE Transactions on Image ProcessinG*, 13:1327–1344, 2003.

[144] Oliver Schall, Alexander G. Belyaev, and Hans-Peter Seidel. Adaptive feature-preserving non-local denoising of static and time-varying range data. *Computer-Aided Design*, 40(6):701–707, 2008. doi: 10.1016/j.cad.2008.01.011. URL `http://dx.doi.org/10.1016/j.cad.2008.01.011`.

[145] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. *ACM Trans. Graph.*, 22(3):950–953, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882368. URL `http://doi.acm.org/10.1145/882262.882368`.

[146] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998. doi: 10.1109/ICCV.1998.710815.

[147] Shin Yoshizawa, A. Belyaev, and H. P. Seidel. Smoothing by example: Mesh denoising by averaging with similarity-based weights. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 9–9, June 2006. doi: 10.1109/SMI.2006.38.

[148] David Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, October 1998. ISSN 0025-5718. doi: 10.1090/S0025-5718-98-00974-0. URL http://dx.doi.org/10.1090/S0025-5718-98-00974-0.

[149] Nina Amenta and Yong Joo Kil. Defining point-set surfaces. *ACM Trans. Graph.*, 23(3):264–270, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015713. URL http://doi.acm.org/10.1145/1015706.1015713.

[150] Shachar Fleishman, Daniel Cohen-Or, and Cláudio T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073227. URL http://doi.acm.org/10.1145/1073204.1073227.

[151] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. *ACM Trans. Graph.*, 26(3), July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276406. URL http://doi.acm.org/10.1145/1276377.1276406.

[152] Z.-Q. Cheng, Y.-Z. Wang, B. Li, K. Xu, G. Dang, and S.-Y. Jin. A survey of methods for moving least squares surfaces. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'08, pages 9–23, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. ISBN 978-3-905674-12-5.

[153] Marc Alexa and Anders Adamson. Interpolatory point set surfaces&mdash;convexity and hermite data. *ACM Trans. Graph.*, 28(2):20:1–20:10, May 2009. ISSN 0730-0301. doi: 10.1145/1516522.1516531. URL http://doi.acm.org/10.1145/1516522.1516531.

[154] Thierry Guillemot, Andres Almansa, and Tamy Boubekeur. Non Local Point Set Surfaces. In *(3DIMPVT 2012) Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 324–331. IEEE, October 2012. ISBN 978-0-7695-4873-9. doi: 10.1109/3DIMPVT.2012.71. URL http://perso.telecom-paristech.fr/~boubek/papers/NLPSS/.

[155] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, November 1992. ISSN 0167-2789. doi: 10.1016/0167-2789(92)90242-F. URL http://dx.doi.org/10.1016/0167-2789(92)90242-F.

[156] S. D. Babacan, R. Molina, and A. K. Katsaggelos. Total variation super resolution using a variational approach. In *2008 15th IEEE International Conference on Image Processing*, pages 641–644, Oct 2008. doi: 10.1109/ICIP.2008.4711836.

[157] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998. doi: 10.1109/ICCV.1998.710815.

[158] Antigoni Panagiotopoulou and Vassilis Anastassopoulos. Regularized super-resolution image reconstruction employing robust error norms. *Optical Engineering*, 48(11):117004–117004, 2009.

[159] A. Kheradmand and P. Milanfar. A general framework for regularized, similarity-based image restoration. *IEEE Transactions on Image Processing*, 23(12):5136–5151, Dec 2014. ISSN 1057-7149. doi: 10.1109/TIP.2014.2362059.

[160] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, Jan 2013. ISSN 1053-5888. doi: 10.1109/MSP.2011.2179329.

[161] Wenshu Li, Chao Zhao, Qiegen Liu, Qingjiang Shi, and Shen Xu. A parameter-adaptive iterative regularization model for image denoising. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–10, 2012. ISSN 1687-6180. doi: 10.1186/1687-6180-2012-222. URL http://dx.doi.org/10.1186/1687-6180-2012-222.

[162] Tom Goldstein, Christoph Studer, and Richard G. Baraniuk. A field guide to forward-backward splitting with a FASTA implementation. *CoRR*, abs/1411.3406, 2014. URL http://arxiv.org/abs/1411.3406.