




2019

## Depth Enhancement and Surface Reconstruction with RGB/D Sequence

Xinxin Zuo

University of Kentucky, xinxin.zuo@uky.edu

Author ORCID Identifier:

 <https://orcid.org/0000-0002-7116-9634>

Digital Object Identifier: <https://doi.org/10.13023/etd.2019.447>

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

### Recommended Citation

Zuo, Xinxin, "Depth Enhancement and Surface Reconstruction with RGB/D Sequence" (2019). *Theses and Dissertations--Computer Science*. 91.

[https://uknowledge.uky.edu/cs\\_etds/91](https://uknowledge.uky.edu/cs_etds/91)

This Doctoral Dissertation is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact [UKnowledge@lsv.uky.edu](mailto:UKnowledge@lsv.uky.edu).

## **STUDENT AGREEMENT:**

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

## **REVIEW, APPROVAL AND ACCEPTANCE**

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Xinxin Zuo, Student

Dr. Ruigang Yang, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

Depth Enhancement and Surface Reconstruction with RGB/D Sequence

---

DISSERTATION

---

A dissertation submitted in partial  
fulfillment of the requirements for the  
degree of Doctor of Philosophy in the  
College of Engineering at the  
University of Kentucky

By  
Xinxin Zuo  
Lexington, Kentucky

Director: Dr. Ruigang Yang, Professor of Computer Science  
Lexington, Kentucky 2019

Copyright© Xinxin Zuo 2019

ORCID: <https://orcid.org/0000-0002-7116-9634>

## ABSTRACT OF DISSERTATION

### Depth Enhancement and Surface Reconstruction with RGB/D Sequence

Surface reconstruction and 3D modeling is a challenging task, which has been explored for decades by the computer vision, computer graphics, and machine learning communities. It is fundamental to many applications such as robot navigation, animation and scene understanding, industrial control and medical diagnosis. In this dissertation, I take advantage of the consumer depth sensors for surface reconstruction. Considering its limited performance on capturing detailed surface geometry, a depth enhancement approach is proposed in the first place to recovery small and rich geometric details with captured depth and color sequence. In addition to enhancing its spatial resolution, I present a hybrid camera to improve the temporal resolution of consumer depth sensor and propose an optimization framework to capture high speed motion and generate high speed depth streams. Given the partial scans from the depth sensor, we also develop a novel fusion approach to build up complete and watertight human models with a template guided registration method. Finally, the problem of surface reconstruction for non-Lambertian objects, on which the current depth sensor fails, is addressed by exploiting multi-view images captured with a hand-held color camera and we propose a visual hull based approach to recovery the 3D model.

KEYWORDS: computer vision, surface reconstruction, depth enhancement

Author's signature: \_\_\_\_\_ Xinxin Zuo

Date: \_\_\_\_\_ December 11, 2019

Depth Enhancement and Surface Reconstruction with RGB/D Sequence

By  
Xinxin Zuo

Director of Dissertation: Ruigang Yang

Director of Graduate Studies: Mirosław Truszczyński

Date: December 11, 2019

## ACKNOWLEDGMENTS

For these years, I have received a tremendous amount of help and support from my advisor, my colleagues, my friends and my families. Without them, I will not have been able to finish this dissertation and earn my PhD degree.

I would like to express my first sincere appreciation to my advisor, Dr. Ruigang Yang, for introducing me into the exciting area of computer vision and for years of support and guidance throughout my PhD journey. I am so grateful that I have been able to conduct research on topics that I am really interested in under Dr. Yang's supervision. The inspirations and encouragement I have received from Dr. Yang have had significantly impacts on my ways of thinking, exploring and doing. I am more than fortunate to have been able to walk through my PhD journey with him. He is quite energetic and cheerful, has long-lasting enthusiasm in his career. I have learned a lot from his unique characteristics and they will continue to inspire me through all the challenges that I will face in my future endeavors.

I also want to thank all my committee members, Dr. Brent Seales, Dr. Nathan Jacobs, Dr. Qiang Ye and Dr. Kwok-Wai Ng. I really appreciate their efforts on guiding my study and on my dissertations. I also want to give thanks to my advisor, Jiangbin Zheng, in Northwestern Polytechnical University in China who has helped me a lot and supported me when I first start my PhD.

I have also been so lucky to share my Ph.D. life with a group of lovely people. They are all so creative, inspiring to work with and so nice to get along with. We worked and played together for all these memorable years. They have all contributed to my work and my life. I want to express my thankfulness to all of them, in particular Chao Du, Yajie Zhao, Shunnan Chen, Wei Li, Hao Zhu, Mao Ye, Qi Sun, Hui Zhang, Xiuxiu Li.

Finally but most importantly, I would like to thank all members in my family from the bottom of my heart, in particular my husband, my parents, my sister and my brother. My deepest appreciation goes to them. Without their unconditional love and constant support, I cannot accomplish this today and would not be able to stick to the completion of my PhD study.

# Table of Contents

<b>Acknowledgments</b> . . . . .	<b>iii</b>
<b>Table of Contents</b> . . . . .	<b>iv</b>
<b>List of Figures</b> . . . . .	<b>vi</b>
<b>List of Tables</b> . . . . .	<b>x</b>
<b>Chapter 1 Introduction</b> . . . . .	<b>1</b>
1.1 Background . . . . .	1
1.2 Motivation . . . . .	2
1.3 Contributions . . . . .	4
1.4 Structure . . . . .	5
<b>Chapter 2 Related Work</b> . . . . .	<b>6</b>
2.1 Surface Reconstruction . . . . .	6
2.2 Depth Enhancement . . . . .	9
2.3 Human Body Modeling . . . . .	10
<b>Chapter 3 Detailed Surface Geometry and Albedo Recovery from RGB-D Video Under Natural Illumination</b> . . . . .	<b>13</b>
3.1 Pipeline . . . . .	14
3.2 Preliminary Theory . . . . .	15
3.3 Approach . . . . .	16
3.4 Experimental results . . . . .	23
3.5 Conclusion . . . . .	29
<b>Chapter 4 High-speed Depth Stream Generation from a Hybrid Camera</b> . . . . .	<b>32</b>
4.1 System setup . . . . .	33
4.2 Our Approach . . . . .	33

4.3	Experiments . . . . .	40
4.4	Conclusion . . . . .	47
<b>Chapter 5 SparseFusion: Dynamic Human Body Reconstruction from Sparse</b>		
	<b>RGBD Images . . . . .</b>	<b>48</b>
5.1	Approach . . . . .	49
5.2	Experiments . . . . .	56
5.3	Conclusion and Future work . . . . .	57
<b>Chapter 6 Interactive Visual Hull Refinement for Specular and Transparent</b>		
	<b>Object Surface Reconstruction . . . . .</b>	<b>60</b>
6.1	Approach . . . . .	61
6.2	Internal Contour Tracking . . . . .	65
6.3	Concave refinement . . . . .	69
6.4	Experiments and Results . . . . .	70
6.5	Conclusion . . . . .	73
<b>Chapter 7 Summary . . . . .</b>		
7.1	Conclusions . . . . .	75
7.2	Limitations and Extension . . . . .	76
7.3	Future work . . . . .	78
<b>Bibliography . . . . .</b>		<b>80</b>
<b>Vita . . . . .</b>		<b>96</b>



## LIST OF FIGURES

3.1	System Pipeline. . . . .	14
3.2	Diverse rotation variations resolve local ambiguity. (a) shows a sampled image of the object with a reference pixel marked as red. In (b),(c) and (d) we plot the energy map for this reference pixel with x-axis and y-axis representing the two degree of freedom of a surface normal. The cooler color in these figures corresponds to smaller energy value. As we can see, given a single image the solution lies in a large band as shown in (b). With three images the normal converges better as shown in (c). And finally we will be able to find the optimal surface normal for the reference pixel if we have got enough images under rotation variations as shown in (d). . . . .	17
3.3	Demonstration of correspondence matching. (a) is the reference frame and (b) is one sampled key frame. Image(b) is warped to the reference frame with current transformation, and (c) displays the warped image overlaid with image(a). (d) shows the overlaid result using the flow map computed from warped image and the reference image [19]. (e) is the overlaid image after applying our proposed lighting insensitive robust matching. . . . .	18
3.4	Results on synthetic models. (a1-a3) is the rendered color image of the reference frame; (e1-e3) shows the normal map of the ground-truth mesh after over smoothing; (b1-b3) is the normal map computed after applying shading refinement on the reference frame with its error map displayed in (f1-f3); (c1-c3) and (g1-g3) are the normal map and its corresponding error map achieved by our method. Our recovered albedo map and its error map is also demonstrated in (d1-d3) and (h1-h3) respectively. . . . .	25
3.5	Results when adding salt and pepper noise. (a) shows the computed normal map without our EM framework and (b) is its error map; The normal and error map after applying our EM optimization are shown in(c) and (d) respectively. . . . .	26

3.6	Comparison results on Frog, Shoe and Chinese Fan model. (a1) (a2) and (a3) are the reference color images of Frog, Shoe and Chinese Fan respectively. The outputs from KinectFusion are shown in (b1) (b2) and (b3). The results computed by shading refinement method [105] are displayed in (c1) (c2) and (c3). (d1) (d2) and (d3) are the meshes computed by depth super-resolution method [47]. Finally, (e1) (e2) and (e3) are the output meshes from our approach with their corresponding normal maps displayed in (f1) (f2) and (f3). . . . .	27
3.7	Comparison results on Backpack and Turtle model. (a1) and (a2) are the reference color images of Backpack and Turtle respectively. The output from shading refinement method [105] is shown in (b1) and (b2). The results computed by depth super-resolution [47] are displayed in (c1) and (c2). (d1) and (d2) are the meshes acquired using our method but without applying our locally robust matching procedure. (e1) and (e2) are the output meshes of our method and the corresponding normal maps are displayed in (f1) and (f2). . . . .	28
3.8	Results on Book model. (a) is the reference color image. (b) shows the refined mesh with shading refinement method [105]. (c) shows the super-resolution results [47]. The recovered mesh surface from our method is displayed in (d). . . . .	30
3.9	Comparison results on albedo recovery or intrinsic decomposition of the Turtle, Frog, Shoe and Backpack model. The first column is the input color image with its corresponding depth map. The second column shows the result of Chen [21]. The third column is the decomposed albedo and shading images from method in [60]. Finally, the last column demonstrates the result achieved by our method. . . . .	31
4.1	System Pipeline. . . . .	33
4.2	Results on intrinsic decomposition. (a) a sampled middle frame; (b) decomposed shading image when constant shading smooth weight $\omega_t^n$ set as 1.0; (c) shading image computed when $\omega_t^n$ is set as 6.0; (d) our shading image with adaptive smooth shading weight. . . . .	38
4.3	Results on sampled images from hand waving stream. (a) Input color image and depth map. (b-e) Albedo and Shading images estimated by three recent approaches for intrinsic decomposition and by our approach. . . . .	42

4.4	Results on the hand waving sequence. To better show the shape of the hands, we adjust the perspective angle of the generated mesh. We have two key frame color images and the initial meshes as input shown in (a) and (c). (d) and (h) are the key frame meshes after refinement. The second row shows the interpolation result of one sampled frame shown in (b). (e)-(g) are the recovered meshes using BL, our proposed method and SBL, respectively. . . . .	43
4.5	Results on the towel shaking sequence. To better demonstrate the shape of the towel, we adjust the perspective angle of the generated mesh which differs from the color image. First row shows the input color sequence with left most and right most have the corresponding input depth frame, while the middle ones are sampled frames between these two frames. Second row gives the input depth on the left and right, and interpolated depth with BL method for the middle frames. Third row displays the shading refinement result using SBL method. The last row shows the recovered depth from our method. . . . .	44
4.6	Results on synthetic data (simple case). . . . .	45
4.7	Results on synthetic data (more complex case). The two shading images (b) and (d) are the same to each other as we set the sequence to have half period of sine wave and these two key frames have the same shape, but they have the global displacement in depth, as shown in (c). . . . .	45
4.8	Quantitative results for real datasets. All the results are computed with squared mean error. We discard the extreme outliers around the surface boundaries during the evaluation. . . . .	46
5.1	Initial Fitting results. (a) is the input RGBD frame and we show the detect joints on the color image. (b) shows the optimized SMPL aligned with the input scan. (c) shows the deformed input scan that fits even better to the SMPL model. . . . .	51
5.2	Pairwise registration results. (a) and (b) are two sampled pieces. (c) shows our registration result of (a) and (b). The mesh of (a) is deformed onto the mesh of (b). . . . .	53
5.3	Texture optimization results. . . . .	55
5.4	Results on a synthetic dataset. . . . .	57
5.5	Results on real datasets. The left two columns are sampled input scans and the three right columns are the fused model and models deformed to some input scans. . . . .	58
5.6	Results on changing topologies. . . . .	59

6.1	The reconstructed models for glossy trophy and glass sculpture. The left column shows the capture object; the middle column shows the reconstructed 3D model using visual hull; and the right column presents the reconstructed surface using our proposed method. . . . .	61
6.2	The system pipeline of our approach . . . . .	62
6.3	Illustration for occluding contours and visual hull. $S$ is the real surface and $CH(S)$ is the convex hull which is same as visual hull ( $VH(S)$ ) in 2d. $P$ is the internal occluding contour point while $Q$ is the external occluding contour point. . . . .	63
6.4	Locally Convex Carving. The left figure represents the case that the intersected tangent lines coming from two different convex surface and the contour normal are opposite from each other, and in this case we cannot carve out the Region $R$ . The right one is what we defined as locally convex carving cases where the Region $R$ can be carved. All the tangent lines in these figures indicate internal contours. $N_p$ and $N_q$ are the contour normals and $N_{ep}$ is the normal for the epipolar plane. . . . .	64
6.5	Locally convex carving Illustration of convex and concave cases. The figure presents the two cases that illustrate the additional regions that can be carved out with LLC and also the region that we will get after the locally convex carving operation. . . . .	65
6.6	Geometric illustration for contour prediction. See more details in the text. . . . .	66
6.7	Intensity histogram. The upper row is captured images with internal contours highlighted in red, and the second row is the histogram of corresponding contours. Images in the first and second column are from nearby viewpoints and so as the third and fourth column. . . . .	68
6.8	Concave boundary illustration. The two images illustrate the boundary cues used in concave fitting for these two models. The projection of these 3D boundary vertices is $RV$ area in the frontal 2d image. . . . .	69
6.9	Comparison of contour tracking. The detected pixels are highlighted in images. The first row shows the results using only the gradient data term and the pairwise smoothness term. The second row shows the results with all the data terms and regularization terms. . . . .	71
6.10	Comparison of reconstructed 3d models. For each object, the first row shows the original image from one viewpoint and reconstructed visual hull surface, and the second row shows the rendered and reconstructed model with our proposed method. . . . .	72

## LIST OF TABLES

3.1	Quantitative Evaluation. . . . .	24
4.1	The parameter settings for our experiments. . . . .	40
5.1	Reconstruction Error . . . . .	56
6.1	Mean error of tracked contours. The table gives the mean pixel error on four datasets and the rows indicate the terms that were incorporated. G donates the gradient term data (Eq. 6.1) and IH is the term using histogram of color intensity (Eq. 6.3); P and T are two regularization terms of Eq. 6.5 and Eq. 6.7 respectively. . . . .	71

# Chapter 1

## Introduction

Nowadays it becomes quite easy to acquire images using a camera or mobile phone, and meanwhile there are many computer vision techniques, such as object detection and recognition [72, 26] that operate on 2D images or videos. However, 2D images are not sufficient for tasks like navigation [81], object manipulation [62], remote control [52], scene understanding [130] since we live in a 3D world where scenes have volume and are spatially arranged with objects occluding each other. The ability to reason about the 3D properties is the basic technique for accomplishing these tasks, and has been widely used in games, virtual reality, teleconferences, etc. As humans, we perceive the three-dimensional structure of the world with apparent ease as we have left and right eyes using which we can infer the distance. But for computers, it is a non-trivial task and it has been studied for decades by computer vision community [129, 45].

### 1.1 Background

Computer graphics studies the forward models of how light is reflected from the surface of an objects, scattered by the atmosphere, refracted through camera lenses and finally projected onto a 2D image plane. It is assumed that 3D shapes and surface appearance are already available. But in computer vision, we are trying to do the inverse by reconstructing the surface properties, such as shapes and appearance from the observed images. In fact, the desire to recover the three-dimensional structure of the world from images and to use it as a stepping stone towards full scene understanding is an important branch of computer vision.

3D reconstruction is a longstanding ill-posed problem that has been explored for decades. Early attempts at 3D reconstruction involved extracting edges and then inferring the 3D structure of an object or a blocks world from the topological structure of the 2D

lines [115]. Several line labeling algorithms [138, 58] were developed at that time. Later on, the modeling of non-polyhedral objects was studied using general cylinders [14]. Similarly, the shape-from-X methods [99, 161] estimate 3D shapes from cues contained in the image. For example, the shape from shading approach [55] exploits the shading information and infers the surface by analyzing the image formation process.

Starting from the late 70s, feature-based correspondences matching algorithms [93, 94] emerged and 3D structure could be reconstructed by triangulation. The structure-from-motion framework [136] was proposed based on this feature matching strategy which recovers 3D structure and camera motion simultaneously with bundle adjustment optimization. Later on, as we go from sparse to dense and watertight surface reconstruction, many approaches have been proposed on multi-view stereo [6, 140] which utilize multiple color cameras to reconstruct 3D models by exploiting the photo consistency constraints together with smoothness regularizations. After nearly four decades of active research, we can now achieve very good performance with high precision and robustness on stereo matching [122]. In this case, the surrounding camera arrays have been widely used in laboratory environment especially when we are dealing with humans experiencing non-rigid motion. Another branch of 3D modeling [167] is to directly exploit the depth sensors from which the depth information is readily available without resorting to the classical stereo matching. The availability of low cost commodity depth sensors, such as Microsoft Kinect, has made the static scene modeling substantially easier than ever. Many scanning systems [145, 98, 34] have been proposed for indoor modeling and they were also extended to fusion of dynamic objects lately [111, 87, 63]. More recently, with the avenue of deep learning techniques, researchers have explored more lightweight solutions that are able to recover 3D shapes of objects from a single [134, 23] or a few RGB images [155, 22]. It has become a new trend in surface reconstruction, especially for the modeling of human body [65, 165, 119] and 3D face recovery [33].

## 1.2 Motivation

Although classic computer vision techniques such as image segmentation [125], object tracking [77] and recognition [15] are usually studied to implement on images and videos in 2D domain, the depth information has been proven to be substantially useful for solving various computer vision problems, like indoor modeling [126], 3D object detection [46], scene understanding [130], and so on. With the prevail of affordable consumer depth cameras, 3D information becomes easier to acquire without the effort of extracting 3D structure from 2D images.

However, despite of the ubiquitous usage of depth sensors, the captured depth images generally suffer from various degrees of noises, unavailable data in certain areas, as well as low resolution [162, 44]. To be even worse, it will fail completely on objects with highly non-lambertian surface reflectance as neither the stereo based depth sensor nor the Time-of-Flight camera can establish correct correspondences or measure the time shift in this case. Also, the depth sensor can only capture a partial piece of the object from a single view. A fusion procedure will be needed to recover a complete 3D model. These limitations and problems have motivated my research as described below in details.

1. The current available consumer depth sensors have limited resolution and accuracy as compared to color cameras. As a result, fine-scale structural details of an object cannot be recovered. Therefore, I believe that exploiting high quality color image as guidance to enhance the depth map would be desired.

2. Among all the publicly available commodity depth sensors, the SwissRanger and PMD can capture the depth at higher speed than 30Hz, but with a much lower resolution at about  $100 * 200$ . For the well known Kinect depth sensors version 1 and version 2, both have a refresh rate of 30Hz. In the meantime, high-speed video as high as 120Hz has been commonly adopted in consumer-grade cameras. Thereafter, I believe augmenting these videos with a corresponding depth stream will enable new multimedia applications, such as 3D slow-motion video.

3. Deformable objects are one of the most general categories in our daily lives. One particular example would be human body. 3D dynamic digital humans are essential for a variety of applications ranging from gaming, visual effects to free-viewpoint videos. However, high-end capture solutions use a large number of cameras, and are restricted to professional as they operate under controlled lighting conditions and studio settings. Instead of using surrounding cameras, a single sensor is preferred which is more portable and easy to set up.

4. 3D reconstruction of objects with lambertian reflectance has been studied for many years. However, there are a large portion of objects in our daily life which are made of non-lambertian material. Due to their non-Lambertian surface reflectance properties, establishing correspondences a fundamental requirement for many 3D reconstruction algorithms becomes difficult or even impossible. So targeting for reconstructing objects with complex surface material such as specular or transparent, we want to investigate novel methods for such tasks. We believe that this is a step forward in reconstructing objects made of more general materials.



## 1.3 Contributions

In this dissertation, I make contributions in three main areas: 1) a novel approach is proposed to enhance the spatial and temporal resolution of the depth sensor with the help of corresponding RGB videos, 2) I propose methods to reconstruct complete and watertight surface for deformable objects, particularly for human bodies, using a single depth sensor, and 3) I go beyond the lambertian surface and propose a method for 3D reconstruction of specular and transparent objects. The major contributions are listed in details as below:

1. First I present a novel approach for depth map enhancement that recovers both highly detailed surface geometry and its appearance from an RGB-D video sequence. Instead of making any assumption about the surface albedo or controlled object motion and environment lighting, I exploit the lighting variation introduced by casual object movement. We are able to recover the surface normal and albedo simultaneously, without any regularization term under natural illumination.

2. To improve the temporal resolution of the depth sensor, I present a hybrid camera system that combines a high-speed color camera with a depth sensor, e.g. Kinect depth sensor, to generate a high-speed depth sequence. I find that simply interpolating the low-speed depth frames is not satisfactory, where interpolation artifacts and loss of surface details are often visible. Therefore, I present an optimization-based framework to estimate the high-resolution/high-speed depth stream, taking into consideration temporal smoothness and shading/depth consistency.

3. A novel framework is proposed to build up 3D human avatars with sparse frames using a single RGBD camera. It is a challenging problem as we consider the various pose changes and surface occlusion. I address this problem by exploiting a generative human template to find initial alignment between every two frames that have great overlap. A global non-rigid registration procedure is performed afterwards to deform those partial scans into a unified model. Finally, I build consistent and clear texture maps for the reconstructed human model with a flow based texture map optimization approach.

4. To address the difficulties of surface reconstruction for non-Lambertian objects, I present a method using standard multi-view images. I extend the original visual hull concept to incorporate 3D cues presented by *internal occluding contours*, i.e., occluding contours that are inside the object's silhouettes. It is discovered that these internal contours, which are results of convex parts on an object's surface, can lead to tighter fit than the original visual hull.

## 1.4 Structure

The remainder of this dissertation is structured as follows. In Chapter 2, I will discuss the necessary background and previous work on surface reconstruction, depth enhancement and human body modeling. I present a novel depth map enhancement approach in Chapter 3 that recovers the surface details beyond the resolution of current depth sensors as well as the surface appearance. Next, our high speed depth stream generation framework is described in Chapter 4 showing my system setups and the proposed depth stream up-sampling approach. In Chapter 5, I describe our method of building up human avatars from sparse RGBD frames. In addition to lambertian surface modeling, in Chapter 6 I will deal with the problem of surface reconstruction for specular and transparent objects using multi-view images captured by a hand-held camera. The conclusion and future work are finally presented in Chapter 7.

# Chapter 2

## Related Work

In this chapter, I will review some previous works on surface reconstruction, depth enhancement, human modeling and also 3D modeling of non-lambertian object which are related to my work.

### 2.1 Surface Reconstruction

We can divide the approaches of surface reconstruction into passive and active ones depending on whether any active light or signal is involved.

#### 2.1.1 Passive Methods

For the passive methods, the input is purely 2D color images and the goal of image-based 3D reconstruction [114] is to infer the 3D geometry and structure of objects and scenes from one or multiple 2D images. Ingenious work on "Shape-from-X" has utilized priors on natural images to infer geometric features, with "X" being shading [161], texture [8], specularity [50], silhouettes [141], shadow [109], motion [136] and so on. For example, the shape from silhouettes approach allows to obtain 3D shape of an object from their profiles in multiple views by volume intersections. The very first attempts [18] dated back to 1990, where the reconstructed object tends to be simple in shape. Later on, the stereo vision was brought up and the shape-from-motion framework [120] was studied to reconstruct the 3D shape of the scene while calculating the position of camera. Meanwhile, great success was achieved in multi-view stereo (MVS) [122], which addresses the problem of dense 3D model reconstruction from a collection of images taken from known viewpoints with intrinsically calibrated cameras. The passive reconstruction methods also give us the opportunity to use massive amounts of visual information available on the web. To give an

example, there is a project, "Building Rome in a Day" [5] in which basing on a collection of thousands of images of Rome, they made a visual reconstruction of one of the city's main parts. Till now, there are softwares such as the Altizure [1] which is able to reconstruct the 3D models of individual objects and also city-scale complex scenes.

More recently, the learning based solutions [155, 156, 128, 139] are growing rapidly as with the development of large 3D shape databases [3, 69, 121]. Rather than using conventional handcrafted image features and matching metrics [54], recent studies on stereo matching apply the deep learning techniques for better pair-wise patch matching [49]. Besides of stereo matching, the first end-to-end network for the MVS problem, called SurfaceNet, was proposed [61], which pre-computed the cost volume with sophisticated voxel-wise view selection, and used 3D CNN to regularize and infer the surface voxels. As compared with the conventional approaches on MVS, the learning based methods have demonstrated superior performance on dealing with textureless objects [155, 116]. Besides, instead of using dense multiple images, we could take sparse images or even single image as input. For example, Choy et al. [25] proposed an unified framework for single and multi-view reconstruction by using a 3D recurrent neural network based on long-short-term memory. The learning based approaches have become a new trend in surface reconstruction, while there are still some issues that need further investigation. For example, if we want to reconstruct the objects at a finer resolution for higher quality reconstructions, it will become extremely computationally expensive. Besides, the generalization ability is another concern especially when we have limited training dataset. Therefore, researchers are starting to work on unsupervised approaches [67] which is more scalable as large amounts of training data can be more easily acquired.

### **2.1.2 Active Methods**

Basically, there are two main approaches of range sensing, namely triangulation and Time-of-Flight [27]. The first can be implemented as a passive approach, i.e., stereo vision, or as an active system, such as structured light [66]. Stereo vision calculates the disparity between two images taken at different positions. The structured light camera projects an infrared light pattern onto the scene and estimate the disparity from the perspective distortion of the pattern. Ranging scanners and ToF cameras, on the other hand, measure the time it takes for light emitted by an illumination unit to travel to the object and return to the detector.

The core technology behind the structured light or time-of-flight based depth cameras dates back several decades. However, the advent of affordable consumer grade RGB-D

cameras like Microsoft Kinect and Intel RealSense has brought profound advances in visual scene reconstruction methods. For instance, the seminal KinectFusion work [101] which enables real-time scanning and scan integration, had remarkable impact in the computer graphics and vision communities. After that, many following up systems [102, 24, 164] were proposed to tackle the drifting problem [144] and live scanning of large scenes [31]. Furthermore, researchers go beyond the rigid objects modeling to capturing dense 3D geometry models of dynamic scenes, such as models of moving humans [131], or of general deformable surfaces [87]. It becomes possible to obtain detailed models of the non-rigid objects with only a single depth camera.

### 2.1.3 Surface Reconstruction of Non-Lambertian Objects

Surface reconstruction of non-Lambertian objects is challenging and methods using traditional stereo correspondence are not sufficient for these objects, since the complex reflection effects are not valid under the Lambertian assumption. There are successful approaches [32, 82, 142] that use structured light methods relying on specialized patterns, where the surface depth or normal is computed by analyzing the captured patterns. In paper [32] a checkerboard pattern is used and observed after distorted by the transparent objects; Liu et al. [82] design a set of frequency-based patterns. Zickler et al. [166] use Helmholtz stereopsis for surface reconstruction with arbitrary and unknown surface reflectance. The captured signal is transformed from the time domain into the frequency domain to solve the correspondence problem. As for specular objects, existing state-of-art methods can be broadly classified into two categories, namely *shape from specular flow* and *shape from specular correspondences* [83]. The first method assumes a known continuous motion and tries to track the dense specular flows, while the second one uses a reference plane with a known pattern as guidance to predict the unknown surface. The above methods all need careful setup and extra projectors or light sensors. There are also methods [91] that try to separate the specular reflection effects from diffuse reflection and use traditional photometric stereo methods for surface reconstruction afterwards. However, it is rather difficult to perform the separation on highly specular objects as we consider the complex reflection, surrounding environments and also the lighting conditions. Recently, Wu et al. [149] have proposed to train a neural network achieve this goal. But its generalization capability is questionable with the limited training dataset.

## 2.2 Depth Enhancement

Existing depth cameras generally suffer from various degrees of noises, as well as low resolution as compared to high quality body scanners. It poses significant challenges for shape reconstruction, especially when rich geometric details are desired. As the resolution of color images is usually several times higher and there is a high correlation between structural features of the color image and the depth map (e.g., object edges), it is natural to use the color image as guidance for depth map enhancement [106, 37, 151, 17, 96].

### 2.2.1 Depth super-resolution vs. Shading based refinement approaches

For depth super-resolution, the basic idea is to use the corresponding color image captured from the same scene to recover a high resolution depth map by exploiting the strong structural correlations between depth and texture. One way is to recast the depth super-resolution task as a global optimization problem [152, 79], in which, the data term penalizes the difference between the observation and the recovered depth, while the smooth term regularizes neighboring pixels based on the designed priors. For instance, image guided depth upsampling using anisotropic Total Generalized Variation [38] and Non-Local Means [107] are very classical color assisted depth image super-resolution approaches. However, these methods often use hand-designed objective functions which cannot express priors in real images well and are typically time-consuming. Another category of depth upsampling methods [97, 84, 13] uses designed filters to apply joint filtering on the depth map under guidance of the color image. For example, Yang et al. [153] employed edge-preserving filters to upsample a depth image. Hua et al. [56] approximately applied the filtering procedure with local gradient information of the depth image. These methods are established on the assumption that local pixels with similar color will have similar depth value. However, sometimes this assumption is unfounded which will result in texture copying artifact, and blurry edges will occur on textureless color and textured depth or when the color and depth edges are not well aligned.

Another promising category is the learning-based methods [78, 143], which learns the relation between the low resolution and high resolution depth map. It enjoys fast testing speed, and delivers more promising performance than the above methods, but a sufficient amount of training data will be needed to generalize well to different scenario of test data.

From the depth super-resolution, we will get a high resolution, noise-free depth image with clear surface edges. However, the detailed structural information is still unrevealed. There are other methods that take advantage of the shading information contained in color

images so as to recover geometric surface details. The Shape-from-shading (SfS) problem has long been studied ever since the pioneering work by Horn [55]. It aims to estimate surface normal (and then indirectly surface shape) from a single image. There are various regularization terms or prior assumptions [10, 12] that have been enforced to deal with the inherently ill-posed problem. Some methods [48, 157] have shown that SfS can be used to refine the noisy depth map captured from RGB-D cameras as well.

On the basis of Shape-from-Shading (SfS) techniques, most of these shading based methods implement shading refinement on a single RGB-D frame [157, 105, 148]. The inherent ambiguity of SfS is not resolved exactly, but with the initial depth close to the real surface, Wu et al. [148] and Roy et al. [105] have achieved good performance in recovering surface details. More recently, Haefner et al. [47] have combined heterogeneous depth and color data to jointly solve the ill-posed depth super-resolution and shape from shading problem. Varying albedo poses another challenge as it needs to be factored out before lighting estimation and shading refinement. Some works [48] assume uniform or piecewise constant albedo. Yu et al. [157] dealt with this by clustering a fixed set of discrete albedos before optimizing geometry. A better, yet more complex strategy, is to simultaneously optimize for unknown albedos and refine geometry [68]. There are also previous works that adopt the shading constraints to improve the coarse 3D shape reconstructed from multi-view stereo [147]. More recently, Maier et al. [90] have optimized textures and the geometry encoded in a signed distance function in a unified framework under estimation of spatially-varying spherical harmonics which has achieved the state-of-the-art results on reconstructed scene geometry.

## **2.3 Human Body Modeling**

### **2.3.1 Template-free vs. template-based body modeling**

Template free methods reconstruct the moving geometry by mesh deformation [6, 20] or using volumetric representations for the surface [57, 7]. The advantage of these methods is that they allow reconstruction of general dynamic objects. While flexible, such approaches require high-quality multi-view input data. An alternative solution is to use a depth camera and perform non-rigid registration between incoming depth frames and a concurrently updated, initially incomplete template [111]. While general, such template-free approaches are limited to slow and careful motion. There are some papers that exploit pre-scanned human models as template, which makes the surface tracking problem easier to handle as the overall shape is already available. For example, in paper. [89], the template was

pre-scanned in the first place and then got deformed to fit the input acquired from a depth sensor. Later on, Guo et al. [64] improved the surface tracking performance by incorporating both L0 and L2 regularizations. These works are in-between template-free and template-based methods.

As compared with template-free methods, template-based approaches leverage a parametric body model for human pose and shape estimation from images. Early models in computer vision were based on simple primitives [95, 41]. The recent statistical human models, like SCAPE [9] and SMPL model [88], are learned from thousands of scans of humans under real pose. The pose and shape deformations are encoded in the parametric model. Therefore, instead of tracking the deformation of all those vertices on the surface, some works [36, 110] solved the pose and shape coefficients of the statistical model. Also, with the progress of deep convolutional neural networks and human pose estimation, it has become possible to recover body shapes from a single image [104, 137, 135] by regressing the parameters of statistical human models. The first automatic method was proposed in SMPLify [16] where the SMPL model was fitted to the 2D keypoints estimated from an image using ConvNets with an optimization technique. Constraints like silhouettes are also incorporated for shape estimation [133, 165]. Kanazawa et al. [65] proposed an end-to-end learning system of human body and shape based on generative adversarial networks (GANs). The template-based approaches work naturally with the learning methods as we just need to predict the coefficients using the neural network. However, there are some exceptions [103] which are template-free and use volumetric representations where a neural network is trained predict the volume occupancy.

Overall, the template-based approaches are more reliable in handling occlusions, complex motion, and work well with limited input such a single or few images. However, the recovered human body lies on the space spanned by the models that we have used to train the parametric model. On the contrary, the template-free approaches are more flexible and can represent surface with geometric details. But in the meantime, the reconstruction relies on more reliable input data, such as multi-view or depth images.

### **2.3.2 RGB image vs. depth input**

The human shape reconstruction problem has been studied for decades under the multi-view stereo setup [146, 6]. They exploit the correspondence cues between images of neighboring views and also the temporal consistency to construct the involving surface. The multiple cameras are synchronized and this controlled setup is usually used in the laboratory. On the contrary, deep learning based human body reconstruction methods have demon-



strated its advantages in human body recovery from images in the wild [104, 137, 135]. They take a single RGB image as input or use sparse images of the human subject from different view directions [80]. Despite of the widespread usage, the reconstructed human body usually lacks sufficient surface details. More importantly the inherent depth ambiguity of the RGB image stops the reconstructed human body from fitting closely to the real surface.

The ambiguity can be easily resolved by utilizing a depth sensor. There are some works that use only a single depth sensor for the non-rigid objects reconstruction and specifically for the human body. First, as an extension to the KinectFusion system, a dynamic fusion approach [111] has been proposed which takes non-rigid motion into account by solving a non-rigid warp field for every frame. Later on, sparse feature information [87] and dense color correspondences [63] in the color sequence were incorporated to improve the robustness of surface tracking. Yu et al. [131] enforced skeleton constraints in the typical fusion pipeline to get better performance on both surface fusion and skeleton tracking. A more robust fusion approach [132] was proposed by tracking both the inner and outer surface. Those methods allow the user to move more freely. However, as the sequence proceeds the almost inevitable drifting problem makes it difficult to recover a complete model without loop closure. There are also works [86, 74] that generate partial pieces in the first place and handle the error accumulation problem by using a global non-rigid registration. For example, Dou et al. [86] proposed a non-rigid bundle adjustment method where impressive results have been obtained. But the bundle adjustment could be quite computationally expensive and time-consuming due to the large number of unknowns and search space.

## Chapter 3

# Detailed Surface Geometry and Albedo Recovery from RGB-D Video Under Natural Illumination

Due to the limited spatial resolution and sensor noise, the current consumer depth camera fails to capture the important surface details. In this chapter, we propose a method to recover highly detailed surface geometry as well as its appearance from an RGB-D sequence by exploiting the shading information. More specifically, we capture the RGB-D sequence with a Kinect V2 depth sensor attached with a relatively high quality color camera. During the acquisition process, we can rotate the object casually in front of the cameras with the depth and color cameras being static. In this way, the illumination changes in the image sequence induced by the object’s movement provides us the valuable shading correlation along the sequence, which is critical to resolve the surface normal and albedo without any ambiguity. It resembles the photometric stereo. But instead of controlling the light when imaging a static object, we are allowed to move the object under general natural lighting. This kind of cue has been exploited in multi-view photometric stereo [35] and shape from video [75, 127, 159]. However, they have the environmental lighting constrained to be calibrated directional light and the object is experiencing turntable motion or the motion is assumed be calibrated beforehand. On the contrary our approach works under natural lighting with the object experiencing arbitrary motion using a single RGBD sensor, which makes our method more widely used in everyday environment.

Given the captured RGB-D sequence, first we try to align the RGB-D sequence and find the correspondences among the images using a novel robust matching technique. Then the environmental lighting is estimated using the intensity ratios of the aligned sequence, which effectively factors out the impact of varying albedo. Finally, we formulate an Expectation-

Maximization framework in which the surface normal and its albedo map can be calculated robustly, in the presence of some non-Lambertian reflection or cast shadow. A detailed surface mesh is obtained after integration of the initial depth map with the estimated normal map.

The main contribution is that we utilize the dynamic photometric information along the sequence to recover the surface details beyond the resolution of current depth sensors. Compared to previous depth enhancement schemes that use the color information, our method, to the best of our knowledge, is the least restrictive. It allows arbitrary surface albedo, does not require controlled or calibrated lighting or turntable capture. To achieve these, we make two technical contributions. The first is a novel image registration scheme that is robust to lighting variations and the second is an EM optimization scheme to produce per-pixel normal and albedo map under general lighting.

### 3.1 Pipeline

An overview of our depth enhancement and albedo recovery framework is shown in Fig. 3.1.

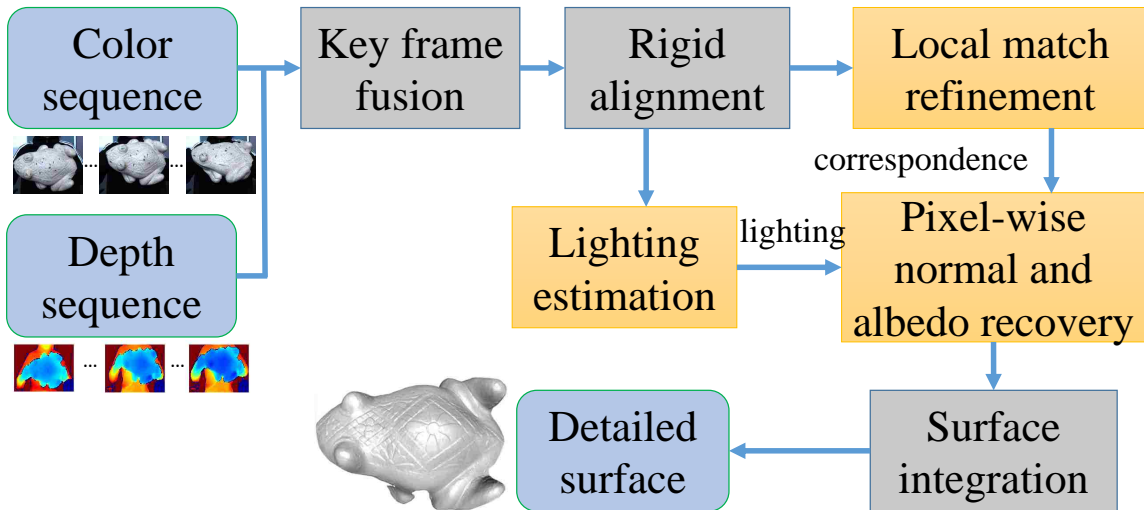


Figure 3.1: System Pipeline.

First, instead of using all those frames of the sequence which is redundant and computationally too expensive, we fuse every  $M = 20$  to generate  $N$  key frame depth maps from the RGB-D sequence via KinectFusion [101]. They are smoother and more accurate than the raw depth maps. The extrinsic parameters between these key frames are computed and refined with bundle adjustment afterwards. Next, a robust pixel matching strategy is

proposed to find correspondences across the key frame images dealing with possible mis-alignments caused by imprecise initial depth maps or image distortion. We call this procedure Local Match Refinement, as the correspondences are locally searched and refined starting from the correspondences achieved by warping the images guided by the initial depth maps. For lighting estimation, we utilize the entire sequence to make the estimation more robust. Finally, given the computed lighting and correspondences along the sequence, we recover the surface normal and albedo image under our robust EM framework. To the end, the recovered normal can be integrated with any key frame depth map to generate a surface model with much more structural details revealed.

## 3.2 Preliminary Theory

Before introducing our proposed approach in detail, we demonstrate some basic theory and derivations in this section.

While environmental lighting can be arbitrarily complex, the appearance of a diffuse object can be described by a low dimensional model [113]. Under this assumption, the shading function  $s$  for Lambertian reflectance can be modeled as a quadratic function of the surface normal with  $A$ ,  $b$ ,  $c$  represented as the lighting parameters.

$$s(\mathbf{n}) = \mathbf{n}^T A \mathbf{n} + b^T \mathbf{n} + c \quad (3.1)$$

Generally the captured image is generated by multiplying the shading function with surface albedo  $\rho(p)$

$$I(p) = \rho(p) s(\mathbf{n}_p) \quad (3.2)$$

Given a single image as observation, as for each pixel we have three equations with five unknowns to be estimated, it may not be feasible to recover the surface normal and albedo faithfully even though we could suppose the lighting parameters have been determined beforehand. Photometric stereo, with more lighting variations, is a typical solution to resolve the ambiguity. Mathematically, the surface normal and its albedo can be computed by minimizing the following objective function that is formulated for each pixel independently under various lighting conditions.  $A_k, b_k, c_k$  is one set of lighting parameters. No smoothness or albedo regularization is needed here.

$$E(\mathbf{n}, \rho) = \sum_k (\rho(p) (\mathbf{n}_p^T A_k \mathbf{n}_p + b_k^T \mathbf{n}_p + c_k) - I_k(p))^2 \quad (3.3)$$

The underlying principle of our enhancement method is based on the above photometric stereo theory, but we do not need to set the object to be static and manually change

the lighting conditions; instead we have captured the RGB-D sequence of the object under arbitrary motion in uncalibrated natural illumination. In this case, the lighting variations induced by object motion resembles the classic photometric stereo in some way. We describe the derivations in the following.

Suppose we set the first frame as the reference frame, and also we can find the correspondences for pixels along the sequence. For example, for pixel  $p$  in the reference frame, its correspondence in frame  $k$  is  $W(p)$ . The appearance of the pixel  $W(p)$  is generated as,

$$\begin{aligned} I_k(W(p)) &= \rho(p) \left( (R_k \mathbf{n}_p)^T A (R_k \mathbf{n}_p) + b^T (R_k \mathbf{n}_p) + c \right) \\ &= \rho(p) \left( \mathbf{n}_p^T (R_k^T A R_k) \mathbf{n}_p + (b^T R_k) \mathbf{n}_p + c \right) \end{aligned} \quad (3.4)$$

where  $\rho$  is the albedo for pixel  $p$  which equals to the albedo of pixel  $W(p)$  and  $\mathbf{n}_p$  is surface normal under reference frame coordinate.  $R_k$  is the rotation from the reference frame to frame  $k$ . Therefore, the surface normal for the corresponding pixel  $W(p)$  in image  $I_k$  can be represented as  $R_k \mathbf{n}_p$ . As demonstrated in the above equation, the rotation  $R_k$  can be extracted and applied to the lighting vectors, from which we will get the lightings for frame  $k$  as  $R_k^T A R_k$ ,  $b^T R_k$  and  $c$ .

Therefore as similar to the photometric stereo, the changes of lighting induced by the object motion provide valuable cues to recover the surface normal and its albedo. We illustrate this in Fig. 3.2 with the energy plot showing that with more and more images under diverse rotation variations included in the energy function, we are able to resolve the local ambiguity and converge to the optimal solution without relying on any smoothness regularizations.

### 3.3 Approach

There are four major parts in the pipeline as shown in Fig. 3.1, including robust pixel matching among the images, lighting estimation, and normal and albedo recovery. We will describe them successively in details in the following sections.

First of all, the key frame depth maps  $D_1 \sim D_N$  are obtained via depth fusion with the corresponding color images denoted as  $I_1 \sim I_N$ .

#### 3.3.1 Robust Pixel Matching

##### Rigid alignment

First, the global rigid transformation between key frames are calculated by detecting SIFT or ORB features followed by feature matching. These extrinsic parameters are further

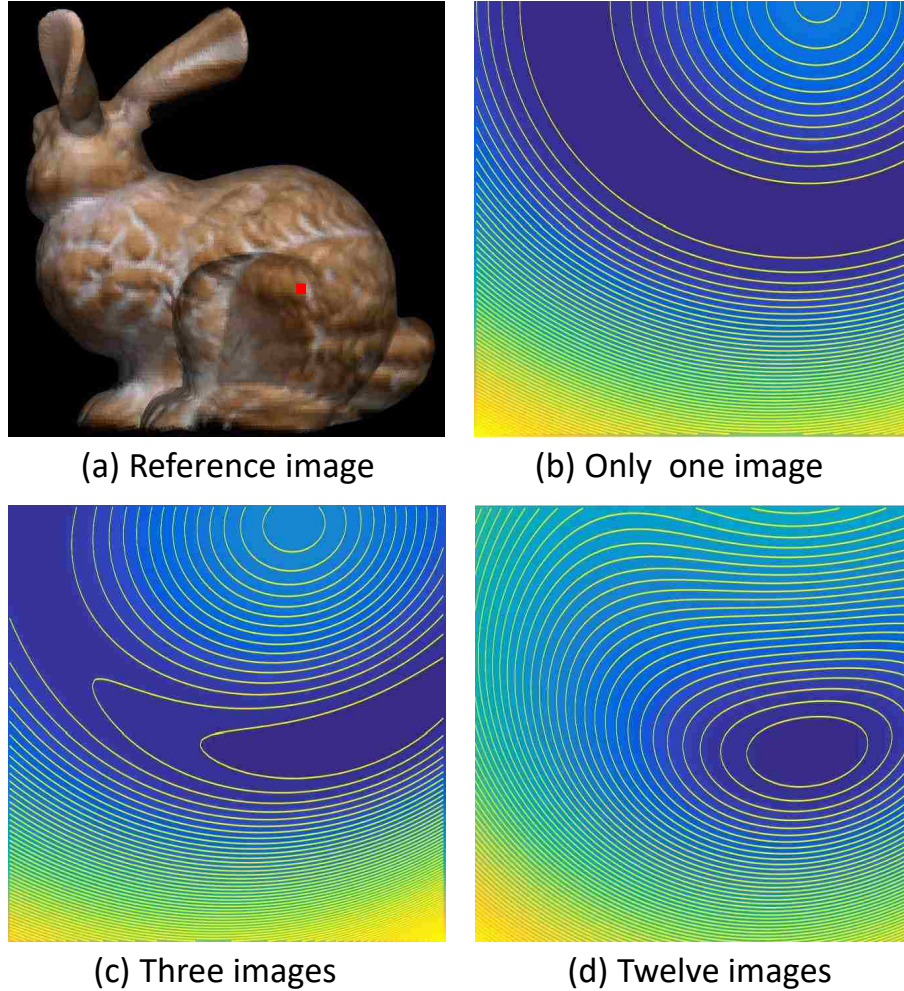


Figure 3.2: Diverse rotation variations resolve local ambiguity. (a) shows a sampled image of the object with a reference pixel marked as red. In (b),(c) and (d) we plot the energy map for this reference pixel with x-axis and y-axis representing the two degree of freedom of a surface normal. The cooler color in these figures corresponds to smaller energy value. As we can see, given a single image the solution lies in a large band as shown in (b). With three images the normal converges better as shown in (c). And finally we will be able to find the optimal surface normal for the reference pixel if we have got enough images under rotation variations as shown in (d).

refined with bundle adjustment and finally we get the rotation  $R_1 \sim R_N$  and translation matrix  $T_1 \sim T_N$  with respect to a global coordinate for each key frame from which we can warp any key frame to other frames.

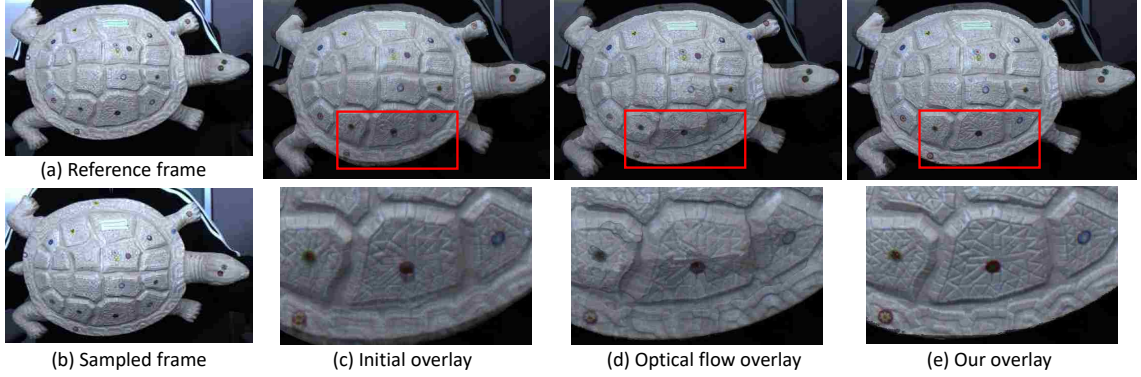


Figure 3.3: Demonstration of correspondence matching. (a) is the reference frame and (b) is one sampled key frame. Image(b) is warped to the reference frame with current transformation, and (c) displays the warped image overlaid with image(a). (d) shows the overlaid result using the flow map computed from warped image and the reference image [19]. (e) is the overlaid image after applying our proposed lighting insensitive robust matching.

### Lighting insensitive Local Match Refinement

These key frames can be warped into any reference frame given the current transformation. However misalignments still exist after bundle adjustment as shown in Fig. 3.3(c), which is caused by the imprecise depth maps, image distortion and the imperfect synchronization between the captured color and depth sequence. Optical flow is often used as a solution to find correspondences between two images. Considering that the misalignment may be severe, we have tried to use a large displacement optical flow computation approach [19] to find the correspondences between the warped image and the reference image. However, since the consistency assumption is not maintained in our case, the alignment has got even worse in some part with great illumination changes as displayed in Fig. 3.3(d).

Our matching approach is implemented on every reference frame and we find correspondences from the reference frame to other key frame images. Suppose we have the reference depth map and corresponding color image denoted as  $D_{ref}$  and  $I_{ref}$  respectively. For each pixel  $p = (u, v)$  in  $I_{ref}$ , its current corresponding pixel  $q$  after bundle adjustment in image  $I_k$  is computed as,

$$\lambda \begin{bmatrix} q \\ 1 \end{bmatrix} = K \left( R_k (R_{ref}^{-1} (K^{-1} \begin{bmatrix} u \\ v \\ D_{ref}(u, v) \end{bmatrix} - T_{ref})) + T_k \right) \quad (3.5)$$

In the above equation,  $K$  is the camera intrinsic matrix.  $R_{ref}$  and  $T_{ref}$  are the rotation and translation matrix that transform the 3D point from world coordinate to the reference frame. Similarly,  $R_k$  and  $T_k$  are the rotation and translation matrix for frame  $k$ .

The corresponding pixel  $p$  in  $I_{\text{ref}}$  and  $q$  in  $I_k$  may not be the correct correspondence because of the misalignment. Therefore, we implement a local search strategy to find its best matching pixel in  $I_k$ .

For each pixel  $p$  in  $I_{\text{ref}}$ , we set a searching region around it and find its best match in  $I_k$  via NCC (Normalized Cross Correlation). However, the intensity consistency is not preserved as the object is subject to arbitrary movements. This makes the original NCC not suitable for matching in this case. To deal with this problem, we apply chromacity normalization in the color image to eliminate the effect of lighting variations [39] and use the normalized images for matching. For each pixel  $p$ , its appearance is generated as,

$$I_{ch}(p) = \rho_{ch}(p)s(\mathbf{n}_p) \quad ch \in \{R, G, B\}, \quad (3.6)$$

in which  $s(p)$  is the shading function that accounts for the lighting or normal variation.

So the chromacity normalization is implemented as,

$$I_{ch}^{cn}(p) = \frac{I_{ch}(p)}{I_R(p) + I_G(p) + I_B(p)} \quad ch \in \{R, G, B\}, \quad (3.7)$$

After the above normalization, NCC can then be applied for the matching which will be insensitive to the photometric inconsistency induced by lighting factor. For the NCC computation, we perform it separately on each channel of the color image.

Specially, the color image  $I_{\text{ref}}$  is warped to the color frame  $I_k$  under the guidance of  $D_{\text{ref}}$  and we get the warped color image  $I_{\text{ref}_k}$ . The NCC patch matching is implemented in  $I_{\text{ref}_k}$  with  $I_k$  instead of using  $I_{\text{ref}}$  directly. Since  $I_k$  and  $I_{\text{ref}_k}$  are in the same viewpoint, the fattening effect of NCC is successfully avoided.

Although for each pixel in  $I_{\text{ref}}$  (or  $I_{\text{ref}_k}$ ) we can find the corresponding pixel in  $I_k$  that has the largest matching score, we cannot guarantee they are always the correct correspondence. To tackle this problem, we only keep the pixels that are reliable and use these pixels as control vertices to deform all the other pixels to find their correct correspondences.

Our criteria of reliable matches is that, 1) the largest matching score should be larger than  $\text{thres}_S$ ; 2) the difference between the largest score and second largest score of local peaks should be larger than  $\text{thres}_\Delta$ . If these principles are maintained, the pixel in the searching region that has the largest score is chosen as the correspondence.  $\text{thres}_S$  is set to be 0.75 and  $\text{thres}_\Delta$  is 0.05 in our experiments.

Next we use these reliable matches as control vertices to deform the image  $I_{\text{ref}_k}$  so that it has an optimal match with  $I_k$ . As for each control vertices  $o_l$  in  $I_{\text{ref}_k}$ , the deformation function is defined as

$$f(o_l) = o_l + \Delta_l, \quad (3.8)$$



where  $\Delta_l$  is the motion vector between the optimal correspondence and its initial correspondence in  $I_k$ .

For other pixels the deformation is formulated via bilinear interpolation with control vertices [163],

$$f(u) = u + \sum_l (\theta_l^u \Delta_l) \quad (3.9)$$

The interpolation coefficients  $\theta_l^u$  is set according to the distance to control vertices and only neighboring vertices will affect the deformation.

Finally, our objective function is defined to maintain the photo consistency of the two normalized images.

$$E(\Delta) = \sum_p \left( I_{\text{ref}_k}^{cn}(f(p, \Delta)) - I_k^{cn}(p) \right) + \lambda \sum_l \|\Delta_l - \hat{\Delta}_l\|^2 \quad (3.10)$$

where  $\hat{\Delta}$  is the initial deformation vector for the control vertices between the current optimal correspondence obtained from matching and its initial correspondence.  $\lambda$  is the control weight set to be 10. Since we have good initials  $\hat{\Delta}$ , the optimization will converge quite fast.

Some matching results are demonstrated in Fig. 3.3(e).

### 3.3.2 Lighting estimation

In this section, we demonstrate how to compute the lighting parameters  $A$ ,  $b$ ,  $c$  for each reference frame. Since The unknown albedo poses challenges for lighting estimation, there are some methods that cluster the image into different parts and use the mean value as their albedos. The lighting and albedo is estimated in an iterative way. Instead of trying to resolve the ambiguity from a single frame, we employ the aligned color sequence and depth maps for robust lighting estimation, eliminating the need to make prior assumptions about albedo.

With the aligned color images we can compute the ratio images with respect to the reference image, from which the albedo will get canceled out. In details, for each pixel  $p$  in  $I_{\text{ref}}$ , suppose its corresponding pixel in  $I_k$  is denoted as  $q$ , then the ratio value is computed as,

$$\begin{aligned} \frac{I_k(q)}{I_{\text{ref}}(p)} &= \frac{\rho(q)(\mathbf{n}_q^T A \mathbf{n}_q + b^T \mathbf{n}_q + c)}{\rho(p)(\mathbf{n}_p^T A \mathbf{n}_p + b^T \mathbf{n}_p + c)} \\ &= \frac{\mathbf{n}_q^T A \mathbf{n}_q + b^T \mathbf{n}_q + c}{\mathbf{n}_p^T A \mathbf{n}_p + b^T \mathbf{n}_p + c} \end{aligned} \quad (3.11)$$

Therefore, the environmental lighting can be achieved from the following minimization,

$$\arg \min_{A,b,c} \sum_k \sum_{p \in I_{\text{ref}}} \gamma_p \left( \frac{\mathbf{n}_q^T A \mathbf{n}_q + b^T \mathbf{n}_q + c}{\mathbf{n}_p^T A \mathbf{n}_p + b^T \mathbf{n}_p + c} - \frac{I_k(q)}{I_{\text{ref}}(p)} \right)^2 \quad (3.12)$$

The normal  $\mathbf{n}$  are approximated using the initial normals computed with the key frame depth maps. The weighting term  $\gamma_p$  is set to prevent using pixels with intensity that are too dark or too bright which might be caused by cast shadow or specularities. Besides, we also ignore pixels with great image gradients that are sensitive to misalignments. The lighting vectors can get updated iteratively after the albedo recovery with refined normal maps.

### 3.3.3 Normal and albedo recovery

With the key frame color images all aligned into the reference image ( $I_1^W \sim I_N^W$ ), and the estimated environmental lighting ( $A, b, c$ ), we are ready to recover the surface normal and its albedo. We have the object rotation matrix  $R_1 \sim R_N$  for each frame with respect to the reference frame. Then for each pixel  $p$  in the reference frame, our goal is to find the optimal albedo  $\rho(p)$  and normal  $\mathbf{n}(p)$  conforming the pixel observations  $\mathbf{I}(p) = \{I_k^W(p)\}_{k=1}^N$ . *We drop the index of pixel locations for simplicity in the following description.* The objective function can be defined as:

$$E(\mathbf{n}, \rho | \mathbf{I}) = \sum_k (s_k(\mathbf{n})\rho - I_k^W)^2, \quad (3.13)$$

$$s_k(\mathbf{n}) = \mathbf{n}^T R_k^T A R_k \mathbf{n} + b^T R_k \mathbf{n} + c \quad (3.14)$$

The surface normal and albedo can be estimated from minimization of the above function. However, the outliers have not been taken into consideration. They will affect the result if the observations violate the Lambertian assumption. To deal with these outliers, we introduce a set of hidden states  $H_k = \{0, 1\}$  indicating whether the observation is actually generated by the Lambertian model. An expectation-maximization (EM) algorithm is developed to solve the problem. While our formulation is inspired by [150], we extend it from its original directional light assumption to general lighting. More specifically, we denote the parameters to be estimated as  $\Omega = \{\mathbf{n}, \rho, \sigma, \alpha\}$  and the observation probability conditioned on parameters  $\Omega$  is given as,

$$P(I_k^W | \Omega) = \alpha \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(s_k(\mathbf{n})\rho - I_k^W)^2}{2\sigma^2}\right) + (1 - \alpha) \frac{1}{C} \quad (3.15)$$

$P(H_k = 1) = \alpha$  is the prior probability of  $H_k$  indicating the proportion of observations generated by the Lambertian model.  $\frac{1}{C}$  is the probability as being an outlier which is assumed to be uniform distribution and we set  $C$  to be 10 in our implementation.

The posterior probability of the hidden variable  $H_k$  is updated in every E-step using the following equation given the computed parameters  $\Omega'$  in current iteration and the observation  $I_k^W$ ,

$$\begin{aligned}\omega_k &= P(H_k = 1 | I_k^W, \Omega') \\ &= \frac{\alpha \exp\left(-\frac{(s_k(\mathbf{n})\rho - I_k^W)^2}{2\sigma^2}\right)}{\alpha \exp\left(-\frac{(s_k(\mathbf{n})\rho - I_k^W)^2}{2\sigma^2}\right) + \frac{1-\alpha}{C}}\end{aligned}\quad (3.16)$$

Next, in the following M-step, we maximize the complete-data log-likelihood given the marginal distribution  $H_k$  obtained from the E-step.

$$\begin{aligned}Q(\Omega | \Omega') &= \sum_k \log P(I_k^W, H_k = 1 | \Omega) \omega_k \\ &+ \sum_k \log P(I_k^W, H_k = 0 | \Omega) (1 - \omega_k) \\ &= \sum_k \log\left(\frac{\alpha}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(s_k(\mathbf{n})\rho - I_k^W)^2}{2\sigma^2}\right)\right) \omega_k \\ &+ \sum_k \log\left(\frac{1-\alpha}{C}\right) (1 - \omega_k)\end{aligned}\quad (3.17)$$

To maximize the above function, we set the first derivative of  $Q$  with respect to  $\alpha$ ,  $\sigma$  and  $\rho$  equal to zero. In this way, the updating rules for these parameters are obtained,

$$\begin{aligned}\alpha &= \frac{1}{N} \sum_k \omega_k \\ \sigma &= \sqrt{\frac{\sum_k (s_k(\mathbf{n})\rho - I_k^W)^2 \omega_k}{\sum_k \omega_k}} \\ \rho &= \frac{1}{\sum_k s_k(\mathbf{n})^2 \omega_k} \sum_k s_k(\mathbf{n}) \omega_k I_k^W\end{aligned}\quad (3.18)$$

Since the function  $Q$  is nonlinear to surface normal  $\mathbf{n}$ , the updated normal is achieved by fixing other parameters and solving the following energy minimization.

$$\arg \min_{\mathbf{n}} \sum_k ((\mathbf{n}^T R_k^T A R_k \mathbf{n} + b^T R_k \mathbf{n} + c)\rho - I_k^W)^2 \omega_k \quad (3.19)$$

The above EM iterative optimization process is performed until no further improvement on the recovered normal and albedo. The initial parameter of  $\alpha$  and  $\sigma$  is set to be 0.75 and 0.05 respectively for all the datasets used in this chapter.

Finally, the recovered normal is integrated with the depth map to get enhanced surface geometry with structural details [160].

### 3.3.4 Implementation details

As a preprocessing step, the object is first segmented from the image by integrating both color and depth information into GrabCut [117] framework. We manually masked the first frame with the rest of frames segmented automatically.

We implement most parts of our framework in Matlab and it takes us approximately 820s to process a dataset with 500 ~ 600 frames. Considering that the normal and albedo is computed in pixel-wise manner, the running time could be reduced further with parallel computation.

## 3.4 Experimental results

In the experiments, we validate our method on synthetic and real datasets with quantitative and qualitative evaluation.

### 3.4.1 Synthetic datasets

In this section we perform quantitative evaluations of our method on several synthetic models. First given the 3D model, twenty images together with their corresponding depth maps are rendered under natural illumination. The rendered ground-truth depth maps are over smoothed to filter out the structural details. Those smoothed depth maps and rendered color images are taken as input for our method. We have compared our method with a shading refinement approach [105] from a single RGB-D image which has achieved good performance on depth refinement.

Fig. 3.4 shows the comparison results of our recovered normal map and surface. For each model, the first column is the reference color map and over smoothed mesh (displayed as normal map). These are the input for the shading refinement method [105]. The output of the shading refinement method is displayed in the second column. The texture copy artifacts are caused by imperfect separation of albedo and shading layers. In comparison, the surface normal can be recovered successfully with our pixel-wise recovery method with quite small error shown in the third column. The albedo map computed from our method together with its error map is demonstrated in the last column.

We display the quantitative results in Table 3.1 showing the mean error of computed normal maps, extracted albedo images and also the enhanced depth maps. As we can see, the error of our computed normal map is quite small as compared with the shading refinement approach. For the Armadillo and Lion model, the normal error of the shading refinement approach becomes even larger than the initial over smoothed normal as caused

by the texture copy problem. We have also performed evaluation on our recovered albedo image with the mean error shown in the Table 3.1. We have normalized the images into 0 1. Since we cannot resolve the scale ambiguity of the computed albedo and shading image, we have calculated a scale factor with six randomly selected pixels in the ground-truth albedo image, which are divided by the values of corresponding pixels in our recovered albedo image.

Table 3.1: Quantitative Evaluation.

Model	<b>Bunny</b>	<b>Armadillo</b>	<b>Lion</b>
Initial depth error	1.2150	1.1129	1.3927
Initial normal error(degree)	8.4236	11.7554	14.3837
Shading normal error(degree)	6.7591	14.6649	23.5301
Our normal error(degree)	<b>1.3485</b>	<b>4.5130</b>	<b>6.1094</b>
Our albedo error	0.0127	0.0384	0.0295
Our depth error	0.2506	0.3129	0.2927

Fig. 3.5 is shown to demonstrate the effectiveness of our EM framework for robust normal recovery in the presence of outliers. We have picked four out of those twenty images randomly and added the salt and pepper noise with 0.50 density. It means the abrupt noise will affect approximately fifty percent of the image pixels. As we can see from the first two columns, the recovered normal map without EM optimization is noisy (the mean error is 8.37 degree), while we can achieve much better performance after applying our EM method, which is shown in the last two columns and the mean error decreased to 1.49 degree.

### 3.4.2 Real datasets

We have captured the datasets of real objects using the depth sensor of Kinect V2 with resolution of  $512 \times 424$  and a PointGrey color camera with resolution of  $1920 \times 1080$ . Several objects are captured, namely the Frog, Shoe, Backpack, Turtle, etc. We will demonstrate the comparison results of our recovered surface normal and albedo with some state-of-the-art approaches in the following.

#### Surface normal and geometry recovery

Fig. 3.6 shows the comparison results of a Frog, Shoe and Chinese Fan model. We have made comparisons with a shading refinement approach [105] and a depth super-resolution

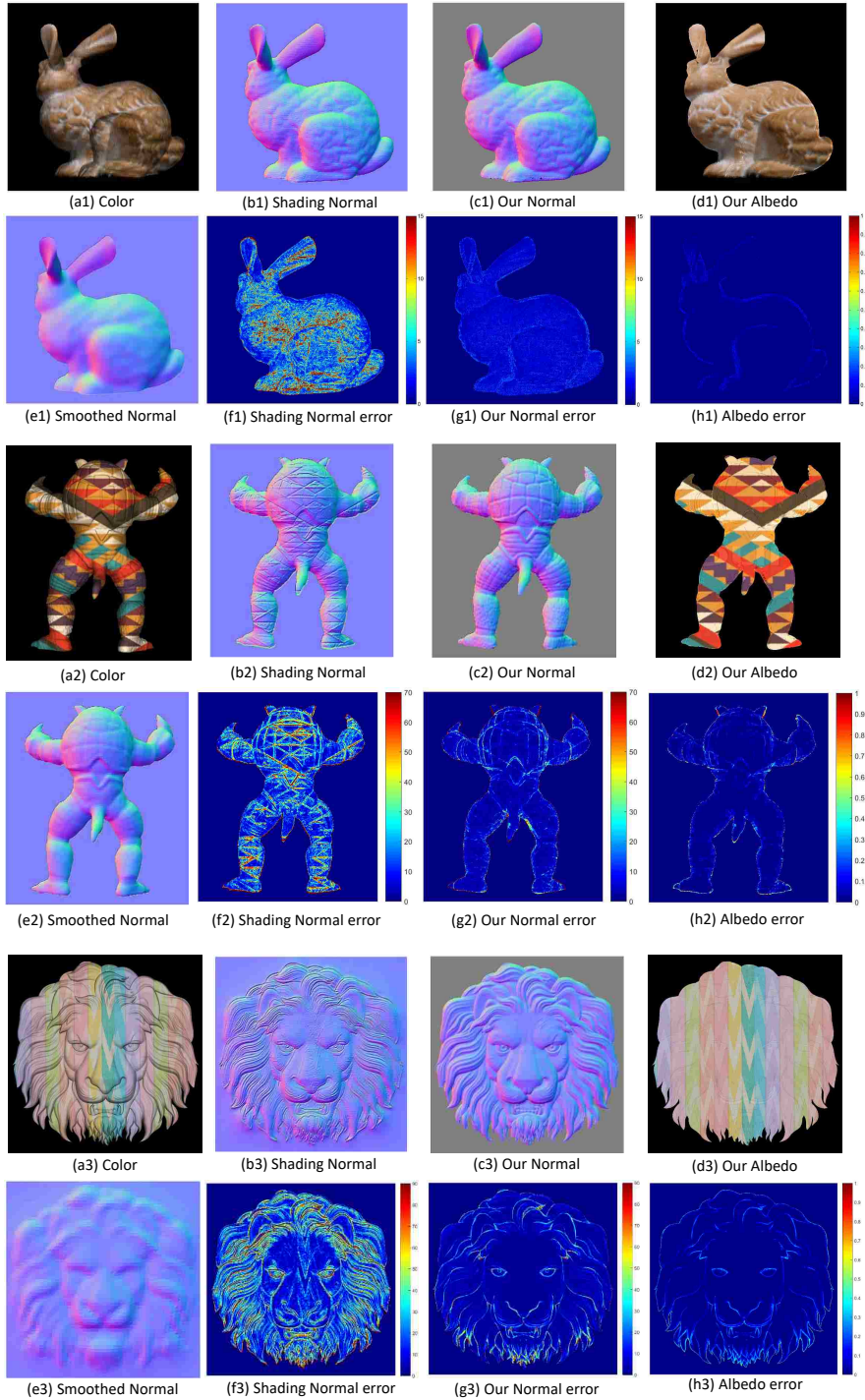


Figure 3.4: Results on synthetic models. (a1-a3) is the rendered color image of the reference frame; (e1-e3) shows the normal map of the ground-truth mesh after over smoothing; (b1-b3) is the normal map computed after applying shading refinement on the reference frame with its error map displayed in (f1-f3); (c1-c3) and (g1-g3) are the normal map and its corresponding error map achieved by our method. Our recovered albedo map and its error map is also demonstrated in (d1-d3) and (h1-h3) respectively.

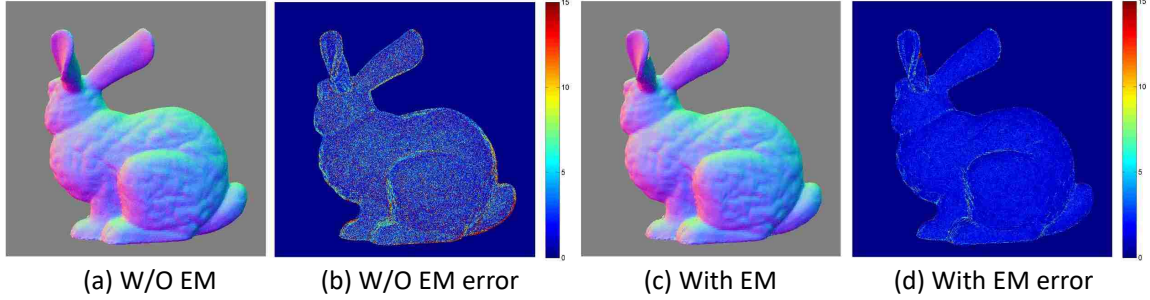


Figure 3.5: Results when adding salt and pepper noise. (a) shows the computed normal map without our EM framework and (b) is its error map; The normal and error map after applying our EM optimization are shown in (c) and (d) respectively.

approach [47] which deals with depth super-resolution and shading refinement problems simultaneously in an unified framework. The color images shown in Fig. 3.6(a1)(a2)(a3) and their corresponding depth images are taken as input for those two methods. As we can see from Fig. 3.6(b1)(b2)(b3) the surface has got over-smoothed after the fusion [101] and the small surface details cannot get revealed as restricted by the resolution and accuracy of the Kinect depth sensor. The shading refinement approach [105] is able to recover some surface details, but some textures are hallucinated as geometry details as well (Fig. 3.6(c1)(c2) and (c3)). Fig. 3.6(d1)(d2) and (d3) displays the results of depth super-resolution [47] for which the colorful textures have caused unpleasant artifacts on the recovered surface as they have also assumed that the surface albedo is piecewise constant. Fig. 3.6(e1,e2,e3) and Fig. 3.6(f1,f2,f3) displays our final results of recovered meshes and surface normals. For the Frog and Shoe model, the small surface details have got successfully extracted and revealed in our results without affected by the textures. For the Chinese Fan model, it contains some concave parts which could cause cast shadow on the images. Although it does not have much small geometric details, the pleats become more sharp in our recovered mesh as compared to the fusion results with smooth surface on other parts as it should be.

Fig. 3.7 shows the comparison results of a very colorful Backpack and Turtle model and in this figure we demonstrate the effectiveness and importance of our local match refinement step. For the Backpack, it actually experiences non-rigid deformation during the movement and therefore we only consider the front part of the backpack which is mostly rigid. We have marked some colorful patterns on the Turtle surface to make the texture more complex to show the superior performance of our pixel-wise recovery method.

Similar to the Frog and Shoe model, the shading refinement approach [105] suffers from the texture-copy problem as shown in (Fig. 3.7(b1) and (b2)). Fig. 3.7(c1) and (c2) displays the super-resolution results [47], for which the surface details have not got re-

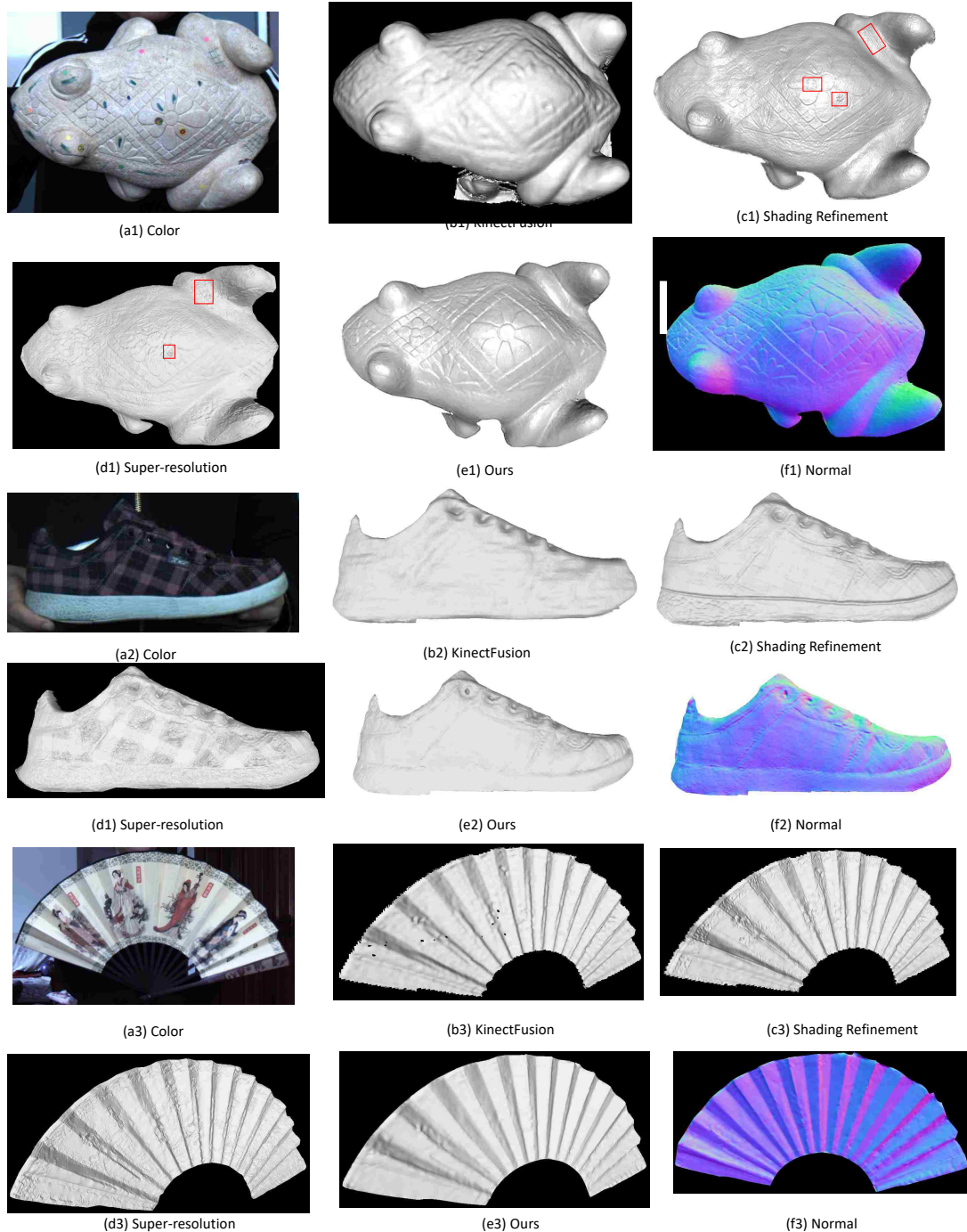


Figure 3.6: Comparison results on Frog, Shoe and Chinese Fan model. (a1) (a2) and (a3) are the reference color images of Frog, Shoe and Chinese Fan respectively. The outputs from KinectFusion are shown in (b1) (b2) and (b3). The results computed by shading refinement method [105] are displayed in (c1) (c2) and (c3). (d1) (d2) and (d3) are the meshes computed by depth super-resolution method [47]. Finally, (e1) (e2) and (e3) are the output meshes from our approach with their corresponding normal maps displayed in (f1) (f2) and (f3).



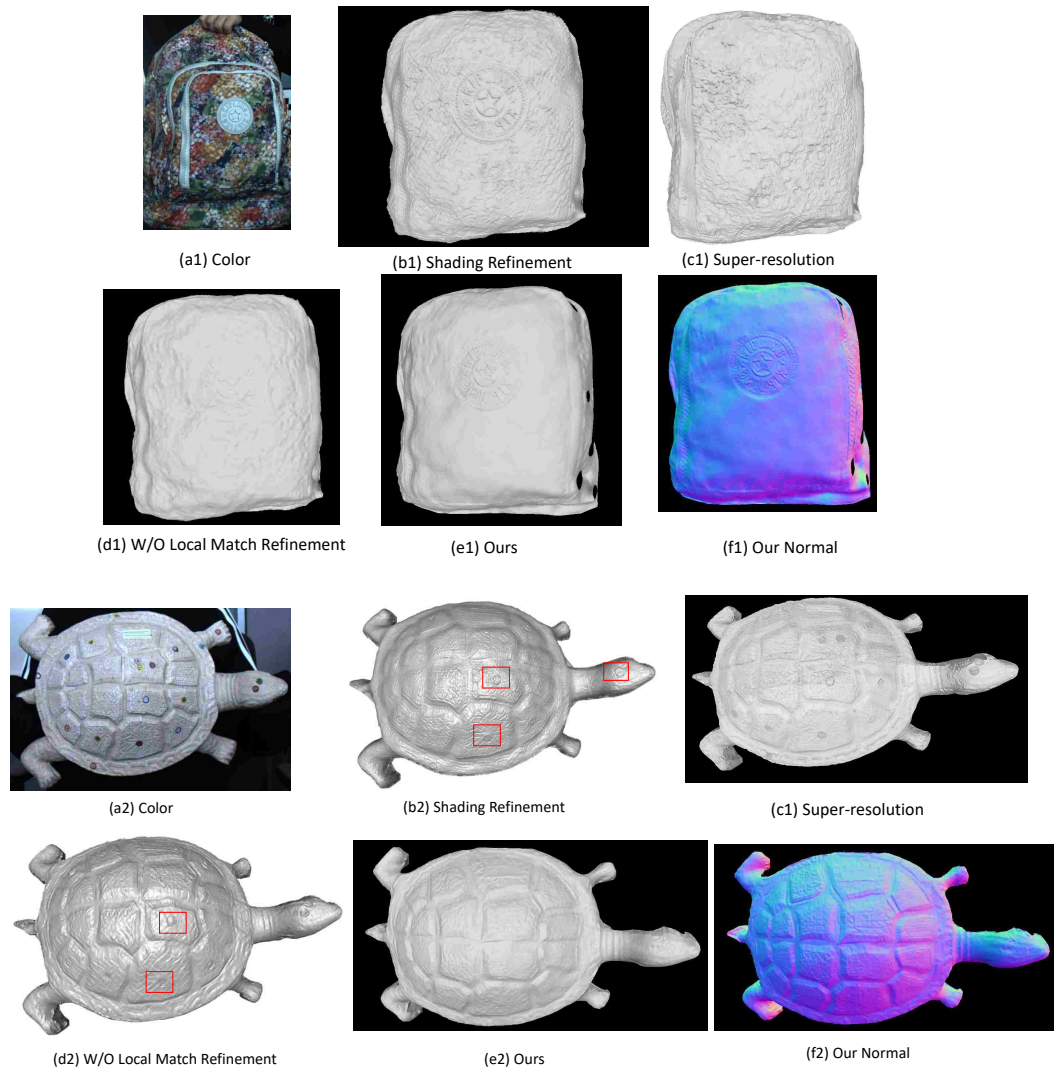


Figure 3.7: Comparison results on Backpack and Turtle model. (a1) and (a2) are the reference color images of Backpack and Turtle respectively. The output from shading refinement method [105] is shown in (b1) and (b2). The results computed by depth super-resolution [47] are displayed in (c1) and (c2). (d1) and (d2) are the meshes acquired using our method but without applying our locally robust matching procedure. (e1) and (e2) are the output meshes of our method and the corresponding normal maps are displayed in (f1) and (f2).

covered clearly. Fig. 3.7(d1) and (d2) shows our results without applying our local match refinement step. The uneven surface in some part is caused by misalignment. We are able to eliminate the artifacts after our locally matching step with real geometric details revealed as shown in Fig. 3.7(e1) and (e2).

To further validate our robustness against texture copy problem, the results of a Book cover with extremely rich textures are demonstrated in Fig. 3.8. As displayed in Fig. 3.8(d) the textures have been successfully factored out from the image with our approach and the recovered model keeps as a planar surface after the enhancement. In comparison, the result from shading refinement method(Fig. 3.8(b)) and depth super-resolution approach(Fig. 3.8(c)) are severely affected by the texture copy effect with lots of fake geometric details appeared.

### **Intrinsic Image Decomposition**

In order to show the performance of our method in albedo recovery, we have also made some comparisons with two state-of-the-art intrinsic image decomposition approaches [21, 60] as displayed in Fig. 3.9. For these two compared methods, they take the RGB-D images of the reference frame as input, as displayed in the first column. The second column shows the result from Chen [21], for which the shading image is over smoothed with the geometry details decomposed into albedo map incorrectly. The method from Jeon [60] has better results on recovered shading images for the Turtle and Frog models as displayed in the third column. However, some textures still stay at the shading image especially for the Shoe and Backpack. In comparison, with our pixel-wise albedo computation method, we are able to recover a much sharper albedo map and the texture copy effect in the geometry is barely noticeable.

## **3.5 Conclusion**

In this chapter, I have presented a novel approach to recover surface details and its albedo map from an RGB-D video sequence. The object is experiencing casual motion from which the induced illumination variation provides us the cue to recover the surface normal and its albedo as well. A robust lighting insensitive local match strategy is proposed to establish correct correspondences from reference frame to other frames. Then, the environmental lighting is estimated by exploiting the whole sequence to get rid of the effect of varying textures. Finally, the surface normal and its albedo is calculated robustly with our EM framework. We have validated our method on both synthetic and real datasets and compared with some state-of-the-art surface refinement and intrinsic decomposition methods.

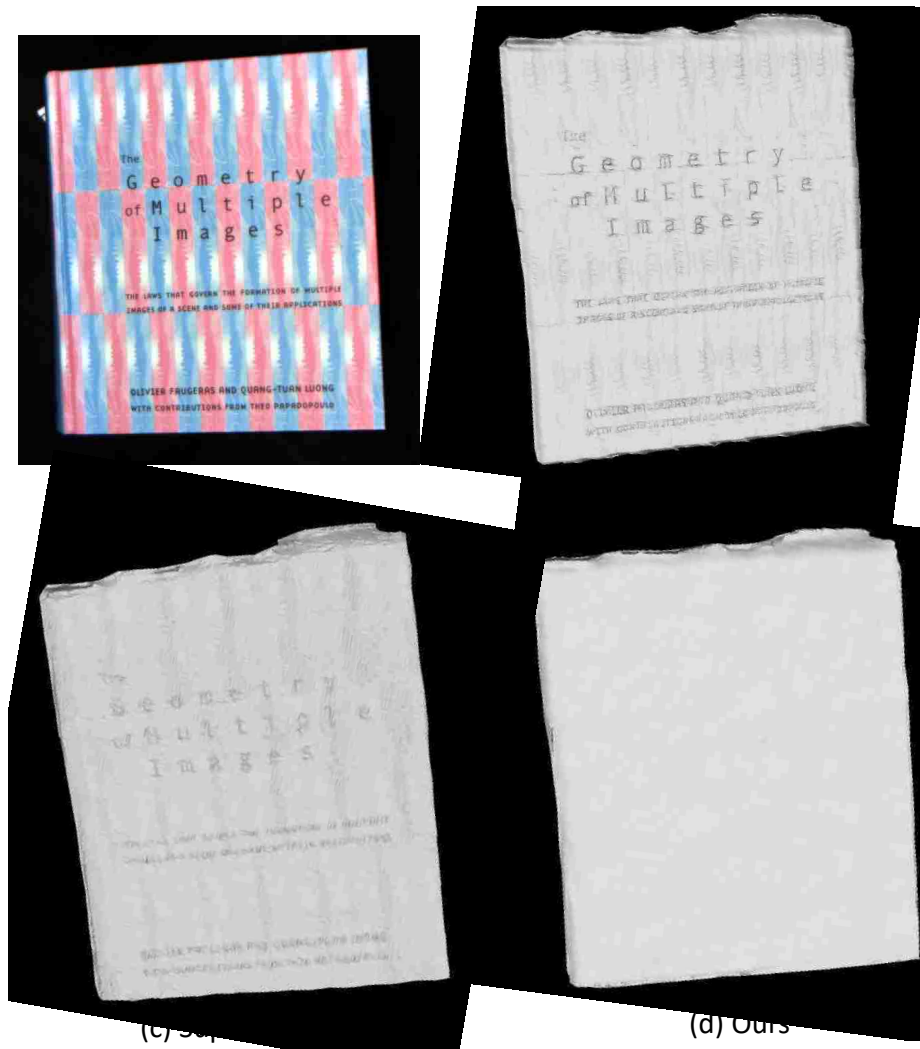


Figure 3.8: Results on Book model. (a) is the reference color image. (b) shows the refined mesh with shading refinement method [105]. (c) shows the super-resolution results [47]. The recovered mesh surface from our method is displayed in (d).

As demonstrated in the experiments, we have achieved good performance on both surface details recovery and intrinsic decomposition.

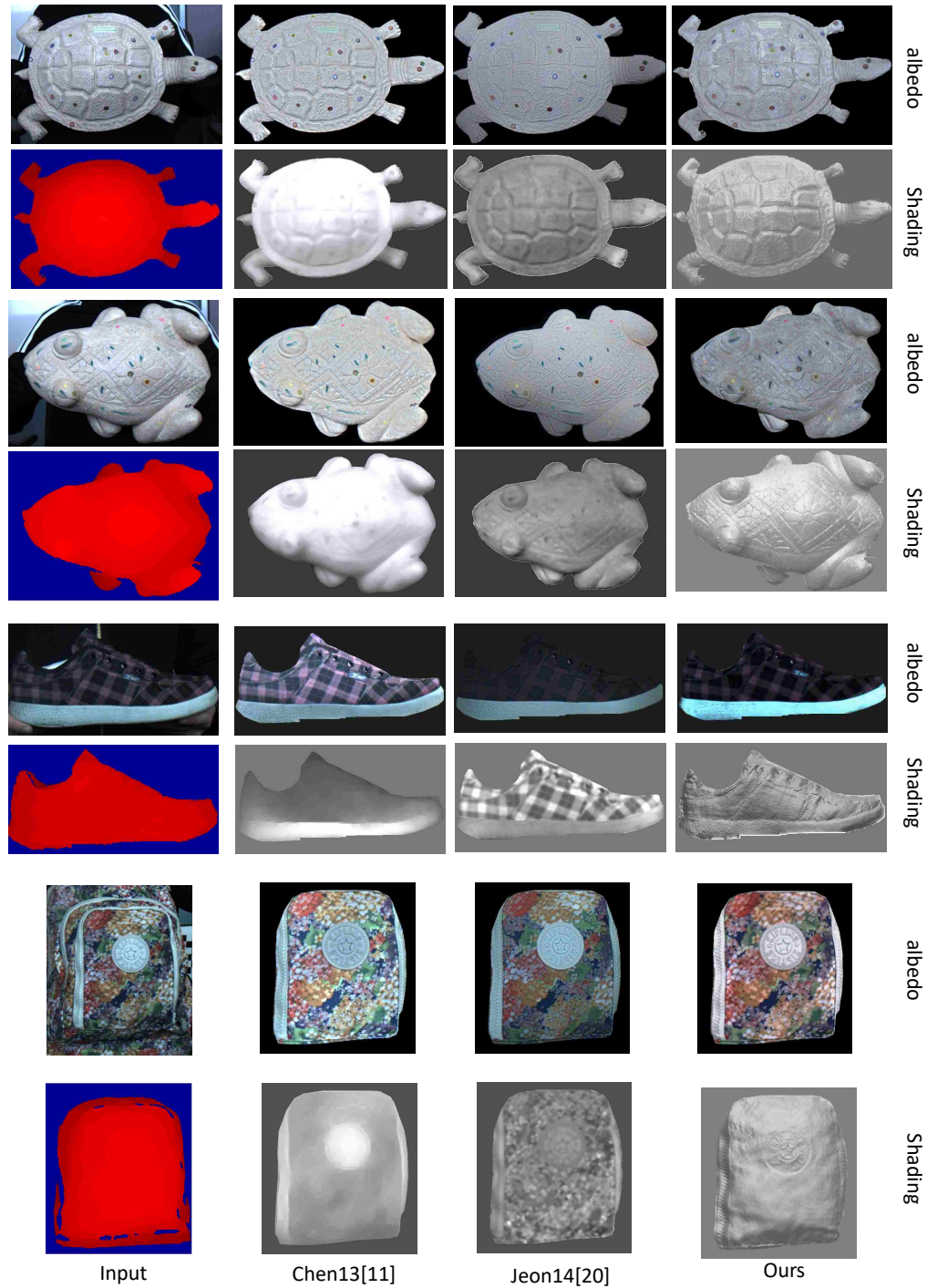


Figure 3.9: Comparison results on albedo recovery or intrinsic decomposition of the Turtle, Frog, Shoe and Backpack model. The first column is the input color image with its corresponding depth map. The second column shows the result of Chen [21]. The third column is the decomposed albedo and shading images from method in [60]. Finally, the last column demonstrates the result achieved by our method.

## Chapter 4

# High-speed Depth Stream Generation from a Hybrid Camera

In addition to limited spatial resolution, the current consumer depth sensor also suffers from low refresh rate. Among all the publicly available commodity depth sensors, the SwissRanger™ and PMD™ can capture the depth at higher speed than 30Hz, but with a much lower resolution at about  $100 \times 200$ . For the well known Kinect depth sensor, it has depth resolution at  $640 \times 480$  for Kinect V1 and  $512 \times 424$  for Kinect V2. Both have a refresh rate of 30Hz. The Azure Kinect, which is the latest version, has much higher resolution( $1024 \times 1024$ ) but the frame rate is still 30Hz. On the other hand, high-speed video has been commonly adopted in consumer-grade cameras and even cellphones. Therefore, in this chapter we present a hybrid camera system that combines a high-speed color camera with a Kinect depth sensor that, with our novel post-processing algorithm, can generate the depth stream that has the same frequency and resolution as the color camera.

Given our hybrid camera setup, one straightforward way to generate high-speed depth stream is to apply bi-directional interpolation based on the optical flow. This is easy to implement. However, it is not always sufficient, since the linear motion assumption is not always true. In fact, usually it is the non-linear motion between frames that makes high-speed video interesting. We present a novel algorithm that enforces the shading constraints with the flow guidance in a unified framework. Instead of recovering the depth sequence frame by frame, we formulate an objective function with the shading constraints within frames and optical flow constraints between frames.

## 4.1 System setup

We now briefly introduce our system setup and the overall processing pipeline. Our hybrid camera uses a PointGrey™ Dragonfly camera as the high-speed color camera. It captures images with resolution  $640 \times 480$  at 180fps. We put the depth camera, for which we use the Kinect depth sensor, next to the PointGrey camera. The original depth stream is captured at 30fps. The two cameras are calibrated and synchronized with system timestamps.

An overview of our high-speed depth map generation framework is shown in Figure 4.1. First, we will obtain the flow information between color images. In the meantime the

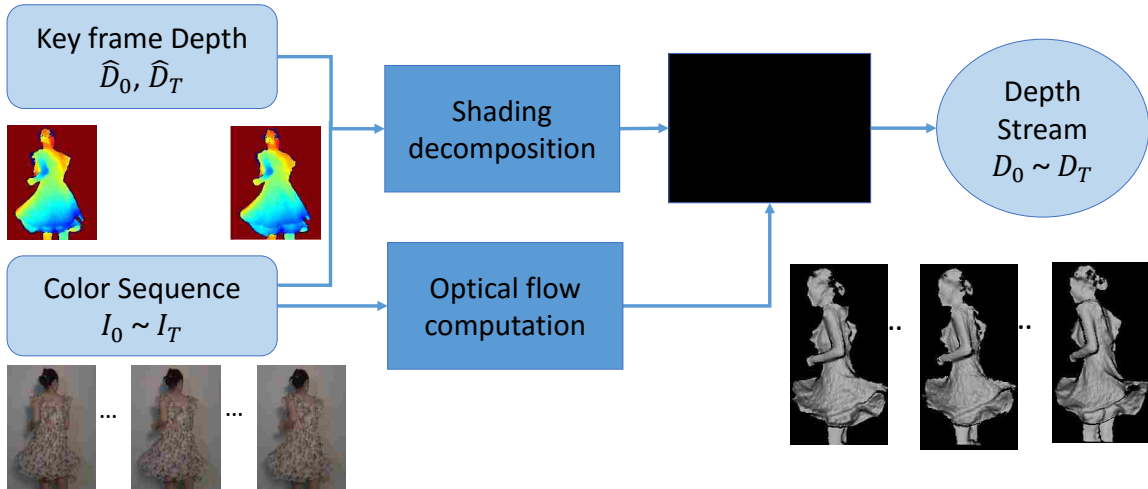


Figure 4.1: System Pipeline.

lighting condition is estimated in these key frames, which is assumed to be changing slowly over time. Then the albedo and shading images for the whole sequence are estimated with a novel decomposition algorithm. It uses the depth cues in key frames and temporal constrains (Section 6.2.2). Finally, the shading cues and flow information are combined together into our proposed global optimization framework, from which we can generate the depth stream ( $D_0 \sim D_T$ ) between any two key frames (Section 6.2.3), leading to an output RGB-plus-depth stream at 180Hz (the same rate as the high-speed color camera). Notice that we do not attempt to increase the spatial resolution of the depth map. Existing algorithms can be adopted if desired.

## 4.2 Our Approach

Suppose we have two depth frames  $\hat{D}_0$  and  $\hat{D}_T$ , which are denoted as the depth key frames, and color frames  $I_0 \sim I_T$ , where  $I_0$  and  $I_T$  correspond to the depth frame  $\hat{D}_0$  and  $\hat{D}_T$

respectively. Our goal is to estimate or refine the depth frames  $D_0 \sim D_T$ .

### 4.2.1 Preprocessing

First, we will describe some pre-processing steps and clarify some notations.

**Depth warp** The two cameras are calibrated and we warp the key frame depth maps into color coordinates using the calibrated intrinsic and extrinsic parameters.

**3D point** Suppose we have a pixel  $p = (i, j)$  with depth  $D(p)$ , then with the depth camera intrinsic  $K$  we can get its 3D point position as

$$X(p) = D(p)K^{-1}(i, j, 1)^T, \quad (4.1)$$

$$K = \begin{pmatrix} f_x & 0 & \mu \\ 0 & f_y & \nu \\ 0 & 0 & 1 \end{pmatrix} \quad (4.2)$$

where  $f_x$  and  $f_y$  are the focal length in  $x$  and  $y$  direction,  $\mu$  and  $\nu$  are the camera's principal point.

**Normal** We use the perspective camera projection model, and the unnormalized normal  $\tilde{\mathbf{n}}_p$  for pixel  $p$  can be computed as,

$$\tilde{\mathbf{n}}(p) = (X(i, j+1) - X(p)) \times (X(i+1, j) - X(p)) \quad (4.3)$$

Then substitute the 3D points in Eq. 4.3 with Eq. 4.2 and Eq. 4.1, the normal can be written as

$$\tilde{\mathbf{n}}(p) = \frac{D(i+1, j) \cdot D(i, j+1)}{f_x \cdot f_y} \begin{pmatrix} \frac{f_x \cdot (D(i, j+1) - D(p))}{D(i+1, j)} \\ \frac{f_y \cdot (D(i+1, j) - D(p))}{D(i, j+1)} \\ \frac{(\mu-j) \cdot (D(i, j+1) - D(p))}{D(i+1, j)} + \frac{(\nu-i) \cdot (D(i+1, j) - D(p))}{D(i, j+1)} - 1 \end{pmatrix} \quad (4.4)$$

Finally, we can normalize it to  $\mathbf{n}(p)$ .

**Optical flow** We have the color sequence  $\mathbf{I} = [I_0, I_1, \dots, I_T]$  and the corresponding mapping between any two neighboring frames can be obtained from optical flow [19].  $W_t(p)$  maps the pixel  $p$  in frame  $t$  to next frame  $t+1$ .

### 4.2.2 Intrinsic decomposition

Before enforcing the shading constraints for generating high-speed depth stream, we need to perform the intrinsic image decomposition to separate the shading effect from albedo and estimate the lighting condition using the depth key frames.

## Lighting estimation

Similar to the majority of prior work on lighting estimation, we assume the object surfaces to be Lambertian, based on which Spherical Harmonics (SH) can be used to represent the incident lighting  $L$  efficiently. We use the first nine SH basis functions (up to second order), which is a good approximation for Lambertian reflectance [113]. Then the reflected irradiance  $I$  for each pixel  $p$  can be represented as

$$I(p) = A(p) \cdot S(p), \quad (4.5)$$

$$S(p) = \sum_{m=1}^9 l_m H_m(\mathbf{n}(p)), \quad (4.6)$$

where  $A(p)$  is represented as albedo vectors for pixel  $p$  which contains three channels, and  $S(p)$  donates the scalar shading value for  $p$ .  $l_m$  are the corresponding SH coefficients of incident lighting,  $H_m(\mathbf{n}_p)$  represent the SH basis functions (see section 2 in supplementary material for more details).

Similar to [124], we assume that the pixels within the same super-pixel share the same albedo value, therefore we cluster the color image to  $SN$  segments with super-pixel algorithm [4].

For every two neighboring key depth maps with their color images, we can get the  $l_m$  by minimizing the energy function,

$$E_L = \sum_{t \in \{0, T\}} \sum_{sn=1}^{SN} \sum_p \|A_t^{sn}(p) \sum_{m=1}^9 l_m H_m(\mathbf{n}_t(p)) - I_t(p)\|^2, \quad (4.7)$$

where  $SN$  is the number of superpixels segments of color images,  $A_t^{sn}(p)$  means the albedo in frame  $t$  for pixel  $p \in sn$ th segments, which is approximated by the mean color of the superpixel.

We can solve this minimization by computing the mean albedo of the superpixels and lighting coefficients in an iterative way and the iteration starts by setting the albedo as mean color of pixels inside each superpixels. Since we assume that the lighting is changing slowly during the data capture, we use the same lighting coefficients  $l_m$  for the sequence between two key frames.

## Shading and albedo computation

The goal of shading computation is recovering the albedo  $A(p)$  and shading  $S(p)$  that best match the image  $I(p)$ . This is an essential step before SfS can be used. Different from the previous works on shading decomposition using RGBD images, we do not have depth



images in every frame but only some key frames. On the other hand, the depth information in key frames provides us the valuable cues to resolve the ambiguity for single image decomposition. Therefore, we propose to compute the shading images for the sequence between any two key frames in a unified optimization making use of the depth cues in key frames and albedo consistency constraints along the sequence. The objective function is described below.

First, we have the data terms. The first is to match the image input. We operate in the logarithmic domain and  $\hat{A}$ ,  $\hat{S}$  and  $\hat{I}$  stand for the logarithm of albedo, shading and color image, respectively.

$$E_{d.im} = \sum_{t=0}^T \sum_{c \in \{R,G,B\}} \sum_p \omega_t^{lum}(p) \|\hat{A}_t(p, c) + \hat{S}_t(p) - \hat{I}_t(p, c)\|^2, \quad (4.8)$$

where

$$\omega_t^{lum}(p) = lum(p) + \epsilon \quad (4.9)$$

This term is weighted by the luminance of the input intensity image to prevent disproportionately strong affect of dark pixels.

The second data term is to preserve the initial shading images in key frames, which can be computed with the lighting vector(Section 4.2.2) and coarse surface normals. The generated shading images may not be accurate but they are good approximations.

$$E_{d.s} = \sum_{t \in \{0, T\}} \sum_p \left\| \log \left( \sum_{m=1}^9 l_m H_m(\mathbf{n}_p) \right) - \hat{S}_t(p) \right\|^2 \quad (4.10)$$

**Regularization terms** Next, we have the spatial priors for shading and albedo in each frame, as well as the temporal consistency priors of albedo between neighboring frames.

The albedo smoothness term for each frame with adaptive weighting is

$$E_{s.a} = \sum_{t=0}^T \sum_{p, q \in \mathcal{N}} \omega_t^a(p, q) \|\hat{A}_t(p) - \hat{A}_t(q)\|^2 \quad (4.11)$$

The adaptive weight is computed with the differences of intensity and chromaticity between adjacent pixels

$$\omega_t^a(p, q) = \begin{cases} 0 & \text{if } \nabla ch_t > \tau_{ch} \\ 0 & \text{if } \nabla lum_t > \tau_{lum} \\ \exp\left(-\frac{\nabla ch_t^2}{\sigma_{ch}^2}\right) \cdot \exp\left(-\frac{\nabla lum_t^2}{\sigma_{lum}^2}\right) & \text{otherwise} \end{cases} \quad (4.12)$$

where

$$\nabla ch_t = \|ch(I_t(p)) - ch(I_t(q))\|_2 \quad (4.13)$$

$$\nabla lum_t = ||lum(I_t(p)) - lum(I_t(q))|| \quad (4.14)$$

The shading smoothness term is formulated as below

$$E_{s.s} = \sum_{t=0}^T \sum_{p,q \in \mathcal{N}} \omega_t^n(p,q) ||\hat{S}_t(p) - \hat{S}_t(q)||^2, \quad (4.15)$$

For key frame 0 and T, we have coarse depth from which we can compute the surface normal  $\mathbf{n}$ , and then the smooth weight  $\omega_t^n(p, q)$  is set as

$$\omega_t^n(p, q) = 1 - exp\left(-\frac{(\mathbf{n}_p^T \mathbf{n}_q)^2}{\sigma_n^2}\right) \quad (4.16)$$

It means that we will favor smooth shading for adjacent pixels that have similar normal orientations. For other frames  $1 \sim T - 1$  that only have color images, the smooth weight may be set as a constant value. However, the weight is not easy to choose. As shown in figure 4.2 , the texture is not separated clearly from shading image if the weight is small (figure 4.2(b)), while the shading is over-smoothed when the weight is large (figure 4.2(c)).

To deal with this problem, we decide to propagate the adaptive weighting terms computed in key frames into other frames. In more detail, the weight term in frame 0 is propagated forward to next frames using the following equation

$$\omega_t^{n,f}(p, q) = \begin{cases} \omega_{t-1}^{n,f}(p_{t-1}, q_{t-1}) & p_{t-1}, q_{t-1} \in \mathcal{N} \text{ for } 1 < t < T \\ \omega_0^n(p_0, q_0) & p_0, q_0 \in \mathcal{N} \text{ for } t = 1 \\ 0.1 & \text{otherwise for } t \in [1, T - 1] \end{cases} \quad (4.17)$$

In the above formula, for adjacent pixels  $p$  and  $q$  in frame  $t$ , the corresponding pixels  $p_{t-1}$  and  $q_{t-1}$  in previous frame  $t - 1$  can be found with optical flow. The smooth weight is propagated from previous frames when the adjacent pixels stay connected as neighbours, otherwise is set to 0.1. In this way, the smooth shading is propagated and shading details will still get preserved. We can compute the weight  $\omega_t^{n,b}$  using backward propagation from frame  $T$ . The weight is blended as

$$\omega_t^n(p, q) = max\{\omega_t^{n,f}(p, q), \omega_t^{n,b}(p, q)\} \quad (4.18)$$

The temporal albedo consistency is defined as

$$E_{t.a} = \sum_{t=0}^{T-1} \sum_p \omega_t^{fb}(p) \omega_t^{ta}(p) ||\hat{A}_t(p) - \hat{A}_{t+1}(W_t(p))||^2, \quad (4.19)$$

The weight terms are necessary as we need to prevent the artifacts caused by occlusion between frames and the inaccuracy in flow computation. The weighting term  $\omega_t^{fb}(p)$  is used

to discard the occluded regions and is set to 0 if the forward-backward flow consistency check fails, otherwise it is set to 1. We set the weighting term  $\omega_t^{ta}$  as

$$\omega_t^{ta}(p) = \begin{cases} 0 & \|I_t(p) - I_{t+1}(W_t(p))\|_2 > \tau_{ta} \\ 1 & \text{otherwise} \end{cases} \quad (4.20)$$

The final objective function is the weighted sum of all terms, that is

$$E_l = E_{d.im} + \lambda_{d.s} \cdot E_{d.s} + \lambda_{s.a} \cdot E_{s.a} + \lambda_{s.s} \cdot E_{s.s} + \lambda_{t.a} \cdot E_{t.a} \quad (4.21)$$

The objective function can be solved with linear optimization.

With the above propagation framework, for middle frames that do not have depth images, we are still able to decompose the texture from color images while preserving shading details, as shown in figure 4.2(d).

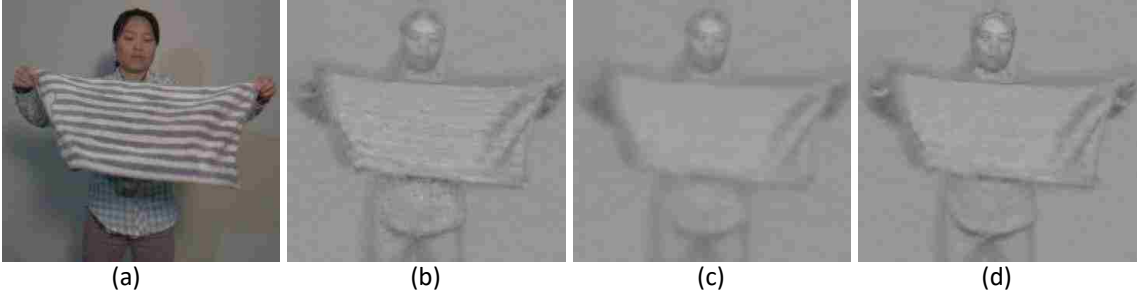


Figure 4.2: Results on intrinsic decomposition. (a) a sampled middle frame; (b) decomposed shading image when constant shading smooth weight  $\omega_t^n$  set as 1.0; (c) shading image computed when  $\omega_t^n$  is set as 6.0; (d) our shading image with adaptive smooth shading weight.

### 4.2.3 Depth stream generation framework

Now we have all the albedo and shading images for each frame ( $A_0 \sim A_T$  and  $S_0 \sim S_T$ ), and the key frame depth images ( $\tilde{D}_0$  and  $\tilde{D}_T$ ), we can recover the in-between depth frames.

Given equally sampled temporal frames indexed by  $t$ , the 3D scene velocity,  $U_t(D(p))$  at pixel  $p$ , is computed as (for simplicity, we omit  $p$  here),

$$U_t(\mathbf{D}) = X_{W_t}(D_{t+1}(W_t)) - X(D_t) \quad (4.22)$$

where  $\mathbf{D} = [D_0, D_1, \dots, D_{T-1}]$ . The above function is the velocity field for the sequence.

It is assumed that the motion between neighboring color frames should not be too fast and will prefer small velocities that are spatially smooth. Our first term is denoted as the velocity term,

$$E_v = \sum_{t=0}^{T-1} \sum_p (||U_t(D)||^2 + \lambda_v \psi(||\nabla U_t(D)||^2)) \quad (4.23)$$

where

$$\psi(a^2) = \sqrt{a^2 + \varepsilon^2} \quad (4.24)$$

The first term in Eq. 4.23 has the similar effect as bi-linear interpolation that will spread the spatial movement in 3D between the two key frames into the whole sequence. The second term in Eq. 4.23 is exploited here to preserve the smooth scene flow field [51]. We do not enforce any linear constraints for this term while only spatial smoothness is favored for the 3d flow field. Therefore, the velocity term will try to preserve the smooth transition along the depth sequence while allowing for non-linear motion.

Secondly, with the albedo images computed from the previous section, we have the shading constraint that penalizes the differences between rendered images and the captured color images. We use the Charbonnier penalty function(Eq. 4.24), which is more robust to the outliers in the shading image,

$$E_s = \sum_{t=0}^T \sum_p \psi(||A_t(p) \sum_{m=1}^9 l_m H_m(\mathbf{n}(p)) - I_t(p)||_2^2) \quad (4.25)$$

For the above formula,  $\mathbf{n}(p)$  can be substituted with depth  $D$  based on the normal computation equation (Eq. 4.4 ). Therefore, the shading constraints are directly enforced on depth streams.

Next, in order to have the smooth surface, we utilize a Laplacian smoothness term for each frame. We use the adaptive weighting again to preserve the surface boundary.

$$E_{lap} = \sum_{t=0}^T \sum_p \left( D_t(p) - \frac{\sum_{q \in \mathcal{N}_p} \omega_t(p, q) \cdot D_t(q)}{\sum_{q \in \mathcal{N}_p} \omega_t(p, q)} \right)^2 \quad (4.26)$$

In the above formula, the weighting term is computed using the Gaussian filter as,

$$\omega_t(p, q) = \exp\left(-\frac{||D_t(p) - D_t(q)||^2}{\sigma_{lap}^2}\right) \quad (4.27)$$

Finally, the data term enforces the constraint provided by two depth key frames  $\tilde{D}_0$  and  $\tilde{D}_T$ , which can be regarded as the boundary condition.

$$E_d = \sum_{t \in \{0, T\}} \sum_p (D_t(p) - \tilde{D}_t(p))^2 \quad (4.28)$$

In summary, our energy function for depth map generation is formulated as

$$E = E_v + \lambda_s \cdot E_s + \lambda_{lap} \cdot E_{lap} + \lambda_d \cdot E_d \quad (4.29)$$

The formulation for generating the depth stream is a non-linear minimization problem. We use the Levenberg-Marquard (LM) method to solve this optimization problem. We have found out that it is difficult to achieve convergence directly since the solution space could have many local minimums. We instead develop a coarse-to-fine refinement strategy. First, we build up the pyramid for shading sequence and two key depth frames. In our implementation, we have four layers for the pyramid. In the first two coarsest levels, the energy function Eq. 4.29 is minimized and we can get the initial rough depth sequence. Then we will refine the depth with shading constraints separately for each frame. In this step, the initial depth propagated from the coarser level will be enhanced with the shading constraints together with Laplacian smoothness with the Eq. 4.25 and Eq. 4.26.

### 4.3 Experiments

We have captured some real data sets with the prototype of our hybrid camera setup described in Section 4.1 and tested our method on these datasets. Also, we have generated some synthetic datasets to validate the method quantitatively. In this section, we will show quantitative and qualitative comparison results. All the parameters in our algorithms are showed on Table 4.1, these values are tuned empirically and remain fixed for all the experiments.

Table 4.1: The parameter settings for our experiments.

Eq. 4.7	Eq. 4.9	Eq. 4.12				Eq. 4.16	Eq. 4.20
$SN$	$\epsilon$	$\tau_{ch}$	$\tau_{lum}$	$\sigma_{ch}$	$\sigma_{lum}$	$\sigma_n$	$\tau_{ta}$
30	0.001	0.06	0.10	0.01	0.05	0.1	0.08
Eq. 4.24	Eq. 4.27	Eq. 4.21			Eq. 4.29		
$\epsilon$	$\sigma_{lap}$	$\lambda_{s-a}$	$\lambda_{s-s}$	$\lambda_{t-a}$	$\lambda_s$	$\lambda_{lap}$	$\lambda_d$
0.001	5mm	2.0	3.0	5.0	0.1	0.01	10

We have also implemented two baseline algorithms. The first is to use the optical flow information to interpolate the in-between frames. We denote this baseline method as the BL (bi-directional interpolation) method. The second is to refine the interpolated depth map with shading information. More specifically, we first apply a low-pass filter to smooth out the interpolated depth maps and then do the refinement with shading constraints, e.g., using method [105], to recover more surface details. We denote this method as the SBL method.

### 4.3.1 Intrinsic image decomposition

We have compared our method with three previous works on intrinsic image decomposition with single RGB-D image as input. As shown in figure 4.3, the upper rows demonstrate the decomposition for key frame images that have captured depth frames as input. For the lower row, it corresponds to a sampled middle frame that has no captured depth as input. For this middle frame, the interpolated depth from the BL method is taken as the input depth for all these comparison methods; for our method the interpolated depth is not needed, since the decomposition is performed using global optimization with key frames and middle frames computed all together.

Chen et al. [21] performs the decomposition without taking surface details into account and tends to get shading image that is quite smooth, as shown in figure 4.3(b). This method performs better on scene level decomposition not on surface details. Jeon et al. [60] explicitly deal with the texture in the image and separate the repeated texture patterns before shading decomposition. Therefore, the grid patterns can be separated into albedo image correctly, while the texture on the scarf stays at the shading image, as shown in figure 4.3(c). Figure 4.3(d) displays the results using the method [11] which takes as input a single RGB-D image and produces as output an improved depth map, albedo image, shading image and illumination model. This approach is sensitive to the outliers in input depth map, especially along the surface boundary, as the zero depth value is not handled explicitly. It will fail in our case where the depth round the moving hands and arms is quite noisy caused by motion blur and also the interpolated depth is wrong around the arms. Figure 4.3(e) shows the results from our method where the texture is successfully decomposed into albedo image while real surface details, i.e. on the hat and the around the arms, can be seen in the shading image.

### 4.3.2 Qualitative evaluation for depth frame generation

In Figure 4.4, we demonstrate the recovered depth of the relative simple case, where the human subject is waving her arms and hands quite fast. For this relative simple case, since the 3D motion can be approximated by the linear motion, the bi-directional interpolation(BL) result is acceptable in most of the parts except for some artifacts along the hands. After the shading refinement(SBL), the depth becomes more smooth and the surface details can be recovered. However, the artifacts have got passed from the interpolation result and the shading refinement could not handle this properly. In comparison with these two methods, our method yield better results thanks to the smooth velocity field constraints in the whole sequence and our bundle optimization approach. Also, considering the motion blur effect

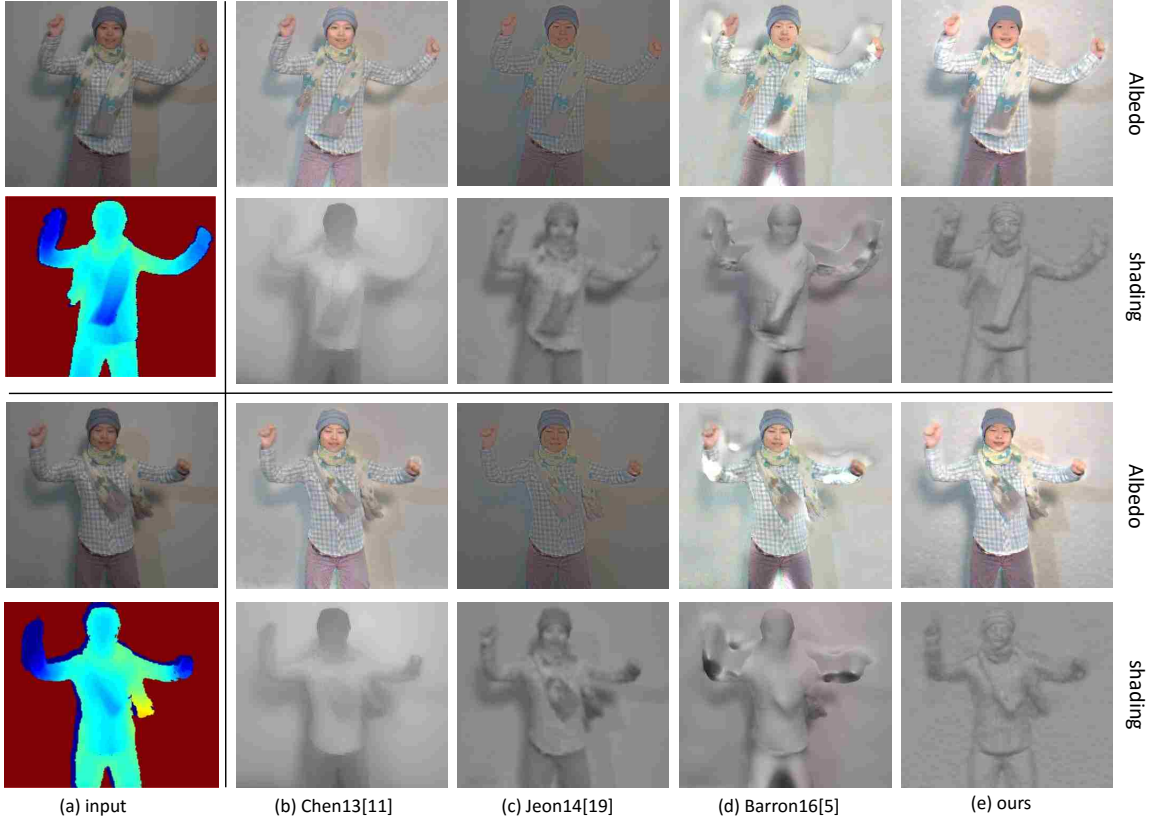


Figure 4.3: Results on sampled images from hand waving stream. (a) Input color image and depth map. (b-e) Albedo and Shading images estimated by three recent approaches for intrinsic decomposition and by our approach.

along the moving boundary, for the key frame depth, we detect the object boundaries with both depth and color contrast and enforce the gradient smoothness in the boundary area along with shading refinement. The refined key frame meshes are shown in (d) and (h).

Figure 4.5 shows the sequence where a towel is waved in front of the hybrid camera. For this case, the towel is experiencing both global translation and local deformations, which are more challenging to recover. Similar to the above case, the error in the interpolated depth will pass to the shading refinement. Also, the texture-copy problem has not been handled properly for the SBL method, as the stripe patterns on the towel are visible as hallucinated surface details. As most of the texture details have been separated clearly with our intrinsic decomposition, the recovered depth from our method is more faithful with the real surface details preserved.

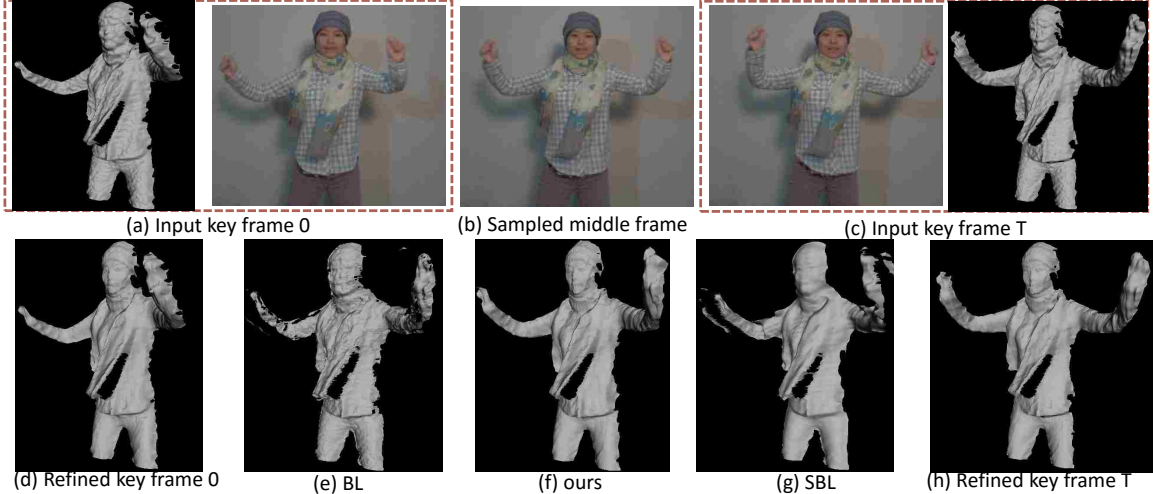


Figure 4.4: Results on the hand waving sequence. To better show the shape of the hands, we adjust the perspective angle of the generated mesh. We have two key frame color images and the initial meshes as input shown in (a) and (c). (d) and (h) are the key frame meshes after refinement. The second row shows the interpolation result of one sampled frame shown in (b). (e)-(g) are the recovered meshes using BL, our proposed method and SBL, respectively.

### 4.3.3 Quantitative evaluation

**Quantitative evaluation for Synthetic data** We use synthetic data to demonstrate the effectiveness of the global optimization framework for depth map generation compared to frame-by-frame method. The shading images are supposed to be given for the optimization framework, therefore we ignore the influence of albedo or texture.

We have generated two cases of synthetic data displayed in Figure 4.6 and Figure 4.7. For each sequence, it consists of seven depth frames with resolution of  $64 \times 64$ . We have the ground-truth lighting condition, therefore we can render the shading images for each frame. The optical flow between any frames are all set to zeros for simplicity, which means that the motion is purely in the  $z$ -direction. Given the first and last depth frame as two key frames and the shading sequence, we present the result computed with BL, SBL, and also our method.

The first case (Figure 4.6) displays a simple case where the surface is bending forward along with the global linear translation. As illustrated in Figure 4.6, both SBL and our approach can recover the depth with quite small error.

For the second case (Figure 4.7), we have generated the depth sequence that has the sine wave in the shape deformation and also the global translation, which is the complex combination of linear and non-linear motion. For this more complex motion, our method



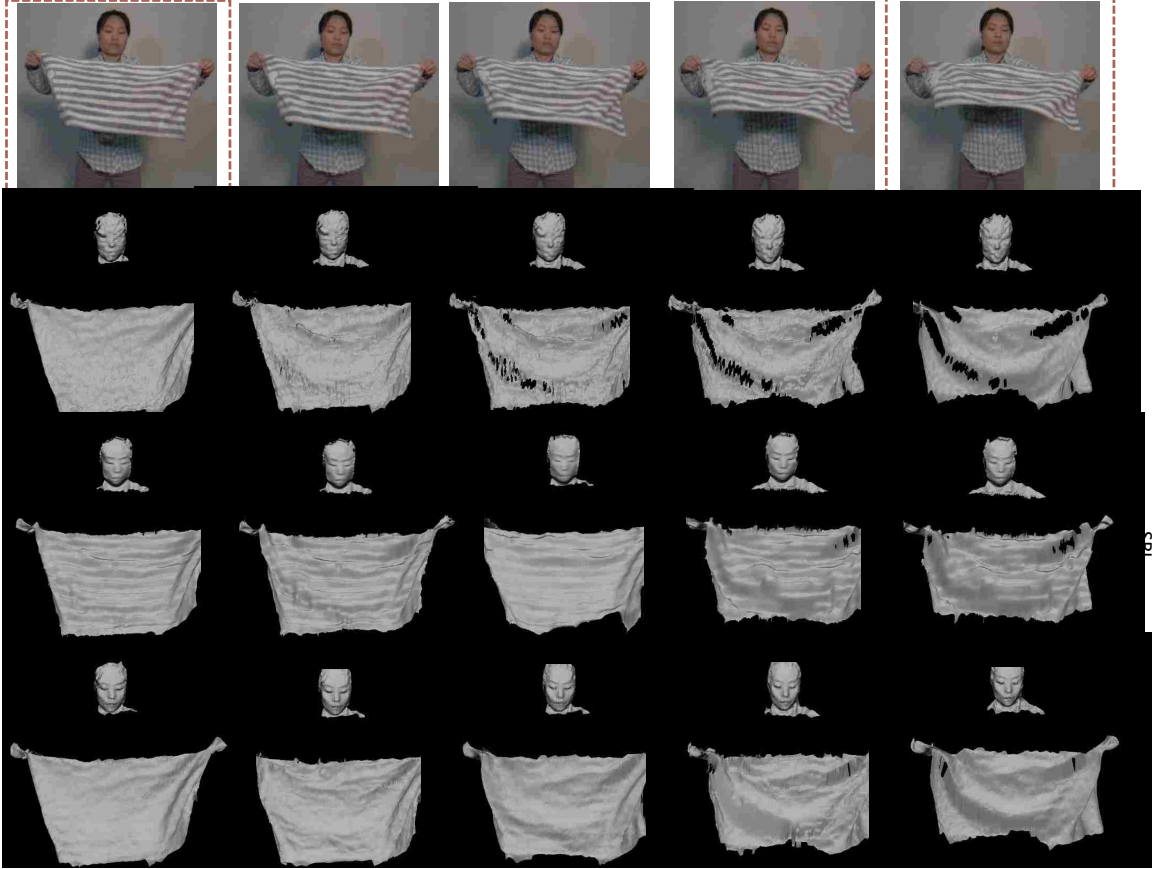


Figure 4.5: Results on the towel shaking sequence. To better demonstrate the shape of the towel, we adjust the perspective angle of the generated mesh which differs from the color image. First row shows the input color sequence with left most and right most have the corresponding input depth frame, while the middle ones are sampled frames between these two frames. Second row gives the input depth on the left and right, and interpolated depth with BL method for the middle frames. Third row displays the shading refinement result using SBL method. The last row shows the recovered depth from our method.

will get smaller error than the BL and SBL method. The regularization terms and the global optimization will make our method converge to a better local minimum that is closer to the real depth.

For Figure 4.6 (Figure 4.7), the first row shows the mesh and shading image, which are the key frames as our input. Figure 4.6a (Figure 4.7a) is the mesh for the first frame with the shading image shown in Figure 4.6b (Figure 4.7b). We put the meshes of the key frames together in Figure 4.6c (Figure 4.7c). The second row shows one sampled frame, where the first column Figure 4.6e (Figure 4.7e) is the groundtruth mesh and the next three columns present the error map for the depth maps recovered with BL, SBL and also our method respectively.

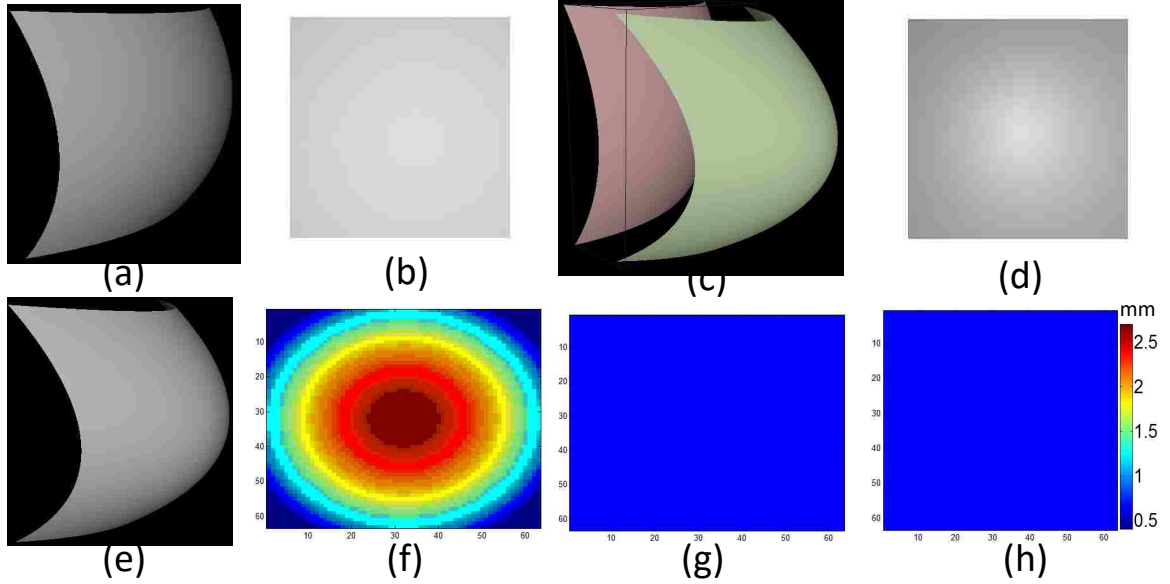


Figure 4.6: Results on synthetic data (simple case).

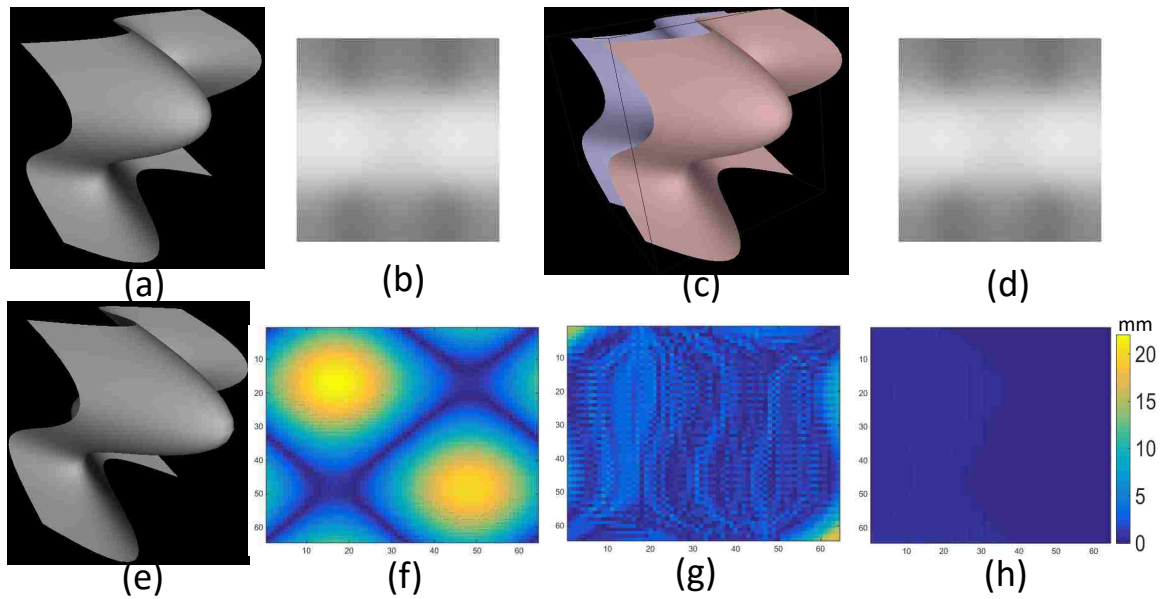


Figure 4.7: Results on synthetic data (more complex case). The two shading images (b) and (d) are the same to each other as we set the sequence to have half period of sine wave and these two key frames have the same shape, but they have the global displacement in depth, as shown in (c).

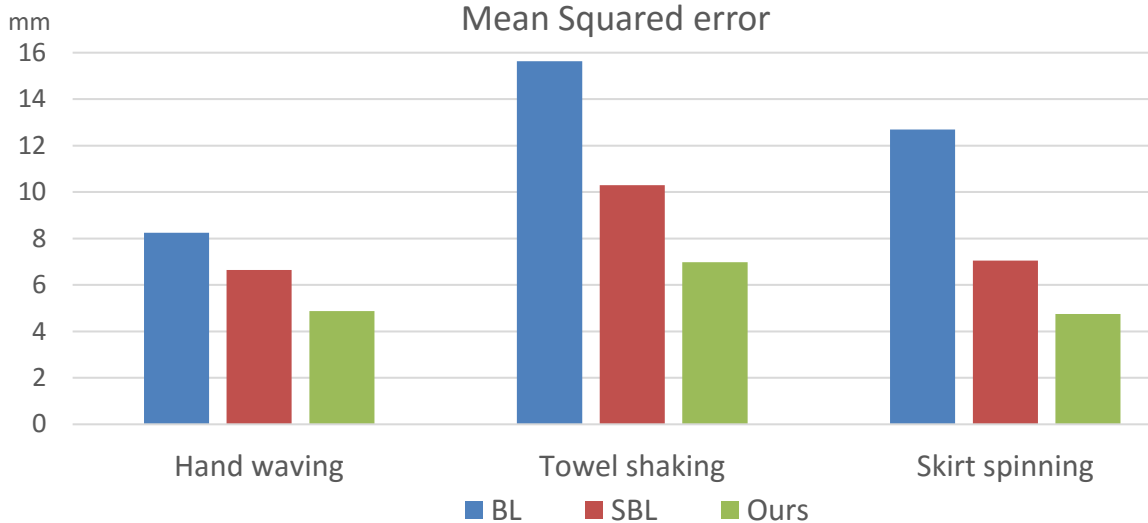


Figure 4.8: Quantitative results for real datasets. All the results are computed with squared mean error. We discard the extreme outliers around the surface boundaries during the evaluation.

To better illustrate the results, we compute the relative depth error with the formula below, where the error is evaluated using the difference between recovered depth and the ground-truth depth divided by the displacement from the first frame,

$$err = \frac{1}{T-1} \sum_{t=1}^{T-1} \frac{\sum_p |D^t(p) - R^t(p)|}{\sum_p |D^t(p) - D^0(p)|} \quad (4.30)$$

For the BL method, the relative error is 0.1962 for the simple case and 0.3788 for complex case. The error will decrease to 0.0012 after the shading refinement (SBL) for the simple case and 0.0837 for the complex one. For our proposed method, the error is 0.0012 and 0.0569 respectively.

**Quantitative evaluation for real data** To better validate our method on the real datasets with some quantitative measurement, we sample several frames from our real datasets and down-sample the captured depth frames into 15Hz and then use these frames to recover all the depth sequences corresponding to color frames (180Hz), which means for every two key frame depth maps, we will get thirteen depth frames interpolated from the interpolation methods. The middle frame corresponds to the original depth frames in 30Hz. The captured depth frames can be approximated as the groundtruth depth to measure the accuracy of the interpolated depth maps. We evaluate the BL method, SBL method and our approach using this strategy on three real datasets we have captured. We evaluate the performance of each method by measuring the mean value of the depth error compared to ground-truth depth. The results are shown in Fig. 4.8. Again our method is the best one.

## 4.4 Conclusion

In this chapter, we present a novel hybrid camera setup, which is composed of a high-speed color camera and a depth sensor. Using this hybrid camera, we propose a framework to recover the depth sequence of the scene with high-speed motion. Considering that the bi-directional interpolation method will fail when there exists non-linear motion in the scene, we exploit the shading constraints in each color frame to overcome this limitation. In our formulation, we use the depth information captured from low-speed depth camera as the boundary constraints for the whole sequence and enforce the SfS constraints in each frame. Also, the depth maps of neighboring frames are associated with our proposed velocity term that preserves smooth motion field. Therefore, we can recovery the depth sequence in a single optimization. Finally, we present the comparison results for real datasets captured with our hybrid camera and also for synthetic datasets. Our high-speed and high-quality RGB-D sequence can be used in many areas where the motion is fast, such as sport event, gait analysis, etc.

## Chapter 5

# SparseFusion: Dynamic Human Body Reconstruction from Sparse RGBD Images

3D modeling or reconstruction of human bodies is a very hot topic which has been studied for decades due to its vast applications in biometrics, virtual reality, gaming, etc. Many scanning systems [59, 29, 85] have been presented under a multi-view setup [30], from which pleasant and impressive results have been achieved. However, the system is usually expensive and not portable. Therefore, instead of using any complex setups, in this chapter we intend to build up complete 3D human avatar using a single commodity RGBD camera, which is a challenging task because of the almost inevitable non-rigid motion and also surface occlusion.

Scanning systems [111, 87, 131, 86, 118, 132] have been proposed by tracking the dynamic surface motion along the RGB-D sequence. Although very impressive and pleasing results have been achieved, they rely on reliable and continuous dense tracking over the whole sequence which is computational expensive and contains much redundant information. To address this issue, we propose to use only several sparse RGBD frames to build up a complete and watertight human model with clear and consistent textures. Similarly, Li [74] has presented the system that takes eight partial pieces as input. However, in this method the user is supposed to keep a certain static pose while rotating in front of the sensor. On the contrary, we are able to handle sparse fusion of human body under various poses.

To achieve this goal, we exploit the SMPL [88] model as a human template to register sparse frames of the human into a canonical model. First, we optimize the SMPL parameters so that the optimized SMPL model will closely fit to the partial scans generated from

the input depth images. We align every two pieces that have sufficient overlap using the correspondences transferred by the SMPL template model. Starting from this initial alignment, we use the color information to get better registration. A global non-rigid registration procedure is performed to get all those partial pieces deformed into a canonical coordinate as guided by those correspondences acquired from the pairwise registration.

## 5.1 Approach

We are given sparse frames of a human subject captured under different poses with different body orientation. It is a partial scan of the human body for each frame and our goal is to build up a complete human model by fusing all those partial scans. In this section, a generative probabilistic human template called SMPL model is exploited to register sparse frames into a canonical model.

The SMPL model is a skinned vertex-based model which parametrizes a triangulated mesh by pose and shape parameters. The shape parameters  $\beta$  are coefficients of a low-dimensional shape space, learned from a training set of thousands of registered 3D human body scans. The pose parameters  $\theta$  represent the joint angle in an axis-angle representation of the relative rotation between body parts. The posed body model  $\mathcal{M}(\beta, \theta)$  is formulated as below given the shape and pose parameters,

$$\mathcal{M}(\beta, \theta) = W(T_P(\beta, \theta), J(\beta), \theta, \Omega) \quad (5.1)$$

$$T_P(\beta, \theta) = T + B_S(\beta) + B_P(\theta) \quad (5.2)$$

where  $T$  is a base template mesh,  $B_S(\beta)$  and  $B_P(\theta)$  are vectors of vertices representing offsets from the base template as controlled by the shape and pose parameters respectively.  $W()$  is a blend skinning function which transforms the mesh from  $T$  pose to the current pose as controlled by the joint position  $J(\beta)$  and blending weights  $\Omega$ . More details about the SMPL model can be found in paper [88].

We optimize the SMPL model to let it fit to each of the partial scans. And then We align every two partial pieces that have great overlap regions by using the correspondences conveyed by the SMPL model. After that, we register those pieces altogether with a global non-rigid registration approach. In the following equations,  $M_1 \sim M_N$  denotes the partial scans from the depth images and  $I_1 \sim I_N$  are the color images.

### 5.1.1 Initial fitting

For every frame of the RGBD images, we solve the pose and shape parameters of the SMPL model so that the generated 3D human model fits as closely as possible to the captured RGBD image. For each frame  $M_k$  and  $I_k$ , we achieve this by minimizing the following objective:

$$E(\boldsymbol{\beta}, \boldsymbol{\theta}) = E_{data}(\boldsymbol{\beta}, \boldsymbol{\theta}) + E_r(\boldsymbol{\theta}) \quad (5.3)$$

The data term  $E_{data}$  is defined as:

$$E_{data}(\boldsymbol{\beta}, \boldsymbol{\theta}) = E_{surface}(\boldsymbol{\beta}, \boldsymbol{\theta}) + E_{joints}(\boldsymbol{\beta}, \boldsymbol{\theta}) \quad (5.4)$$

For each vertex  $M_k^i$  in the surface  $M_k$ , we minimize its distance to the closest vertex on the generated SMPL model  $\mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta})$ :

$$E_{surface}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i \in |M_k|} \min_{v \in \mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta})} \|M_k^i - v\|_2^2 \quad (5.5)$$

The joints fitting term  $E_{joints}(\boldsymbol{\beta}, \boldsymbol{\theta})$  is to match the model joints to the joints of the partial scans denoted as  $\hat{J}_{est,i}$ .  $f()$  is the function that transforms the joint from its rest pose to current positions as controlled by the pose parameters using the chain rule defined by the human skeleton. We compute the 2D joint locations in the color image using OpenPose [158], after which the 3D human joints are estimated by back-projecting the 2D joints into 3D space with the depth information.  $\rho()$  is a robust Geman-McClure penalty function [42]. This term is important to enable solving large pose changes.

$$E_{joints}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i \in |J|} \omega_i \rho(f(J(\boldsymbol{\beta})_i, \boldsymbol{\theta}) - \hat{J}_{est,i}) \quad (5.6)$$

The other term  $E_r(\boldsymbol{\theta})$  is a pose regularization term formulated as below which penalizes unusual poses. It is defined as a Gaussian mixture model trained from the CMU dataset [2] where  $N(\boldsymbol{\theta}; \mu_{\boldsymbol{\theta},i}, \Sigma_{\boldsymbol{\theta},i})$  is a Gaussian distribution with its mean and variance denoted as  $\mu_{\boldsymbol{\theta},i}$  and  $\Sigma_{\boldsymbol{\theta},i}$  respectively.

$$E_r(\boldsymbol{\theta}) = -\log \sum_i (c_i N(\boldsymbol{\theta}; \mu_{\boldsymbol{\theta},i}, \Sigma_{\boldsymbol{\theta},i})) \quad (5.7)$$

We get the shape and pose parameters for each piece by minimizing the above objective function so that the optimized SMPL model will fit to the partial scans.

Furthermore, we propose a bundle adjustment approach to refine the shape and pose parameters by minimizing the total misalignment error of all those partial pieces to the SMPL model with respect to a consistent body shape and their poses respectively. For each piece

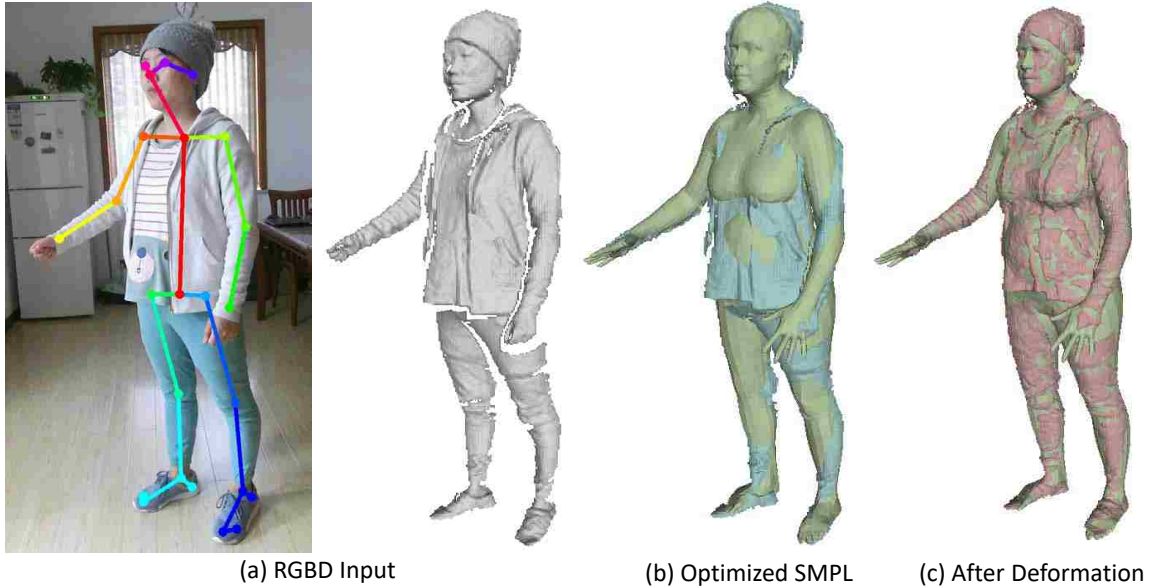


Figure 5.1: Initial Fitting results. (a) is the input RGBD frame and we show the detect joints on the color image. (b) shows the optimized SMPL aligned with the input scan. (c) shows the deformed input scan that fits even better to the SMPL model.

they should have consistent body shapes as for the same human subject. Mathematically the objective function is formulated as below,

$$E(\Omega, \beta) = \sum_{k=1}^N E_{surface}(\beta, \theta_k) \quad (5.8)$$

$$\Omega = \{\theta_1, \theta_2, \dots, \theta_N\} \quad (5.9)$$

We initialize the pose parameters with those computed separately from each piece. The shape parameters are initialized by the one computed from a frontal piece. We show the fitting results in Fig. 5.1 showing the optimized SMPL that fits to the input partial scans.

### 5.1.2 Template guided pairwise alignment

After we get the optimized SMPL model that fits to the input RGBD images, we take it as guidance for initial alignment of those partial scans. Before that, since we cannot find any SMPL model that will fit perfectly to the input mesh, we further deform the input mesh onto the optimized SMPL model to get better alignment, as shown in Fig. 5.1(c). After this, we can establish correspondences from every input scan to the optimized SMPL model via nearest search. And then the correspondences between every two input scans are established through the SMPL model.



Now suppose we want to register the partial piece  $M_i$  to  $M_j$ . We exploit the Embedded Deformation Model [112] to parametrize the mesh. That is, a set of graph nodes  $(g_1, g_2, \dots, g_l)$  are uniformly sampled throughout the mesh, and for each node  $g_i$ , it has an affine transformation specified by a  $3 \times 3$  matrix  $\mathbf{A}_i$  and a  $3 \times 1$  translation vector  $\mathbf{t}_i$ . For each vertex  $v$  it gets deformed by its  $K$  nearest graph nodes with a set of weights:

$$\Phi(v) = \sum_{i=1}^K w_i(v) [\mathbf{A}_i(v - g_i) + g_i + \mathbf{t}_i] \quad (5.10)$$

We compute the deformation from  $M_i$  to  $M_j$  by building a graph for the mesh  $M_i$  and the deformation parameters  $\mathbf{A}_1 \sim \mathbf{A}_l$  (denoted as  $\mathcal{A}$ ) and  $\mathbf{t}_1 \sim \mathbf{t}_l$  (denoted as  $\mathcal{T}$ ) are optimized by minimizing the following objective function:

$$E(\mathcal{A}, \mathcal{T}) = E_{reg}(\mathcal{A}) + E_s(\mathcal{A}, \mathcal{T}) + E_{cor}(\mathcal{A}, \mathcal{T}) \quad (5.11)$$

The term  $E_{reg}$  serves as the as-rigid-as-possible term that prevents arbitrary surface distortion.

$$E_{reg}(\mathcal{A}) = \sum_{i=1}^l \|\mathbf{A}_i \mathbf{A}_i^T - I\|_2^2. \quad (5.12)$$

The smoothness term  $E_s$  ensures smooth deformation of neighboring graph nodes.

$$E_s(\mathcal{A}, \mathcal{T}) = \sum_{(i,j) \in \mu} \|\mathbf{A}_i(g_j - g_i) + g_i + \mathbf{t}_i - (g_j + \mathbf{t}_j)\|_2^2. \quad (5.13)$$

The term  $E_{cor}$  is our data term which penalizes the distances between correspondences on these two pieces, which are extracted through the above optimized SMPL model  $S_i$  for  $M_i$  and  $S_j$  for  $M_j$ . Specifically, for a vertex  $v_p$  on piece  $M_i$ , we find its nearest vertex on  $S_i$  within a certain threshold, which is denoted as  $v_s$ . And we extract the vertex from  $S_j$  which has the same vertex index as  $v_s$ . Then we find the nearest vertex for  $v_s$  with respect to the mesh  $M_j$ , which is denoted as  $v_q$ . The distance between  $v_p$  and  $v_q$  is minimized.

$$E_{cor}(\mathcal{A}, \mathcal{T}) = \sum_{(v_p, v_q) \in \mathcal{C}_{ij}} \|\Phi(v_p) - v_q\|_2^2. \quad (5.14)$$

To get better alignment, we use the color information to refine the initial registration. In details, first every partial scan is textured with its corresponding color image. Suppose we have got the deformed mesh of  $M_i$  which is aligned to  $M_j$  after the above registration, and we denote it as  $D_i^j$ . Now, we render a color image  $I_i$  with the deformed mesh  $D_i^j$  onto the same space with respect to the color image  $I_j$ . We compute a flow field from  $I_i$  to  $I_j$  and map the flow correspondences to the meshes. Finally, the deformation from  $M_i$  to  $M_j$  is

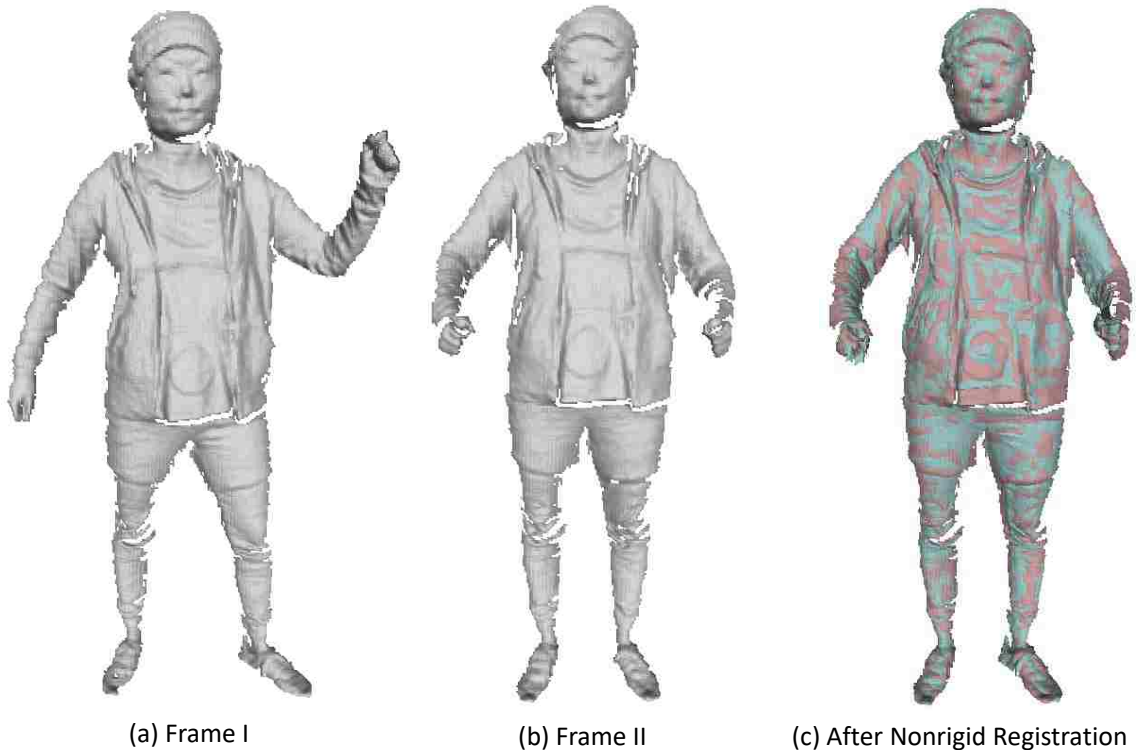


Figure 5.2: Pairwise registration results. (a) and (b) are two sampled pieces. (c) shows our registration result of (a) and (b). The mesh of (a) is deformed onto the mesh of (b).

further optimized using the EDM by enforcing the color correspondences. We show some sampled pairwise registration results in Fig. 5.2.

**Topology Change** With the template guided initial alignment, we are able to deal with the topological changes quite conveniently. That is, while building up the embedded graph, we set further constraints that the vertex is controlled by the graph nodes belonging to either the same body part or neighboring parts defined by its parents or child nodes. In the meanwhile, the smoothness constraints are enforced on nodes that belong to the same body parts.

### 5.1.3 Global alignment

After the initial alignment, we are able to establish correspondences between those partial pieces, with which we can align them globally into a canonical model. Similar to the registration of two partial pieces, we exploit the Embedded Deformation Model here to extrapolate the deformation field, which means for every partial piece ( $M_1 \sim M_N$ ) we have a deformation graph embedded with it and our goal will be to solve those graph parameters ( $\mathbb{A} = \mathcal{A}_1 \sim \mathcal{A}_N, \mathbb{T} = \mathcal{T}_1 \sim \mathcal{T}_N$ ) altogether. The objective function is formulated

as,

$$E(\mathbb{A}, \mathbb{T}) = \sum_{i=1}^N [\alpha_{reg} E_r(\mathcal{A}_i, \mathcal{T}_i) + \alpha_s E_s(\mathcal{A}_i, \mathcal{T}_i)] + \alpha_{corr} E_{corr}(\mathbb{A}, \mathbb{T}) \quad (5.15)$$

The first two terms are the as-rigid-as-possible and smoothness term respectively as defined in Equation 5.12 and 5.13. We have the third term  $E_{corr}$  defined as below as the data term enforcing the correspondences between partial scans achieved from the above pairwise initial alignment.

$$E_{corr}(\mathbb{A}, \mathbb{T}) = \sum_{(M_s, M_r) \in \mathcal{U}} \sum_{(p_i, q_i) \in C_{sr}} \|\phi(M_s^{p_i}, \mathcal{A}_s, \mathcal{T}_s) - M_r^{q_i}\|_2^2 \quad (5.16)$$

where  $M_s$  and  $M_r$  are any two pieces that have sufficient overlaps, and  $C_{sr}$  is the correspondence set we have got after the pairwise alignment. The deformed mesh of  $M_s$  is supposed to fit onto the target mesh  $M_r$  as controlled by the correspondences. Besides, vertices of the reference frame is enforced as fixed constraints.

Finally, with all those input partial pieces deformed to a canonical space, we apply Poisson surface reconstruction to get the final human model  $S$  that is watertight.

#### 5.1.4 Texture optimization

In some applications such as free-viewpoint video generation and teleconference, a 3D geometric human body is not enough and we want the model to be textured. We describe our texture optimization approach in this section. The input is the reconstructed human model together with those partial pieces that are aligned to the canonical model and their corresponding color images. Our goal is to generate consistent and clear texture map for the 3D human model given the input.

Many texture mapping methods project mesh onto multiple image planes, and then adopt weighted average blending strategy to synthesize model textures. However, the generated texture gets quite blurry in our case as the misalignment between those partial pieces still exist which means the textures from different images are not perfectly matched. Therefore, instead of directly synthesizing from multiple images, we optimize a warping field for every image consecutively before attaching these to the mesh model to eliminate possible misalignment.

Starting from the reference frame, we attach the corresponding image onto the reconstructed mesh model by projecting the mesh onto the image plane and compute the texture coordinates for every face that is visible in the reference frame. For the next neighboring

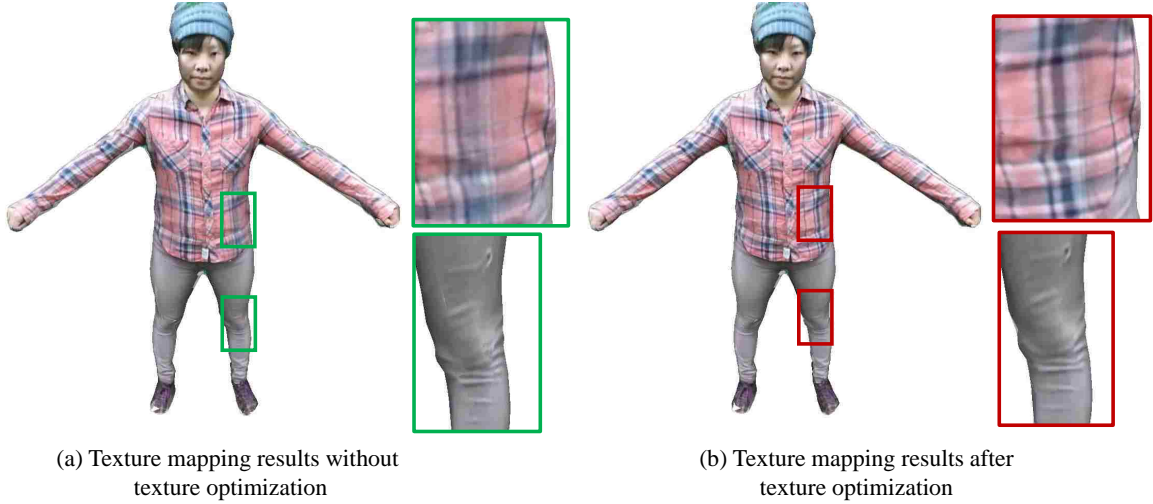


Figure 5.3: Texture optimization results.

frame  $k$ , first we render an color image  $I_{model}$  with the current textured mesh with respect to the view direction of frame  $k$ . In the meanwhile, we have the color image  $I_k$  rendered from the deformed partial mesh piece  $M_k$  that is textured with its corresponding captured image. The possible misalignment between the overlap regions in  $I_{model}$  and  $I_k$  will cause visual seams if we attach the image  $I_k$  directly onto the current human mesh. To address this problem, instead of adjusting the texture coordinates for each face in the 3D mesh which is difficult to optimize, we try to find a warping field  $W_k$  for  $I_k$  in the image plane so that the warped image will get well aligned with  $I_{model}$ . In details, first we detect the overlap regions of the texture map between  $I_{model}$  and  $I_k$ , which we denote as  $\Omega_o$ . A flow field  $\hat{W}_k$  is computed from  $I_k$  to  $I_{model}$  for the overlap part. Next, I propagate the flow field onto the non-overlap part  $\Omega_N$  by minimizing the following objective function, from which the overall warping field is estimated,

$$E(W_k) = \sum_{p \in \Omega_o} (\|W_k(p) - \hat{W}_k(p)\|^2) + \lambda_s \sum_{(p,q) \in N} \|W_k(p) - W_k(q)\|^2 + \lambda_b \sum_{p \in \Omega_N} \|W_k(p)\|^2 \quad (5.17)$$

Where we keep the warping field to be as smooth as possible with the second term. The last term is a boundary term that is enforced to set constraints for pixels that are not connected to the overlap regions.

Afterwards, we select optimal texture image for each face of the human model to generate the final texture maps. In Fig. 5.3, we show the texture mapping results w/o our texture optimization procedure.

Table 5.1: Reconstruction Error

frame number	1	6	8
mean error(mm)	17.1	9.2	7.4

## 5.2 Experiments

We demonstrate the effectiveness of our approach in the experimental part with both quantitative and qualitative results.

### 5.2.1 Quantitative evaluation on synthetic datasets

We tested our system on a synthetic dataset that we have created using Poser [108]. We have selected eight models of a human subject under different poses from Poser and synthesize one depth map and color image from each selected frame with a virtual camera rotating around the subject as shown in Fig. 5.4(a). Our reconstruction system results in a shape with respect to the first selected frame which we take as the canonical frame as shown in Fig. 5.4(b). We plot the error map to show the geometric error of our reconstructed model with respect to the groundtruth model. The error for each vertex is computed via a nearest search to the groundtruth mesh.

We also evaluated our method with only six input frames. As shown in Fig. 5.4(g), we are able to reconstruct the human model with quite sparse frames. We evaluate the reconstruction error of our method using 1, 6 and 8 frames. We take the optimized SMPL model as the reconstructed model for only one frame. Table 5.1 shows the mean error of our reconstructed model.

We compare our results with 3D self-portrait [74], which also takes eight partial pieces as input. As can be seen in Fig. 5.4(f), it becomes quite difficult to align all those partial pieces when there are large pose changes, and the misalignment still exists especially around the arms and legs. On the contrary, we are able to align those partial pieces successfully under our framework, as we have taken the pose variations into account by optimizing both pose and shape parameters of the SMPL model to fit to the input partial scans.

### 5.2.2 Qualitative evaluation on real datasets

For the qualitative evaluation, we have captured several sequences of human subjects with Microsoft Kinect V2. The results of our method are displayed in Fig. 5.5. For each reconstruction, we use twelve RGBD images as input. For each person, we have actually

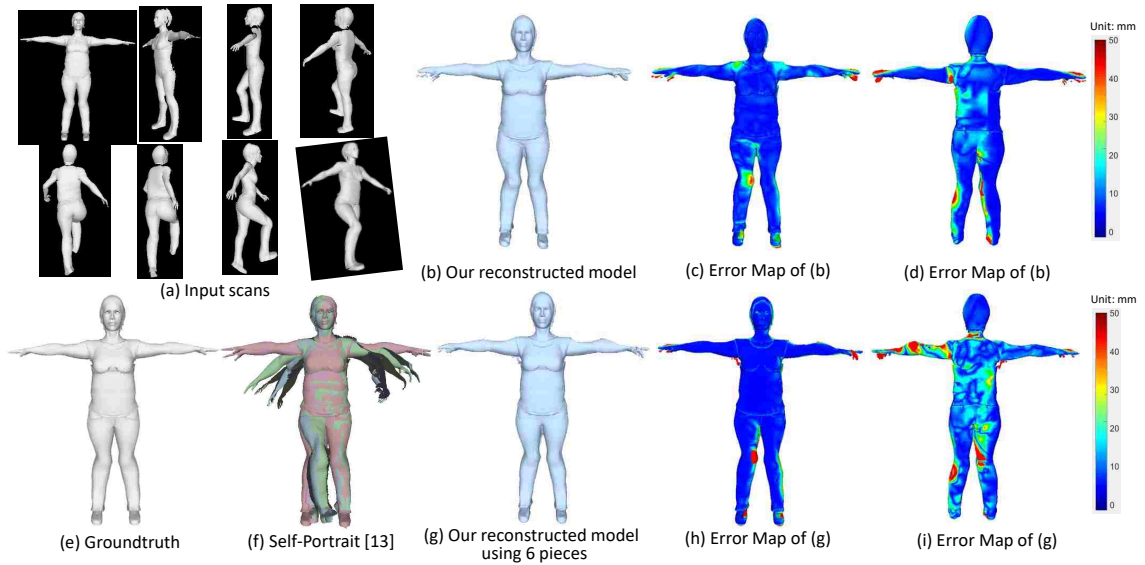


Figure 5.4: Results on a synthetic dataset.

captured a sequence with 360 frames with the person turns around in front of the camera and we select one frame every thirty frames from the sequence. The partial pieces are generated from the selected frames and are smoothed as preprocessing. We take a frontal piece as the canonical space and deform all other pieces onto it. But we are able to generate the fused mesh with respect to any input scan by deforming the canonical mesh onto it.

We demonstrate the effectiveness of our method on dealing with topology changes in Fig. 5.6.

### 5.3 Conclusion and Future work

In this chapter, we have proposed a novel SparseFusion approach to build up a complete human avatar from only sparse RGBD images. In order to align those partial pieces of human body under different poses and viewpoints together, we have exploited the SMPL model as a human template and used it as a bridge to align those partial pieces into a unified 3D model. Experiments on both synthetic and real datasets demonstrate the capability of our framework to reconstruct complete human bodies with accuracy in millimeters.

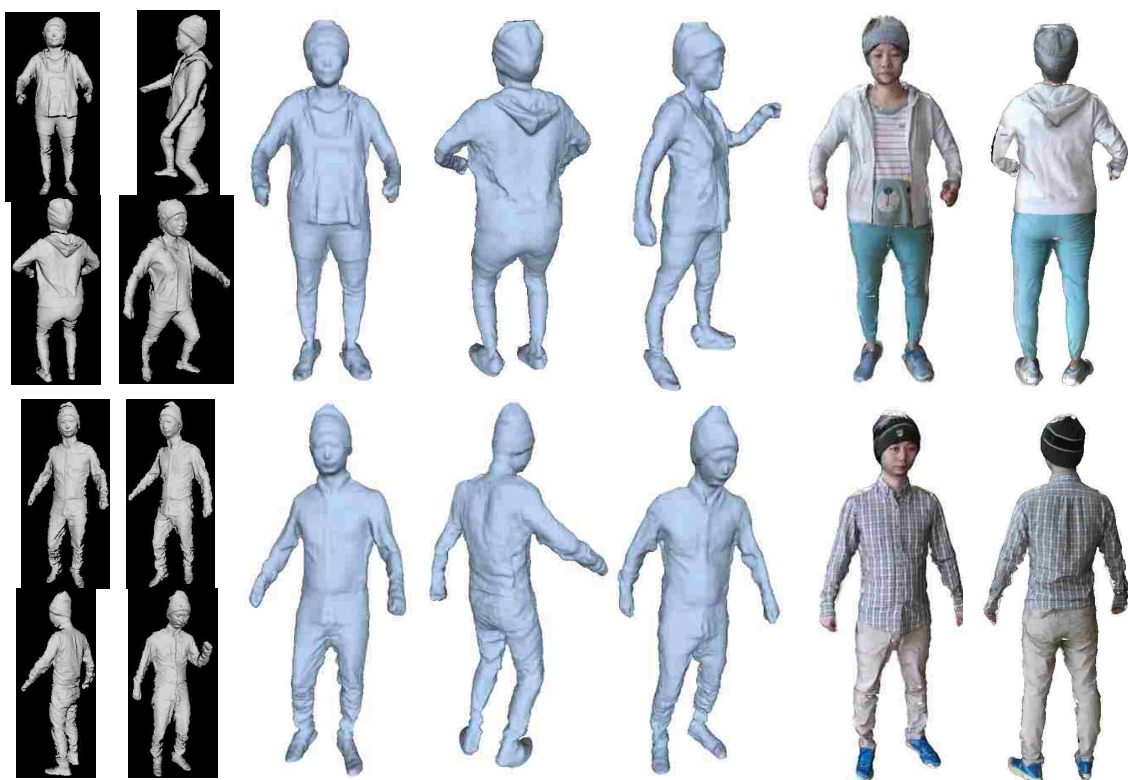


Figure 5.5: Results on real datasets. The left two columns are sampled input scans and the three right columns are the fused model and models deformed to some input scans.



(a) Sampled Frames



(b) Reconstructed Human Bodies

Figure 5.6: Results on changing topologies.



## Chapter 6

# Interactive Visual Hull Refinement for Specular and Transparent Object Surface Reconstruction

In this chapter, I will go beyond the Lambertian surface reconstruction and focus on the 3D reconstruction of specular or transparent objects, which is still a challenging problem in computer vision. Due to their non-Lambertian surface reflectance properties, establishing correspondences – a fundamental requirement for many 3D reconstruction algorithms – becomes difficult or even impossible. The active depth sensors also fail in this case. The reason is that for the stereo based depth sensor like Kinect v1, it cannot find correct correspondences and for the TOF camera, it fails to calculate the phase shift of the emitted signal. Therefore existing methods on reconstructing these difficult objects typically use additional constraints such as specialized active illumination with stripes or checkerboard pattern or known reference objects (e.g., [53, 100, 32, 82, 142]).

In this work we aim to reconstruct highly specular surfaces like glass sculptures and glossy trophies (such as these shown in Figure 6.1), from a multi-view images set casually captured with a hand-held camera, without using active illumination or reference objects (except a few markers for pose estimation). Naturally we decide to use a *visual hull*-based approach that does not require pixel correspondences. The fundamental limitation of the visual hull representation is that it is unable to model concavity. Through careful geometric analysis, we show that some type of concavity can actually be removed by using the *internal occluding contours*, i.e., occluding contours that are inside the object's silhouette. Based on that we present a new visual hull refinement method, which we refer to as *Locally Convex Carving* (LCC).



Figure 6.1: The reconstructed models for glossy trophy and glass sculpture. The left column shows the capture object; the middle column shows the reconstructed 3D model using visual hull; and the right column presents the reconstructed surface using our proposed method.

## 6.1 Approach

Our goal is to reconstruct highly non-Lambertian objects from a set of casually-captured multi-view images, without using any active lights. We assume that viewpoint for each image is known and the object has been segmented from the background. There are many existing techniques and tools that can achieve these two requirements. From the calibrated and segmented images, we first construct a visual hull of the 3D model using traditional volumetric visual hull reconstruction. The overall pipeline is shown in Figure 6.2, in which we develop a novel contour tracking method and a new visual hull refinement scheme that we referred to as *Locally Convex Carving*, finally concave areas are identified and interpolated using boundary conditions. In the next few sections we will present our methods in detail.

### 6.1.1 Terminology

An object’s contour provides important clues about the object shape. Suppose a 3D object  $S$  is viewed by a camera. The object’s silhouette image contains values that distinguish

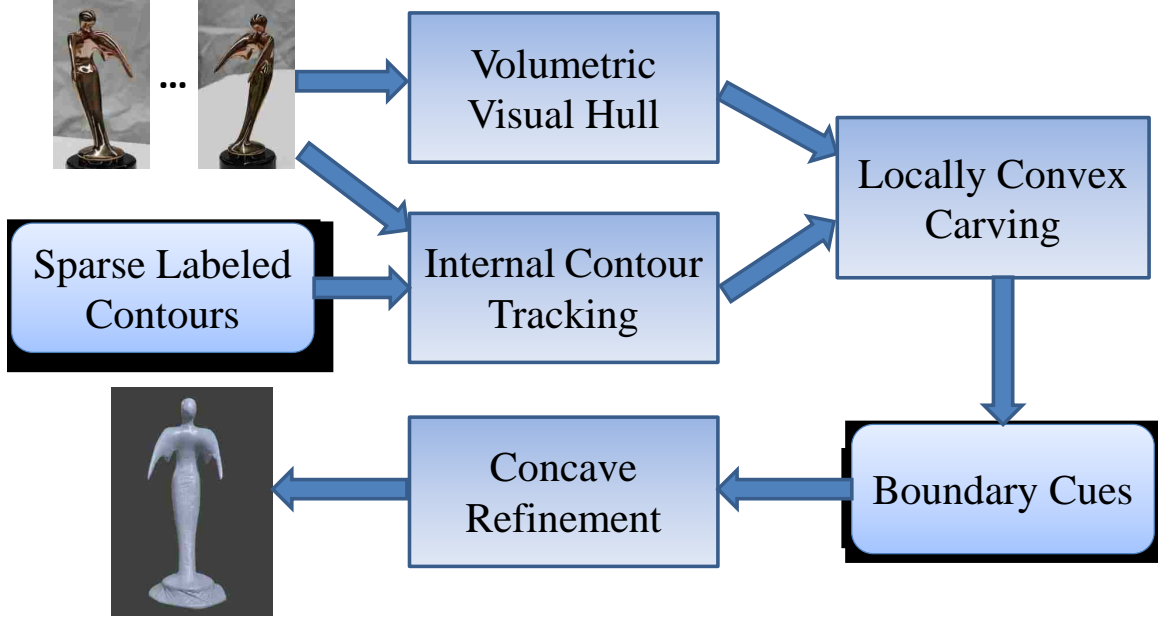


Figure 6.2: The system pipeline of our approach

regions where the object is or is not present. Combined with calibration information for the camera, Each pixel in a silhouette defines a ray in scene space (denoted as  $E^3$ ) that intersects the object  $S$  at some unknown depth. The union of these view rays for all pixels in the silhouette defines a generalized cone within which  $S$  must lie. If we are presented with multiple views of  $S$ , the intersection of these generalized cones from all views defines a volume in  $E^3$  that must contain  $S$ . As the number of the reference views, taking from different locations, goes to infinity, the intersection volume converges to the shape known as the objects *visual hull*, a term defined by Laurentini [76]. The visual hull, denoted as  $VH(S)$ , is guaranteed to contain the object  $S$ . In 2D, the visual hull is equal to the convex hull of the object (denoted as  $CH(S)$ ). For 3D scenes, the visual hull is a tighter fit than the convex hull.

A less-known term, also defined by Laurentini [76], is the *internal visual hull* ( $IVH(S)$ ).  $CH(S)$  segments  $E^3$  into two regions. When all the reference views are taken outside  $CH(S)$ ,  $VH(S)$  is formed. If views are taken outside  $S$ , e.g., including concave areas in  $S$  but still outside  $S$ ,  $IVH(S)$  is formed. It can be shown that  $IVH(S)$  is an even tighter fit than  $VH(S)$ , i.e.,  $S \leq IVH(S) \leq VH(S) \leq CH(S)$ . However it is often difficult, if not impossible, to take pictures in concave areas in  $S$ . So  $IVH(S)$  mostly remains a theoretical concept.

To explain our method, a few more terms need to be defined. As shown in Figure 6.3, a contour is a view-dependent concept, a point in a contour is a point on  $S$  for which

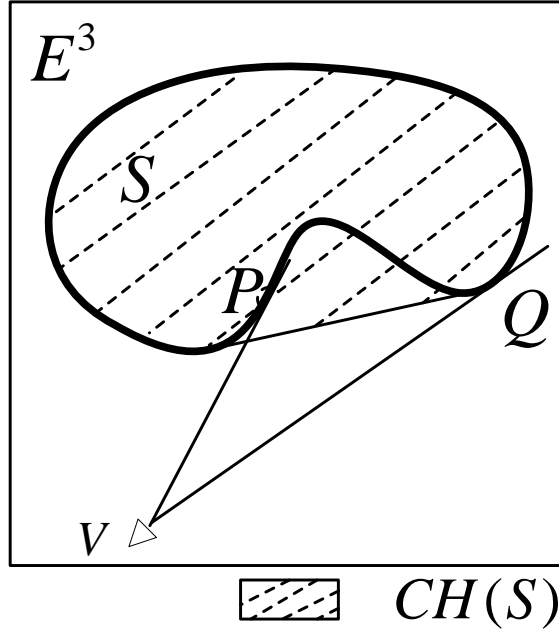


Figure 6.3: Illustration for occluding contours and visual hull.  $S$  is the real surface and  $CH(S)$  is the convex hull which is same as visual hull ( $VH(S)$ ) in 2d.  $P$  is the internal occluding contour point while  $Q$  is the external occluding contour point.

a tangent line is intersecting  $S$ . The intersection point divide the tangent line into two segments. When both segments are outside  $CH(S)$ , it is called an *external contour point* (e.g.  $Q$  in Figure 6.3); when at least one of them is inside the  $CH(S)$ , it is called an *internal contour point* (e.g.  $P$  in Figure 6.3).  $VH(S)$  can also be thought of as a union of the external contour points.  $IVH(S)$ , on the other hand, is the union of all contour points, both external and internal, resulting a tighter fit.

The central idea of our method is to use the internal contours, *without* requiring pictures taken from the concave regions of  $S$ . First we want to point out that most internal contours are actually visible from outside, as long as one segment of the tangent line is outside  $CH(S)$ . For now we assume that the internal contours are already detected and present our internal contour carving algorithm below.

### 6.1.2 Locally Convex Carving

Our carving algorithm starts with an already calculated  $VH(S)$  and detected internal contours. Unlike the external contours, an internal contour captured outside  $CH(S)$  does not define a clear region that separates  $S$  from  $E^3$ . However, by definition, it does point to a set of points that are *locally convex*. We just do not know where they are on the tangent line. Inspired by stereo matching, we plan to use a pair of internal contours to further carve

the volume. As illustrated in Figure 6.4, let us assume that we are given two internal contours  $IC_0$  and  $IC_1$  with respect to viewpoint  $V_0$  and  $V_1$ . We intersect view rays defined by two contours to define a region  $R$ . At a first glance,  $R$  is in the concavity and should be removed (shown in Figure 6.4 left). But close observations lead to the conclusion that the intersecting view rays coming from the two sides of concavity may lead to over or under carving. To prevent this, we define our *Rule (1)*: if the contour normals ( $N_p$  and  $N_q$ ) are opposite from each other, no carving should occur. On the other hand, if the two contours are generated from the same continuous convex surface (as shown in Figure 6.4 right), then  $R$  could be safely carved. However if the direction of the contour is close to parallel to the epipolar plane, the intersections cannot be reliably estimated. So we set up *Rule (2)* to prevent this: the angle between the surface normal and the epipolar plane normal should not be too small.

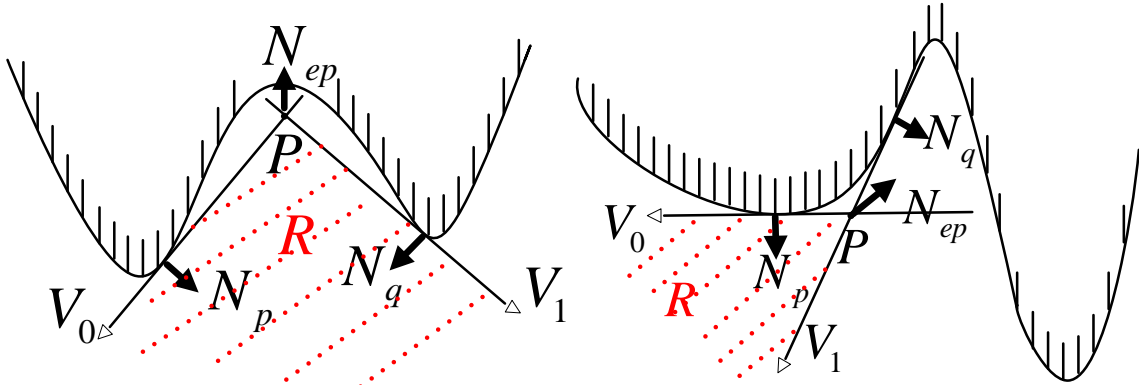


Figure 6.4: Locally Convex Carving. The left figure represents the case that the intersected tangent lines coming from two different convex surface and the contour normal are opposite from each other, and in this case we cannot carve out the Region  $R$ . The right one is what we defined as locally convex carving cases where the Region  $R$  can be carved. All the tangent lines in these figures indicate internal contours.  $N_p$  and  $N_q$  are the contour normals and  $N_{ep}$  is the normal for the epipolar plane.

In summary, our *locally convex carving* is defined in Algorithm 1. The two **if** statements represents the two rules.  $T_o$  is the threshold of the angle between surface normal computed from occluding contours and epipolar plane. Also in practice we have used contours from neighboring viewpoints to do convex carving which is more reliable.

Figure 6.5 shows the effect with and without locally convex carving. While the area carved out may not be that significant, it actually identifies locally convex points, which are usually next to concave areas. These points are critical in our concave fitting step that will be introduced in Section 6.3.

---

**Algorithm 1** Locally Convex Carving

---

```
for pixel  $p$  in  $IC_0$  do  
    find its correspondence  $q$  in  $IC_1$  using the epipolar constraint;  
    triangulate 3D point  $P$  with  $p$  and  $q$ ;  
    Estimate the surface normal of  $p$  and  $q$ , denoted as  $N_p$  and  $N_q$ ;  
    Compute the angle between  $N_p$  and  $N_q$  with epipolar plane ( $N_{ep}$ ), denoted as  $R_p$  and  $R_q$ ;  
    if  $\text{dot}(\text{cross}(N_p, N_{ep}), \text{cross}(N_q, N_{ep})) < 0$  then  
        continue;  
    else if  $R_p < T_o$  or  $R_q < T_o$  then  
        continue;  
    else  
        carve out  $R$ ;  
    end if  
end for
```

---

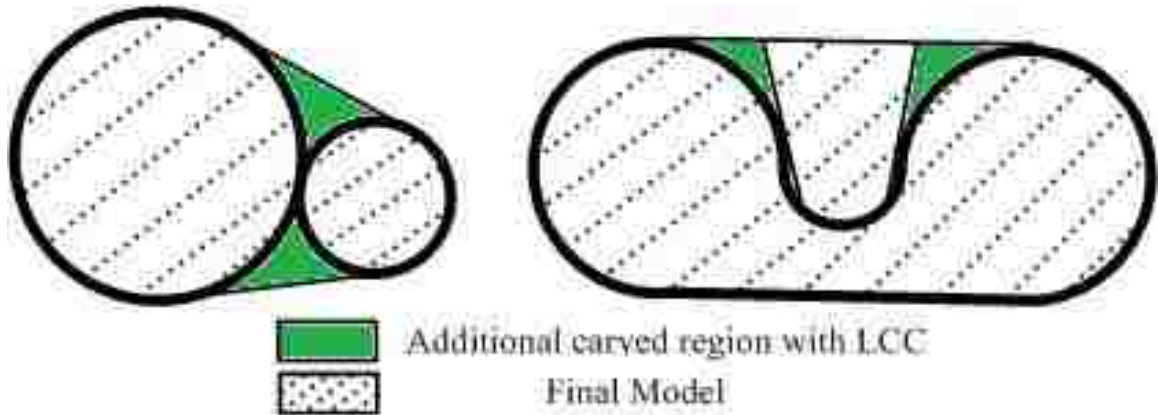


Figure 6.5: Locally convex carving Illustration of convex and concave cases. The figure presents the two cases that illustrate the additional regions that can be carved out with LLC and also the region that we will get after the locally convex carving operation.

## 6.2 Internal Contour Tracking

We have described our locally convex carving approach to refine the visual hull using internal contours. The problem now is how to get the internal contours on the images. While there are previous methods on contour tracking and detection (e.g., [73, 40, 123]), none of them addresses highly specular surfaces in our case. We present a semi-automatic approach for detecting the internal contours with contours labeled in a few key frames.

Given labeled contours from key frames, our algorithm is designed to interpolate the corresponding contours between these frames. Let us assume that we have contours  $C_{i-n}$  and  $C_{i+m}$  in Image  $I_{i-n}$  and  $I_{i+m}$  respectively, the goal is to detect the corresponding

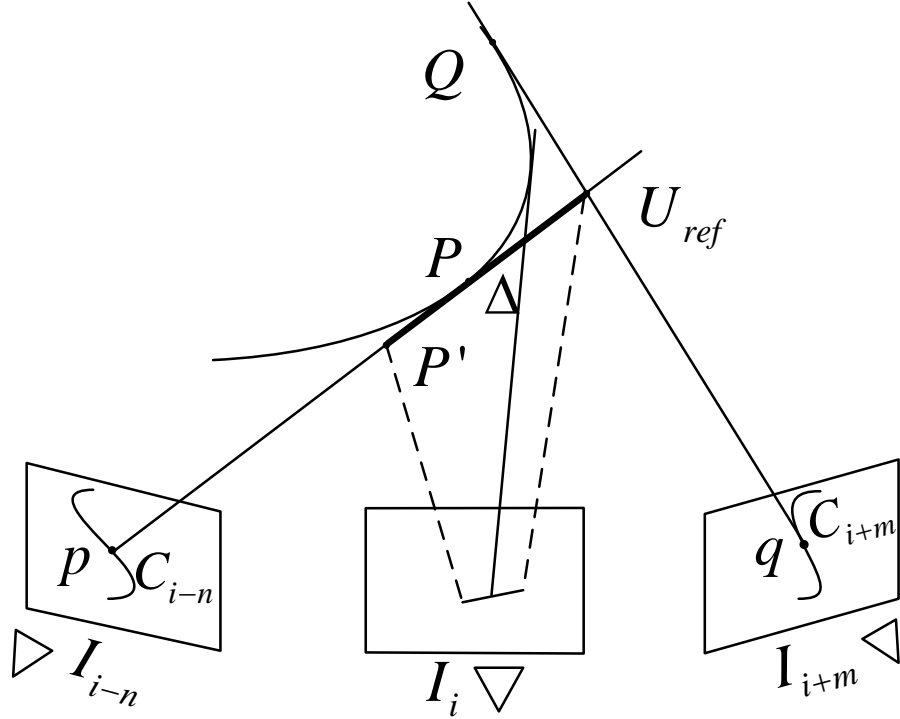


Figure 6.6: Geometric illustration for contour prediction. See more details in the text.

contours in Image  $I_i$ .

We first intersect  $C_{i-n}$  and  $C_{i+m}$  to obtain a set of 3D points, which provides a proxy about where the contour points should be in 3D. The use of the 3D proxy to guide contour detection is the main difference from typical contour tracking/detection formulations. Let  $p$  in  $C_{i-n}$  and  $q$  in  $C_{i+m}$  denote the pair of corresponding pixels and  $U_{ref}$  be the reconstructed 3D point. We also denote the contour points on the surface as  $P$  and  $Q$ . Assuming the surface between  $P$  and  $Q$  is smooth, then the possible contour points for  $I_i$  must be between  $P$  and  $Q$ . This is similar to the order constraint that is often used in stereo matching. Unfortunately we really do not know where  $P$  and  $Q$  are. However,  $U_{ref}$  can be used as loose upper bound for  $Q$ . For the lower bound, we approximate it with a user-defined parameter  $\Delta$ , which defines a 3D point  $P'$  on the back-projection ray for  $p$ , but  $\Delta$  distance away from  $U_{ref}$ . The line between the lower bound  $P'$  and the upper bound  $U_{ref}$  provides the searching space in 3D. Projecting  $P'$  and  $U_{ref}$  to image  $I_i$  defines the possible candidates for a contour point in  $I_i$ .

Once all the candidates for one contour are identified, we develop a global optimization approach to detect the contour points. We formulate it as a labeling problem with a data consistency term and two regularization terms. To create a uniform label set for each

contour point, the 3D line segment between  $P'$  and  $U_{ref}$  is uniformly divided to a set of discrete points  $\{L_k\}$ , each representing a unique label.

### 6.2.1 Data term

The data term expresses the likelihood of point  $L_k$ , which projects to a particular pixel  $u_k$  in Image  $I_i$  that is a contour pixel. It is defined as the weighted sum of two sub-terms as described below.

**Gradient term**  $D_g$  is the gradient term, which favours the contour point to pass through pixels with strong gradient. It can be computed as:

$$D_g(k) = \exp(-\lambda_k G(u_k)^2) \quad (6.1)$$

$$\lambda_k = \omega_1 \mathbf{V}(u_k) \cdot \mathbf{V}(p) + \omega_2 \mathbf{V}(u_k) \cdot \mathbf{V}(q) \quad (6.2)$$

where  $G(\cdot)$  is the gradient magnitude at a given pixel location and  $\mathbf{V}(\cdot)$  is the gradient direction.  $\lambda_k$  is used to preserve the direction of the contour. It considers the differences of the gradient directions between predicted contours and two reference contours.  $\omega_1 = m/(m+n)$  and  $\omega_2 = n/(m+n)$  are the interpolation factors.

**Histogram of intensity term**  $D_h$  is the intensity matching term computed with histogram. It is assumed that the corresponding occluding contours in consecutive frames have similar color distributions (see Figure 6.7 for some examples).  $D_h$  is computed as follows:

$$D_h(k) = \exp(-H(I_i(u_k))^2) \quad (6.3)$$

Where  $H$  is the intensity probability function computed from the intensity of pixels of  $C_{i-m}$  and  $C_{i+n}$ . In essence, we calculate a histogram of all the pixels in  $C_{i-m}$  and  $C_{i+n}$ . For  $u_k$ 's intensity, we look at the corresponding bin to find its normalized probability.

With these terms defined, the data term for one contour point  $u$  can be defined as

$$D(u) = \sum_k (\omega_g D_g(k) + \omega_h D_h(k)), \quad (6.4)$$

where  $\omega_g$  and  $\omega_h$  are weighting factors. The data term for an entire contour,  $D(C_i)$ , is the sum of  $D(u)$  for all  $u$  in  $C_i$ . We use the number of pixels in  $C_{i-n}$  to discretize  $C_i$ .

### 6.2.2 Regularization terms

To preserve the smoothness of the detected contour, we introduce two regularization terms. To abuse the notation a bit, we denote a contour point in  $I_i$  as  $u_j$ , it is different from  $u_k$ ,



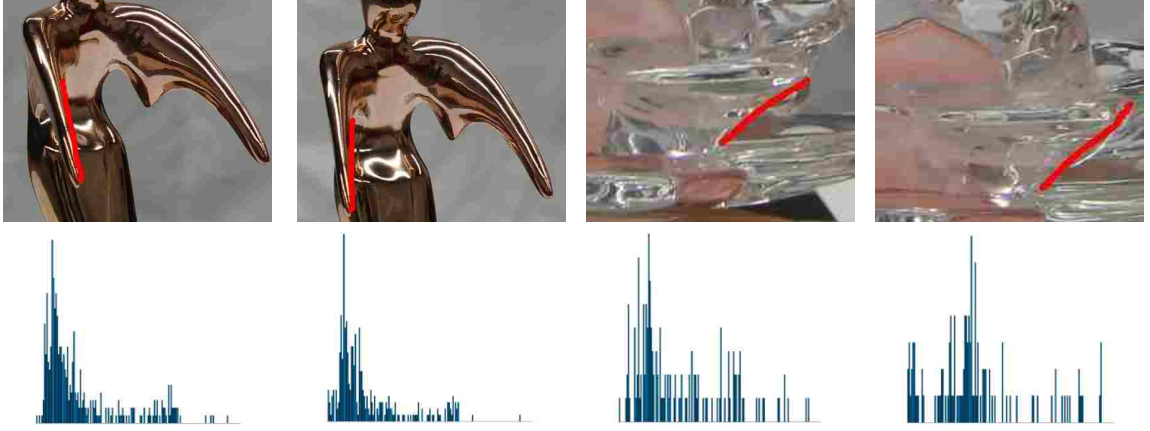


Figure 6.7: Intensity histogram. The upper row is captured images with internal contours highlighted in red, and the second row is the histogram of corresponding contours. Images in the first and second column are from nearby viewpoints and so as the third and fourth column.

which is a candidate for one contour point.  $u_j$  has two neighbors  $u_{j-1}$  and  $u_{j+1}$ . The first term  $V(u_j)$  penalizes large spatial distance of neighboring pixels.

$$V(u_j) = V(u_j|u_{j-1}) + V(u_j|u_{j+1}) \quad (6.5)$$

$$V(u_j|u_{j\pm 1}) = 1 - \exp(-\|u_j - u_{j\pm 1}\|_2^2/\sigma_v^2) \quad (6.6)$$

where  $\sigma_v^2$  is a normalization factor.

The second term  $T(u_j)$  is to preserve the shape of  $C_i$  to be similar with reference contours  $C_{i-n}$  and  $C_{i+m}$ . We denote the corresponding contour points as  $p_j$  and  $q_j$ . We formulate in a way to preserve the Laplacian vector of  $u_j$ . Since a contour is a 1D entity, the Laplacian coordinate of  $u_j$  is calculated as  $\mathbf{L}(u_j) = u_j - \frac{1}{2}(u_{j-1} + u_{j+1})$ . Then  $T(u_j)$  is expressed in the following:

$$T(u_j) = \Delta(\mathbf{L}(u_j), \mathbf{L}(p_j)) + \Delta(\mathbf{L}(u_j), \mathbf{L}(q_j)), \quad (6.7)$$

$$\Delta(\mathbf{L}(p), \mathbf{L}(q)) = 1 - \exp(-\|\mathbf{L}(p) - \mathbf{L}(q)\|_2^2/\sigma_t^2) \quad (6.8)$$

where  $\sigma_t^2$  is again a normalization factor.

### 6.2.3 Energy Function

Finally we get the energy function to be minimized as follows,

$$E(Ci) = D(Ci) + \sum_j (\lambda_v V(u_j) + \lambda_t T(u_j)) \quad (6.9)$$

where  $\lambda_v$ , and  $\lambda_t$  are weighting terms. We have used the SRMP [70] to solve the high-order optimization problem with multiple labels. One contour is labeled each time.

### 6.3 Concave refinement

The locally convex carving (LCC) algorithm presented in Section 6.1.2 is able to reconstruct the convex part of the object surface revealed by internal occluding contours. However It cannot recover concave part since the tangent lines of a concave surface is lying inside the surface. We have developed a simple surface fitting method to estimate the concave part with some user interactions.

The basic idea is to fit a concave surface based on its boundary that can be correctly reconstructed. We first allow the user to mark a concave area. This is done in the image space. An image in which the concavity is most frontal is chosen to allow the user to mark up the concave region  $RC$ . Boundary points near  $RC$  serve as the seed points, denoted as  $RS$ . Many points in  $RS$  can be automatically identified since there is usually a transition from convex to concave, the convex part can be carved out with our LCC algorithm. A user can also include additional boundary points to give a tighter control. Figure 6.8 shows the concave boundary.

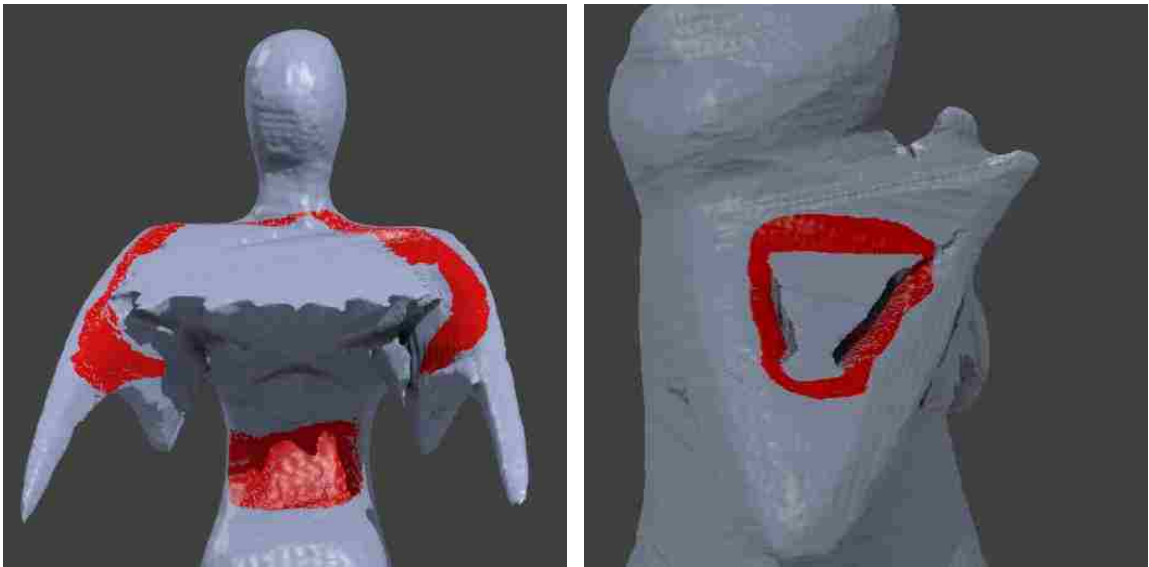


Figure 6.8: Concave boundary illustration. The two images illustrate the boundary cues used in concave fitting for these two models. The projection of these 3D boundary vertices is  $RV$  area in the frontal 2d image.

Then we try to propagate the boundary depth to the concave part under the smoothness

constraints. We will estimate a depth value ( $d_p$ ) for every pixel in  $RC$ . The energy function that needs to be minimized can be expressed as follows:

$$E_d = \lambda_c \sum_{p \in RV} \|d_p - \tilde{d}_p\|_2^2 + \lambda_{r1} \sum_{\substack{p,q \in RC \\ q \in N(p)}} \|d_p - d_q\|_2^2 + \lambda_{r2} \sum_{\substack{p,q \in RC \\ q \in N(p)}} \|\delta d_p - \delta d_q\|_2^2 \quad (6.10)$$

where  $\lambda_c$ ,  $\lambda_{r1}$  and  $\lambda_{r2}$  are the weights for corresponding terms and  $N(p)$  defines the four neighbors of pixel  $p$ . The first data term measures the depth difference between the known point  $\tilde{d}_p$  and the depth value of the reconstructed point  $d_p$ . The second and third term enforce the smoothness constraints with  $\delta d_p$  stands for the depth gradient.

The above energy function can be formulated as a linear least square system for which the global optimum can be efficiently computed. Once we get depth map for  $RC$ , we will use it to carve out any volume that is in the concave part of the reconstructed surface. That leads to the final model.

## 6.4 Experiments and Results

We evaluate the proposed method on four challenging objects, consisting of shiny specular objects (*statue*, *trophy*, and *frog*) and one transparent object made of glass (*lotus*). Forty-five 2000\*3000 images were captured for each object when it was placed on a checkerboard pattern (for pose calculation). The object silhouettes were extracted using Grab-cut [117].

### 6.4.1 Contour Tracking Results

We have verified our contour tracking method on the four datasets. Several contours were scribbled first in a set of key frames. Based on the complexity of objects, generally we need to label internal contours in every 5-7 consecutive images (see supplementary materials). From these labelled contours on the key frames, we ran our contour tracking algorithm. The parameters were set to  $\Delta = 20mm$ ,  $w_g = w_h = 0.8$ ,  $\sigma_v = 2.0$ ,  $\sigma_t = 2.5$ , and  $\lambda_v = 1.2$ ,  $\lambda_t = 1.0$ . These values were tuned empirically and remained fixed for all four data sets.

Figure 6.9 shows some qualitative comparisons with gradient + smoothness only method. As we can see in the first row, for *statue* and *lotus* (right two) the detected contours can be snapped to highlights where the gradient is really strong. With our proposed terms, we can still get good results in these cases, as shown in the second row. For *frog* (left), comparable results are archived since it is not as specular as the other two.

**Quantitative Evaluation** We further manually labelled all the images to quantify the accuracy of our tracking results. We calculate the mean distance (in pixels) between the tracked

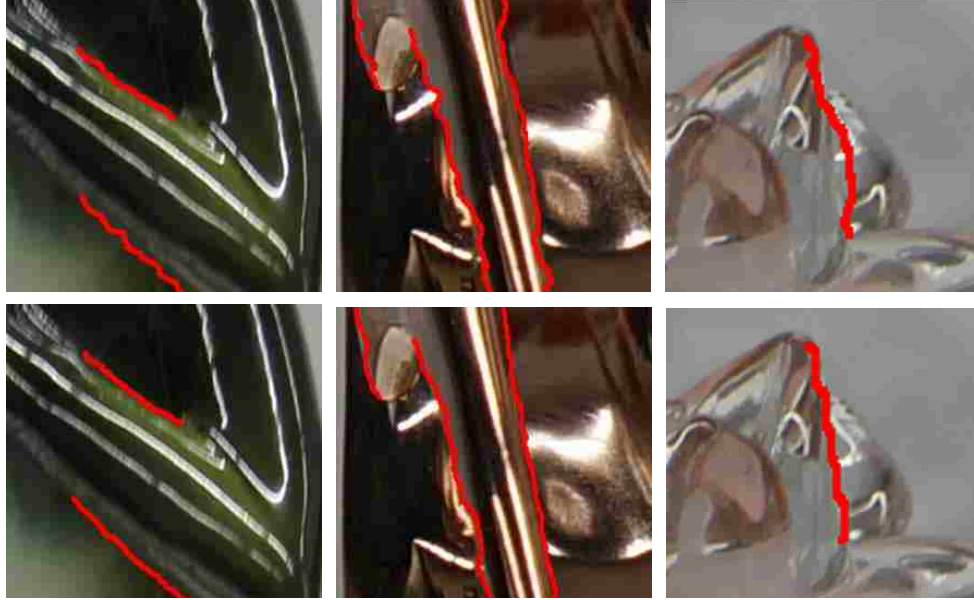


Figure 6.9: Comparison of contour tracking. The detected pixels are highlighted in images. The first row shows the results using only the gradient data term and the pairwise smoothness term. The second row shows the results with all the data terms and regularization terms.

contour and the labelled data, which are considered as the ground truth. The numbers are shown in Table 6.1. It shows the effect of different terms. With only the gradient term we cannot get pleasant results especially for statue and lotus. As we integrate the proposed data terms and regularization terms, the mean pixel error has been reduced to half.

Table 6.1: Mean error of tracked contours. The table gives the mean pixel error on four datasets and the rows indicate the terms that were incorporated. G donates the gradient term data (Eq. 6.1) and IH is the term using histogram of color intensity (Eq. 6.3); P and T are two regularization terms of Eq. 6.5 and Eq. 6.7 respectively.

	Statue	Lotus	Trophy	Frog
G	2.3178	2.3516	1.9448	1.7657
G+IH	1.3902	1.6106	1.2256	1.1423
G+IH+P	1.2503	1.3961	1.1028	0.9869
G+IH+P+T	<b>1.1221</b>	<b>1.2324</b>	<b>1.0509</b>	<b>0.9381</b>

## 6.4.2 Reconstruction Results

We present our final reconstructed 3D models in Figure 6.10 with a comparison to visual hull reconstruction. The threshold  $T_o$  in the LCC algorithm are chosen from 30 to 45

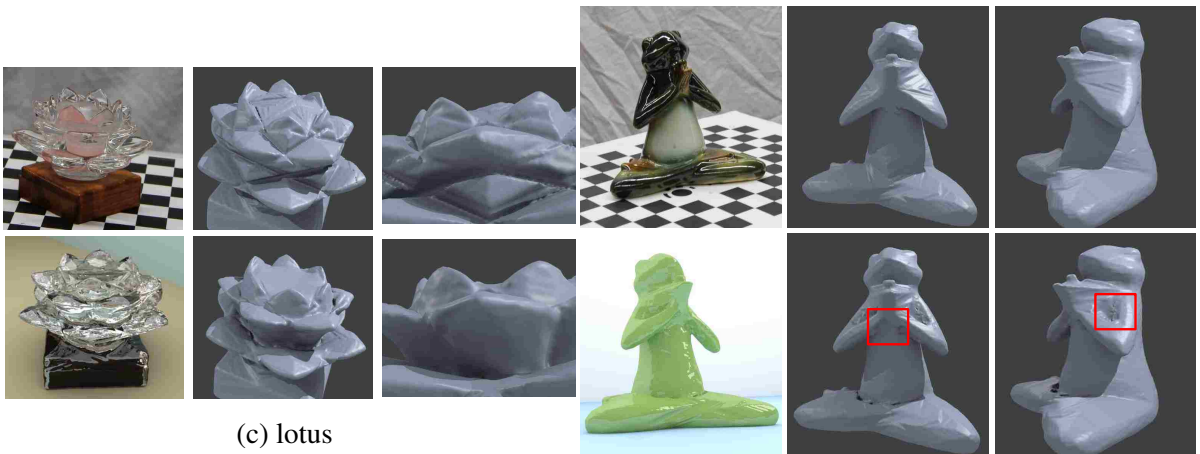
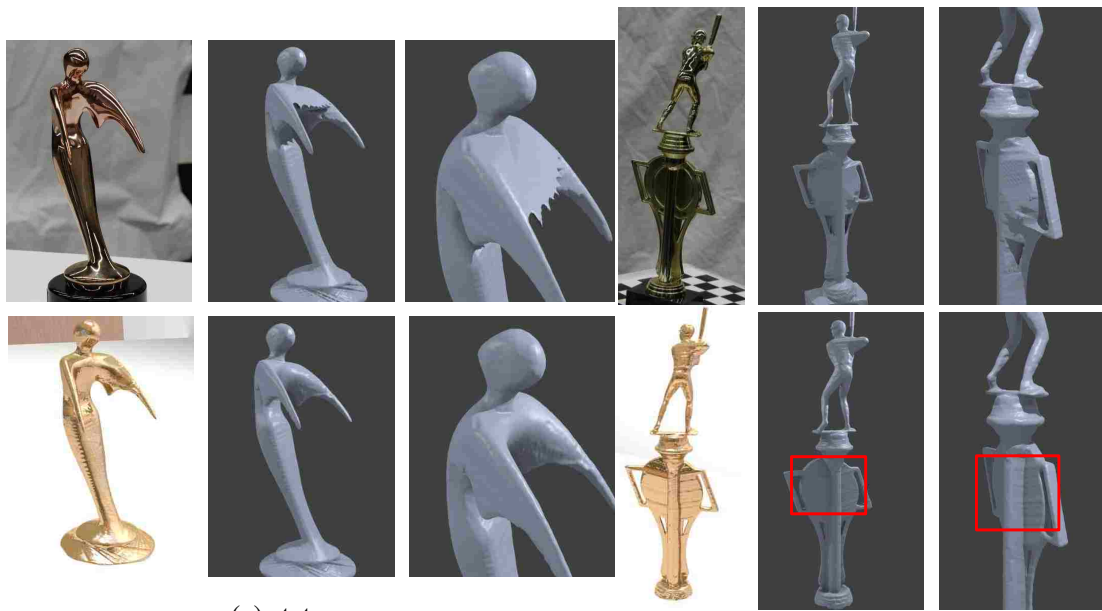


Figure 6.10: Comparison of reconstructed 3d models. For each object, the first row shows the original image from one viewpoint and reconstructed visual hull surface, and the second row shows the rendered and reconstructed model with our proposed method.

degrees. As for the weighting parameters for each term in the fitting formulation, we use  $\lambda_c = 0.5$ ,  $\lambda_{r1} = 0.2$ , and  $\lambda_{r2} = 0.3$  for the four models. We need some user guidance for our interactive system and it will take about twenty minutes to get the model.

For the statue model, the wings of the statue are refined with our LCC method which are tighter to the real surface than the original visual hull model. Also, the concave part between the two wings is reconstructed with our concave fitting method.

For the lotus model, most parts of the petals are actually convex, therefore our LCC method can successfully reconstruct the petals. Original visual hull method, on the other hand, is not able to get this. The concave part on the top of the model is fitted to have smooth transition with the top layer petals.

For the frog model, we are able to carve out the locally convex part along the left/right arms and also the belly under the hands, which provides the boundary and gradient propagation cues for concave fitting. The fitted surface has preserved the tendency of the surface.

For the trophy model, the pillar of the model is completely reconstructed with our method. No concave fitting is performed on this model.

**Limitations** There are still some limitations in our method. Our concave fitting method tends to under fit concave areas. A better user interaction method is needed, such as push of surfaces. Contour tracking on highly specular objects are still very challenging, in particular for areas with small details, such as the face of the baseball player on the trophy. These are difficult for even our human eye to see. Overall our method is better suited for reconstruction of organic shapes without detailed surface relief patterns. Finally image segmentation on glass objects is very difficult, even with user interactions. It probably requires a more controlled setup.

## 6.5 Conclusion

This chapter addressed the problem of multi-view surface reconstruction for non-Lambertian objects. Instead of using active lights or other specialized setup, we aim to use multi-view stereo capture setup under general unknown illumination. Our visual-hull-based reconstruction approach exploits internal contours inside the objects silhouettes to generate a tighter surface model. Given the internal contours, our novel locally convex carving algorithm is able to carve out extra voxels in convex part of the surface. Also, this carving operation provides important boundary cues for fitting concave areas that do not have any contours. Our second contribution is a novel approach for tracking internal contours given contours labeled in sparse key frames. It is designed specifically for highly specular or transparent objects, for which assumptions made in traditional contour

detection/tracking methods are no longer valid. We have validated our methods, both quantitatively and qualitatively, with four datasets of different object materials. Results show that we are able to generate visually pleasing models for very challenging cases.

# Chapter 7

## Summary

In this chapter, I summarize the contributions, limitations of our methods and future research directions.

### 7.1 Conclusions

First I have proposed an approach to recover surface details and its albedo map from an RGB-D video sequence. The basic idea is to exploit the photometric information in the color sequence using the lighting variations induced by casual object movement. The object was rotated freely in front of the camera and we presented a lighting insensitive local match refinement approach to find correspondences along the sequence, after which the surface normal and albedo were recovered in a pixel-wise manner without relying on any regularization. In this way, we can resolve the texture-copy problem which has not been effectively treated in previous shading refinement approach. We believe that this is an interesting discovery that can be extended to multi-view stereo to recover fine details.

In addition to improving the spatial resolution of the depth sensor, I also focused on improving its temporal resolution to capture slow motion in 3D. I have presented a hybrid camera system that combined a high-speed color camera and a consumer depth sensor to generate a high-speed depth sequence. A novel framework was developed that utilized both shading constraints within each frame and optical flow information between neighboring frames. The high speed depth sequence was recovered in a single optimization which was formulated by taking the depth information captured from low-speed depth camera as the boundary constraints and preserving smooth motion field between the depth maps of neighboring frames. The problem of enhancing the temporal resolution of the depth sensor has not received much attention in the computer vision community compared to expanding



the spatial resolution and depth map denosing. And I hope this work can inspire more thoughtful insights on this topic.

Another contribution is that I have developed an approach to build up 3D human avatars with sparse frames using a single RGBD camera. It is a challenging task because of the almost inevitable non-rigid motion and also surface occlusion. I have accomplished this by taking advantage of a generative human template (the SMPL model) to guide the alignment of those pieces from the sparse frames. It becomes quite convenient and easy to get human avatars with our fully automatic method. One biggest problem for current learning based techniques of human shape reconstruction is lack of groundtruth detailed human shapes. The reconstructed human models have rich surface details and clear texture maps, which can serve as the groundtruth model to train a deep neural network for human body estimation.

Last but not the least, I go beyond the lambertian surface assumption and use standard multi-view images to reconstruct the 3D surface of specular and transparent objects. A new visual hull refinement scheme – *Locally Convex Carving* was proposed which can completely reconstruct concavity caused by two or more intersecting convex surfaces. Although this *Locally Convex Carving* is used for the reconstruction of specular and transparent objects, it does not stop it from applying on the general objects which will provide more constraints on surface reconstruction.

## 7.2 Limitations and Extension

For the detailed surface reconstruction approach proposed in Chapter 3, it works well on convex objects that come with a whole piece of surface, but it is challenging to deal with objects that have great concavities and discontinuities since it will be difficult to compute a continuous warping field to get precise alignment due to the self-occlusion. More importantly the concave part will probably get occluded during the capture as we rotate the object. In this case, we will not be able to collect enough evidence of the surface under different lighting condition, therefore the surface normal might not converge to its optimal value. We should definitely conduct further investigation on the reconstruction of more sophisticated objects. Also, our method fails on objects that are too small since the fusion step will fail in the first place. I have used Kinect V2 in our implementation which was supposed to have the best performance in a distance range from 0.5m to 4.5m and it means we cannot let the object be too close to the camera. Therefore, a possible solution is to use a depth sensor that works well in the near range. Another limitation is that we assume the environment lighting does not change abruptly during the capture. The formulation

will become invalid in this case as the changes of pixel intensity are not just caused by the object rotation. We will have to take both lighting changes and object motion into account. As a future work, we could take advantages of the changing lighting conditions in our formulations since more lighting variation will reveal more valuable information about the surface. Finally, right now we are mainly focusing on depth enhancement, while we would like to implement all these procedures in full 3D space and recover a complete model.

For the temporal upsampling method presented in Chapter 4, I have found a few limitations of our method during experiments. First, there are still some artifacts at the discontinuous boundary of the recovered depth which is caused by the motion blur effect of the captured depth sequence. This might be resolved using the normal constraints along the contours on the boundary. Secondly, we have not dealt with the occlusions and topology changes explicitly in our formulation, therefore the artifacts exist in our recovered depth caused by the occlusions and topology changes. In future work, I plan to detect the occlusions via bi-directional matching and track the pixel or 3D point in the whole sequence and try to deal with the occlusions and topology changes. The most important drawback of the proposed method is that it involves bundle optimization of the sequence which has great computational cost and is time consuming. To resolve this problem, we could take advantage of the current deep learning techniques and train a deep neural network to predict the in-between depth frames. First, we will need to build up a dataset and capture various objects in motion with an RGBD camera which could be taken as groundtruth high speed depth. We generate the input depth frames by downsampling of the captured sequence. To build up the network, the basic idea is to have a network to predict the flow between neighboring frames and adopt the Spatial Transformer Network to predict the depth frames as guided by the flow field.

For the human shape reconstruction in Chapter 5, one assumption of the proposed method is that the human subject is wearing clothes that are relatively tight. As we want to deform the captured pieces onto the SMPL model which is a naked human model template, it will pose great challenges with human subject with loose clothes or even in a dress. Therefore, it will be necessary to explicitly recover the shapes of the garment and infer its physical properties [132]. This could be achieved by parsing the clothes from the RGBD frame and fitting the surface with a garment template that is controlled by its type and physical parameters [154]. After the modeling of both human body and clothes, we would also be able to perform tasks like virtual try-on. As an immediate extension to this work, it would be useful if we could model the exact shape of the hands. For now we assume clenched fists and have not dealt with the hands particularly.

For the non-Lambertian surface reconstruction method proposed in Chapter 6, although

good results have been achieved for some cases, the reconstructed models are still lack of geometric details. Theoretically, the proposed local convex carving approach could be applied on any internal contour. However, it is extremely difficult if not impossible to detect the internal contours on surface regions that have small and rich geometric details. Therefore, our method works well on objects with smooth surface. To extend our approach to work on more sophisticated objects, one possible solution is to take the current reconstructed model as an initial estimation and recover the environmental lighting, surface normals and reflectance properties iteratively. For example, for specular objects we could exploit more sophisticated BRDF model (Bidirectional Reflectance Distribution Function) [92] to express the relationship between surface normal, reflectance and lighting.

### 7.3 Future work

To conclude, in this dissertation, I have focused on 3D surface reconstruction, which is an important topic that keeps growing recently in computer vision. I have exploited the depth sensor together with high quality color cameras to recover 3D shapes and corresponding appearance of objects from an RGB/D sequence. In the future, I would like to continue my journey on the reconstruction topic and work on more complex scenarios where objects as well as humans are not standalone but have interactions with each other. For example, currently the human modeling method is designed for a single person, but reconstructing multiple human subjects that interact with each other would be more challenging as we consider the significant occlusions and complex topology as well as interactions between multiple human subjects. We could start from the modeling of each human subject separately, infer the interactions between them, and then refine the 3D body shapes iteratively. I believe this could be a new trend in human modeling as we cannot separate ourselves completely from the environment; instead we usually have interactions with other people and also with the surrounding objects.

For another long-term future work, I want to explore the possibilities of involving conventional approaches on surface reconstruction into the learning procedure. The advantage of learning based approaches is that we could recover 3D shapes of objects from a single or a few RGB images. However, the reconstructed shapes are still quite rough with limited surface details. Besides, learning based techniques perform well on images and objects spanned by the training set, but it is not clear how these methods would perform on a completely unseen object/image categories. On the contrary, the conventional methods reconstruct the surface by exploiting the geometric or photo-consistency constraints contained in the images which will have the results that best explain the 2D observations.

Since we explicitly optimize for the agreement of the model with respect to image features, we will get a good fit. But the optimization tends to be very slow and is quite sensitive to the choice of initialization. In contrast, regression-based methods, that use a deep network to directly estimate the model parameters from pixels, tend to provide reasonable, but not pixel accurate results. Therefore, we could formulate the learning and conventional approach in a loop to fine-tune the network in an self-supervised manner. Basically, we could get reasonable good initial estimation from the learning procedure, and after that we could refine the shapes using conventional approaches to get better results that fit closely to the input data, which will serve as better groundtruth models to train a neural network for the learning module. This strategy has been adopted recently in human body estimation [71] and I believe it could be extended into many tasks.

# Bibliography

- [1] Altizure. <https://www.altizure.com/>. 7
- [2] Mocap. <http://mocap.cs.cmu.edu/>. 50
- [3] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjarholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016. 7
- [4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 35
- [5] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 7
- [6] E. Aguiar, C. Stoll, C. Theobalt, and et al. Performance capture from sparse multi-view video. In *ACM Transactions on Graphics*, page 98, 2008. 2, 10, 11
- [7] Benjamin Allain, Jean-Sébastien Franco, and Edmond Boyer. An efficient volumetric framework for shape tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 268–276, 2015. 10
- [8] John Aloimonos. Shape from texture. *Biological cybernetics*, 58(5):345–360, 1988. 6
- [9] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. 11
- [10] J. T. Barron and J. Malik. High-frequency shape and albedo from shading using natural image statistics. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2521–2528, 2011. 10

- [11] J. T Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):690–703, 2016. 41
- [12] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, 2015. 10
- [13] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *European Conference on Computer Vision*, pages 617–632. Springer, 2016. 9
- [14] Bruce Guenther Baumgart. Geometric modeling for computer vision. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1974. 2
- [15] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):509–522, 2002. 2
- [16] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 561–578, 2016. 11
- [17] M. Böhme, M. Haker, T. Martinetz, and E. Barth. Shading constraint improves accuracy of time-of-flight measurements. *Computer Vision and Image Understanding*, 114(12):1329–1335, 2010. 9
- [18] Edmond Boyer and Marie-Odile Berger. 3d surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233, 1997. 6
- [19] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011. vi, 18, 34
- [20] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Probabilistic deformable surface tracking from multiple videos. In *European conference on computer vision*, pages 326–339. Springer, 2010. 10
- [21] Q. Chen and V. Koltun. A simple model for intrinsic image decomposition with depth cues. In *IEEE International Conference on Computer Vision*, pages 241–248, 2013. vii, 29, 31, 41

- [22] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. *arXiv preprint arXiv:1908.04422*, 2019. 2
- [23] Ian Cherabier, Johannes L Schonberger, Martin R Oswald, Marc Pollefeys, and Andreas Geiger. Learning priors for semantic 3d reconstruction. In *Proceedings of the European conference on computer vision (ECCV)*, pages 314–330, 2018. 2
- [24] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015. 8
- [25] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644, 2016. 7
- [26] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1
- [27] Yan Cui, Sebastian Schuon, Derek Chan, Sebastian Thrun, and Christian Theobalt. 3d shape scanning with a time-of-flight camera. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1173–1180. IEEE, 2010. 7
- [28] Brian Curless. Overview of active vision techniques. In *Proc. SIGGRAPH*, volume 99, 2000.
- [29] D. Zarpalas D. S. Alexiadis and P. Daras. Real-time full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras. *IEEE Trans. Multimed.*, 15(2):339358, 2013. 48
- [30] I. Baran P. Debevec J. Popovic S. Rusinkiewicz D. Vlastic, P. Peers and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Transactions on Graphics*, 28(5):111, 2009. 48
- [31] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(3):24, 2017. 8

- [32] Yuanyuan Ding, Li Feng, Ji Yu, and Jingyi Yu. Dynamic fluid surface acquisition using a camera array. In *IEEE International Conference on Computer Vision*, 2011. 8, 60
- [33] Pengfei Dou, Shishir K Shah, and Ioannis A Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5908–5917, 2017. 2
- [34] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *IEEE International Conference on Robotics and Automation*, pages 1691–1696. IEEE, 2012. 2
- [35] C. H. Esteban, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008. 13
- [36] M. Loper F. Bogo, M. J. Black and J. Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *IEEE International Conference on Computer Vision*, page 23002308, 2015. 11
- [37] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rther, and H. Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *IEEE International Conference on Computer Vision*, pages 993–1000, 2013. 9
- [38] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rütter, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 993–1000, 2013. 9
- [39] G .Finlayson and R. Xu. Illuminant and gamma comprehensive normalisation in logrgb space. *Pattern Recognition Letters*, 24(11):1679–1690, 2003. 19
- [40] Li Gang, Yanghai Tsin, and Yakup Genc. Exploiting occluding contours for real-time 3d tracking: A unified approach. In *IEEE International Conference on Computer Vision*, 2007. 65
- [41] Dariu M Gavrilă and Larry S Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Proceedings cvpr ieee computer society conference on computer vision and pattern recognition*, pages 73–80. IEEE, 1996. 11
- [42] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21, 1987. 50



- [43] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499, 2016.
- [44] S Burak Gokturk, Hakan Yalcin, and Cyrus Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 35–35. IEEE, 2004. 3
- [45] Leonardo Gomes, Olga Regina Pereira Bellon, and Luciano Silva. 3d reconstruction methods for digital preservation of cultural heritage: A survey. *Pattern Recognition Letters*, 50:3–14, 2014. 1
- [46] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014. 2
- [47] B. Haefner, Y. Quéau, T. Möllenhoff, and D. Cremers. Fight ill-posedness with ill-posedness: Single-shot variational depth super-resolution from shading. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 164–174, 2018. vii, 10, 26, 27, 28, 30
- [48] Y. Han, J. Lee, and I. Kweon. High quality shape from a single rgb-d image under uncalibrated natural illumination. In *IEEE International Conference on Computer Vision*, pages 1617–1624, 2013. 10
- [49] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *IEEE International Conference on Computer Vision*, pages 1586–1594, 2017. 7
- [50] Gleen Healey and Thomas O Binford. Local shape from specularities. *Computer Vision, Graphics, and Image Processing*, 42(1):62–86, 1988. 6
- [51] E. Herbst, X. Ren, and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *International Conference on Robotics and Automation*, pages 2276–2282, 2013. 39
- [52] Darko Hercog, Bojan Gergic, Suzana Uran, and Karel Jezernik. A dsp-based remote control laboratory. *IEEE Transactions on Industrial Electronics*, 54(6):3057–3068, 2007. 1

- [53] Aaron Hertzmann and Steven M. Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1254–1264, 2005. 60
- [54] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 7
- [55] B. K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, MIT, 1970. 2, 10
- [56] Kai-Lung Hua, Kai-Han Lo, and Yu-Chiang Frank Wang. Extended guided filtering for depth map upsampling. *IEEE MultiMedia*, 23(2):72–83, 2015. 9
- [57] Chun-Hao Huang, Benjamin Allain, Jean-Sébastien Franco, Nassir Navab, Slobodan Ilic, and Edmond Boyer. Volumetric 3d tracking by detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3862–3870, 2016. 10
- [58] David A Huffman. Impossible object as nonsense sentences. *Machine intelligence*, 6:295–324, 1971. 2
- [59] L. Liu Z. Pan J. Tong, J. Zhou and H. Yan. Scanning 3d full human bodies using kinects. *IEEE Trans. Vis. Comput. Graph.*, 18(4):643650, 2012. 48
- [60] J. Jeon, S. Cho, X. Tong, and S. Lee. Intrinsic image decomposition using structure-texture separation and surface normals. In *European Conference on Computer Vision*, pages 218–233, 2014. vii, 29, 31, 41
- [61] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfacerNet: An end-to-end 3d neural network for multiview stereopsis. In *IEEE International Conference on Computer Vision*, pages 2307–2315, 2017. 7
- [62] Roland S Johansson, Göran Westling, Anders Bäckström, and J Randall Flanagan. Eye–hand coordination in object manipulation. *Journal of Neuroscience*, 21(17):6917–6932, 2001. 1
- [63] T. Yu X. Liu Q. Dai K. Guo, F. Xu and Y. Liu. Real-time geometry, albedo and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics*, 36(3):32, 2017. 2, 12

- [64] Y. Wang Y. Liu K. Guo, F. Xu and Q. Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *IEEE International Conference on Computer Vision*, page 30833091, 2015. 11
- [65] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7122–7131, 2018. 2, 11
- [66] Sagi Katz and Avishai Adler. Depth camera based on structured light and stereo vision, March 8 2012. US Patent App. 12/877,595. 7
- [67] Tejas Khot, Shubham Agrawal, Shubham Tulsiani, Christoph Mertz, Simon Lucey, and Martial Hebert. Learning unsupervised multi-view stereopsis via robust photometric consistency. *arXiv preprint arXiv:1905.02706*, 2019. 7
- [68] K. Kim, A. Torii, and M. Okutomi. Joint estimation of depth, reflectance and illumination for depth refinement. In *IEEE International Conference on Computer Vision Workshops*, 2015. 10
- [69] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):78, 2017. 7
- [70] Vladimir Kolmogorov. A new look at reweighted message passing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5):919–930, 2015. 69
- [71] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. *arXiv preprint arXiv:1909.12828*, 2019. 79
- [72] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23, 2015. 1
- [73] Kiriakos N. Kutulakos and Charles R. Dyer. Occluding contour detection using affine invariants and purposive viewpoint control. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 356–361, 1994. 65
- [74] A. Gudym L. Luo J. Barron L. Hao, E. Vouga and G. Gusev. 3d self-portraits. *ACM Transactions on Graphics*, 32(6):187, 2013. 12, 48, 56

- [75] A. Lakdawalla and A. Hertzmann. Shape from video: Dense shape, texture, motion and lighting from monocular image streams. In *Proceedings of the First International Workshop on Photometric Analysis For Computer Vision*. INRIA, 2007. 13
- [76] Aldo Laurentini. The visual hull of curved objects. In *IEEE International Conference on Computer Vision*, pages 356–361, 1999. 62
- [77] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM transactions on Intelligent Systems and Technology (TIST)*, 4(4):58, 2013. 2
- [78] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *European Conference on Computer Vision*, pages 154–169. Springer, 2016. 9
- [79] Yu Li, Dongbo Min, Minh N Do, and Jiangbo Lu. Fast guided global interpolation for depth and motion. In *European Conference on Computer Vision*, pages 717–733. Springer, 2016. 9
- [80] Junbang Liang and Ming C Lin. Shape-aware human pose and shape reconstruction using multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4352–4362, 2019. 12
- [81] J6 Agila Bitsch Link, Paul Smith, Nicolai Viol, and Klaus Wehrle. Footpath: Accurate map-based indoor navigation using smartphones. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–8. IEEE, 2011. 1
- [82] Ding Liu, Xida Chen, and Yee-Hong Yang. Frequency-based 3d reconstruction of transparent and specular objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–667, 2014. 8, 60
- [83] Miaomiao Liu, Kwan Yee Kenneth Wong, Zhenwen Dai, and Zhihu Chen. Pose estimation from reflections for specular surface recovery. In *IEEE International Conference on Computer Vision*, 2011. 8
- [84] Ming-Yu Liu, Oncel Tuzel, and Yuichi Taguchi. Joint geodesic upsampling of depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 169–176, 2013. 9
- [85] H. Fuchs M. Dou and J. M. Frahm. Scanning and tracking dynamic objects with commodity depth cameras. In *IEEE Symposium on Mixed and Augmented Reality*, page 99106, 2013. 48

- [86] H. Fuchs H A. Fitzgibbon M. Dou, J. Taylor and S. Izadi. 3d scanning deformable objects with a single rgb-d sensor. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 493–501, 2015. 12, 48
- [87] M. Niener C. Theobalt M. Innmann, M. Zollhfer and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision*, pages 362–379, 2015. 2, 8, 12, 48
- [88] J. Romero G. Pons-Moll M. Loper, N. Mahmood and M. J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248, 2015. 11, 48, 49
- [89] C. Rehmann C. Zach M. Zollhfer, S. Izadi and et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics*, 33(4):156, 2014. 10
- [90] R. Maier, K. Kim, D. Cremers, J. Kautz, and M. Nießner. Intrinsic3d: High-quality 3d reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *IEEE International Conference on Computer Vision*, pages 3114–3122, 2017. 10
- [91] Satya P. Mallick, Todd Zickler, David J. Kriegman, and Peter N. Belhumeur. Beyond lambert: Reconstructing specular surfaces using color. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 619–626, 2005. 8
- [92] Satya P Mallick, Todd E Zickler, David J Kriegman, and Peter N Belhumeur. Beyond lambert: Reconstructing specular surfaces using color. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 619–626. Ieee, 2005. 78
- [93] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976. 2
- [94] David Marr and Tomaso Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156):301–328, 1979. 2
- [95] Dimitris Metaxas and Demetri Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993. 11

- [96] D. Min, J. Lu, and M. N Do. Depth video enhancement based on weighted mode filtering. *IEEE Transactions on Image Processing*, 21(3):1176–1190, 2012. 9
- [97] Dongbo Min, Jiangbo Lu, and Minh N Do. Depth video enhancement based on weighted mode filtering. *IEEE Transactions on Image Processing*, 21(3):1176–1190, 2011. 9
- [98] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 2
- [99] Shree K Nayar and Yasuo Nakagawa. Shape from focus. *IEEE Transactions on Pattern analysis and machine intelligence*, 16(8):824–831, 1994. 2
- [100] Diego Nehab, Tim Weyrich, and Szymon Rusinkiewicz. Dense 3d reconstruction from specular consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 60
- [101] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011. 8, 14, 26
- [102] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):169, 2013. 8
- [103] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 International Conference on 3D Vision (3DV)*, pages 484–494. IEEE, 2018. 11
- [104] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. *Proceedings of IEEE International Conference on 3D Vision (3DV)*, 2018. 11, 12
- [105] Roy Or-El, Guy Rosman, Aaron Wetzler, Ron Kimmel, and Alfred M Bruckstein. Rgb-d-fusion: Real-time high precision depth recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5407–5416, 2015. vii, 10, 23, 24, 26, 27, 28, 30, 40

- [106] J. Park, H. Kim, Y. Tai, M. S. Brown, and I. Kweon. High-quality depth map up-sampling and completion for rgb-d cameras. *IEEE Transactions on Image Processing*, 23(12):5559–5572, 2014. 9
- [107] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S Brown, and Inso Kweon. High quality depth map upsampling for 3d-tof cameras. In *2011 International Conference on Computer Vision*, pages 1623–1630. IEEE, 2011. 9
- [108] Poser. Poser. <https://www.posersoftware.com/>. 56
- [109] David Punter. Shape and shadow: on poetry and the uncanny. *A New Companion to the Gothic*, pages 252–264, 2012. 6
- [110] M. Ye Q. Zhang, B. Fu and R. Yang. Quality dynamic human body modeling using a single low-cost depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 676683, 2014. 11
- [111] D. Fox R. Newcombe and S. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015. 2, 10, 12, 48
- [112] J. Schmid R. W. Sumner and M. Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics*, 26(3):80, 2007. 52
- [113] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *ACM SIGGRAPH*, pages 497–500, 2001. 15, 35
- [114] Fabio Remondino and Sabry El-Hakim. Image-based 3d modelling: a review. *The photogrammetric record*, 21(115):269–291, 2006. 6
- [115] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 2
- [116] Andrea Romanoni and Matteo Matteucci. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. *arXiv preprint arXiv:1903.10929*, 2019. 7
- [117] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. 23, 70
- [118] C. Du R. Wang-J. Zheng S. Wang, X. Zuo and R. Yang. Dynamic non-rigid objects reconstruction with a single rgb-d sensor. *Sensors*, 18(3):886, 2018. 48

- [119] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv preprint arXiv:1905.05172*, 2019. 2
- [120] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 6
- [121] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017. 7
- [122] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006. 2, 6
- [123] Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M. Seitz. Occluding contours for multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4002–4009, 2014. 65
- [124] J. Shi, Y. Dong, X. Tong, and Y. Chen. Efficient intrinsic image decomposition for rgb-d images. In *ACM VRST*, pages 17–25, 2015. 35
- [125] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000. 2
- [126] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 2
- [127] D. Simakov, D. Frolova, and R. Basri. Dense shape reconstruction of a moving object under arbitrary, unknown lighting. In *IEEE International Conference on Computer Vision*, volume 3, pages 1202–1209, 2003. 13
- [128] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6040–6049, 2017. 7



- [129] Greg Slabaugh, Ron Schafer, Tom Malzbender, and Bruce Culbertson. A survey of methods for volumetric scene reconstruction from photographs. In *Volume Graphics 2001*, pages 81–100. Springer, 2001. 1
- [130] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 1, 2
- [131] F. Xu Y. Dong-Z. Su T. Yu, K. Guo and et al. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 910–920, 2017. 8, 12, 48
- [132] K. Guo T. Yu, Z. Zheng and et al. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7287–7296, 2018. 12, 48, 77
- [133] J Tan, Ignas Budvytis, and Roberto Cipolla. Indirect deep structured learning for 3d human body shape and pose prediction. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 3, page 6, 2017. 11
- [134] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. 2
- [135] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5236–5246, 2017. 11, 12
- [136] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979. 2, 6
- [137] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. 2018. 11, 12
- [138] David L Waltz. Generating semantic descriptions from drawings of scenes with shadows. 1972. 2

- [139] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 7
- [140] Michael Waschbüsch, Stephan Würmlin, Daniel Coting, Filip Sadlo, and Markus Gross. Scalable 3d video of dynamic scenes. *The Visual Computer*, 21(8-10):629–638, 2005. 2
- [141] Sebastian Weik. A passive full body scanner using shape from silhouettes. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 750–753. IEEE, 2000. 6
- [142] Michael Weinmann, Aljosa Osep, Roland Ruiters, and Reinhard Klein. Multi-view normal field integration for 3d reconstruction of mirroring objects. In *IEEE International Conference on Computer Vision*, 2013. 8, 60
- [143] Yang Wen, Bin Sheng, Ping Li, Weiyao Lin, and David Dagan Feng. Deep color guided coarse-to-fine convolutional network cascade for depth image super-resolution. *IEEE Transactions on Image Processing*, 28(2):994–1006, 2018. 9
- [144] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. *Robotics: Science and Systems*, 2015. 8
- [145] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016. 2
- [146] C. Wu, Y. Liu, Q. Dai, and B. Wilburn. Fusing multiview and photometric stereo for 3d reconstruction under uncalibrated illumination. *IEEE Transactions on Visualization and Computer Graphics*, 17(8):1082–1095, 2011. 11
- [147] C. Wu, K. Varanasi, Y. Liu, H. Seidel, and C. Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In *IEEE International Conference on Computer Vision*, pages 1108–1115, 2011. 10
- [148] C. Wu, M. Zollhöfer, M. Niener, M. Stamminger, S. Izadi, and C. Theobalt. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics*, 33(3), 2014. 10

- [149] Shihao Wu, Hui Huang, Tiziano Portenier, Matan Sela, Daniel Cohen-Or, Ron Kimmel, and Matthias Zwicker. Specular-to-diffuse translation for multi-view reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 183–200, 2018. 8
- [150] T. Wu and C. Tang. Photometric stereo via expectation maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):546–560, 2010. 21
- [151] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang. Color-guided depth recovery from rgb-d data using an adaptive autoregressive model. *IEEE Transactions on Image Processing*, 23(8):3443–3458, 2014. 9
- [152] Jingyu Yang, Xinchun Ye, Kun Li, and Chunping Hou. Depth recovery using an adaptive color-guided auto-regressive model. In *European conference on computer vision*, pages 158–171. Springer, 2012. 9
- [153] Qingxiong Yang, Narendra Ahuja, Ruigang Yang, Kar-Han Tan, James Davis, Bruce Culbertson, John Apostolopoulos, and Gang Wang. Fusion of median and bilateral filtering for range image upsampling. *IEEE Transactions on Image Processing*, 22(12):4841–4852, 2013. 9
- [154] Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C Lin. Physics-inspired garment recovery from a single-view image. *ACM Transactions on Graphics (TOG)*, 37(5):170, 2018. 77
- [155] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision*, pages 767–783, 2018. 2, 7
- [156] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019. 7
- [157] L. Yu, S. Yeung, Y. Tai, and S. Lin. Shading-based shape refinement of rgb-d images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1422, 2013. 10
- [158] T. Simon S. Wei Z. Cao, G. Hidalgo and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv*, page arXiv:1812.08008, 2018. 50

- [159] L. Zhang, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multiview stereo. In *IEEE International Conference on Computer Vision*, pages 618–625, 2003. 13
- [160] Q. Zhang, M. Ye, R. Yang, Y. Matsushita, B. Wilburn, and H. Yu. Edge-preserving photometric stereo via depth fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2479, 2012. 22
- [161] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999. 2, 6
- [162] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 3
- [163] Q. Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics*, 33(4):155, 2014. 20
- [164] Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. Elastic fragments for dense scene reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 473–480, 2013. 8
- [165] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4491–4500, 2019. 2, 11
- [166] Todd E. Zickler, Peter N. Belhumeur, and David J. Kriegman. Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, 49(2-3):215–227, 2002. 8
- [167] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum*, volume 37, pages 625–652. Wiley Online Library, 2018. 2

# Vita

## Xinxin Zuo

### Education and Professional Experience

---

- Jan. 2017 – present: Ph.D student at Gravity Lab, Department of Computer Science, University of Kentucky. (Advisor: Dr. [Ruigang YANG](#))  
Graduate Research Assistant  
General GPA: 4.0/4.0
- Sep. 2012 – Dec. 2016: PhD student in Computer Science and Technology, NPU. (Advisor: Dr. [Jiangbin ZHENG](#))  
General GPA: 88.20/100, Rank: 1/21
- Oct. 2014 – Oct. 2016: Joint Ph.D student at Gravity Lab, Department of Computer Science, University of Kentucky. (Advisor: Dr. [Ruigang YANG](#))
- 2011.9 – 2014.3: Master in Computer Application Technology, NPU. (Advisor: Dr. [Jiangbin ZHENG](#))  
General GPA: 91.35/100, Rank: 1/142
- 2007.9 - 2011.7: Bachelor in Computer Science and Technology, NPU.  
General GPA: 92.03/100, Rank: 1/174

### Publications

---

1. **Xinxin Zuo**, Sen Wang, Jiangbin Zheng, Zhigeng Pan, Ruigang Yang. Detailed Surface Geometry and Albedo Recovery from RGB-D Video Under Natural Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019. (Accepted)
2. Hao Zhu, **Xinxin Zuo**, Sen Wang, Xun Cao, Ruigang Yang. Detailed Human Shape Estimation from a Single Image by Hierarchical Mesh Deformation. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019: 4491-4500. (Oral)
3. Sen Wang\*, **Xinxin Zuo\***, Chao Du, Runxiao Wang, Jiangbin Zheng, Ruigang Yang. Dynamic Non-Rigid Objects Reconstruction with a Single RGB-D Sensor. *Sensors*, 2018, 18(3): 886.

4. **Xinxin Zuo\***, Sen Wang\*, Jiangbin Zheng, Ruigang Yang. Detailed Surface Geometry and Albedo Recovery from RGB-D Video Under Natural Illumination. In *IEEE International Conference on Computer Vision (ICCV)*, 2017: 3133-3142.
5. Sen Wang, **Xinxin Zuo**, Runxiao Wang, Fuhua Cheng, Ruigang Yang. A Generative Human-Robot Motion Retargeting Approach using a Single Depth Sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017: 5369-5376. (**Spotlight**)
6. **Xinxin Zuo**, Sen Wang, Jiangbin Zheng, Ruigang Yang. High-speed Depth Stream Generation from a Hybrid Camera. In *ACM International Conference on Multimedia (ACM MM)*, 2016: 878-887. (**Oral**)
7. **Xinxin Zuo**, Chao Du, Sen Wang, Jiangbin Zheng, Ruigang Yang. Interactive Visual Hull Refinement for Specular and Transparent Object Surface Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2015: 2237-2245.
8. Jiangbin Zheng, **Xinxin Zuo**, Jinchang Ren, Sen Wang. Multiple Depth Maps Integration for 3D Reconstruction using Geodesic Graph Cuts. *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, 2015, 25(3):473-492.

## **Selected Awards & Scholarship**

---

- IEEE CVPR Doctoral Consortium, 2019
- Thaddeus B. Curtz Memorial Scholarship, University of Kentucky, 2018
- ACM student travel grants (for *ACM MM*), 2016
- The National Scholarship, 2008-2011.
- Excellent Bachelor's Degree Thesis, NPU, 2011
- Outstanding Graduate Student, NPU, 2011.
- Excellent Student Scholarship, First Prize, 2008-2011, 2013-2015.

## **Service**

---

- Reviewer for
  - International Journal of Computer Vision (IJCV)
  - IEEE TPAMI, IEEE TIP
- Reviewers of
  - CVPR 2016-2020, ICCV 2017, 2019; ECCV 2018, 2020; AAAI 2020  
BMVC 2019; Pacific Graphics 2016, 2019; ACCV 2018