Indoor Depth Map Completion via Mesh-based Discrete Geometric Processing

Zhenshan Liang

A Thesis

in

The Department

of

Mechanical, Industrial & Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

January 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Zhenshan Liang**

Entitled: **Indoor Depth Map Completion via Mesh-based Discrete Geometric Processing**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Hang Xu

_____ External Examiner
Dr. Yunbo Zhang

_____ Examiner
Dr. Hang Xu

_____ Supervisor
Dr. Tsz Ho Kwok

Approved by _____
Martin D. Pugh, Chair
Department of Mechanical, Industrial & Aerospace Engineering

_____ 2024

_____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Indoor Depth Map Completion via Mesh-based Discrete Geometric Processing

Zhenshan Liang

Depth measurement serves as a pivotal technology with widespread applications, but limitations of commodity depth sensors often result in noisy and incomplete depth maps for indoor scenes. In response, this study introduces a discrete geometric processing-based method aimed at enhancing the completeness of indoor depth maps, particularly in intricate and challenging scenes. The proposed approach involves transforming raw depth maps into a quadrilateral mesh surface and addressing missing depth information through a mesh deformation framework. The mesh deformation framework employs iterations to update constraints on quadrilateral facets until convergence, tackling normal constraints, position constraints, and perspective projection constraints. This optimization problem is resolved using a combination of local shaping and global fusing strategies. In the local step, each quadrilateral facet is projected from the physical imaging plane into camera coordinate system based on perspective projection, preserving only orientation information. The global step then propagates location information through vertices shared by connected facets. Uncertainties in input normal and observed depth information are addressed through updating strategies. Experimental results on 30 selected examples from ScanNet dataset demonstrate the superiority of the proposed method over two existing techniques quantitatively and qualitatively. Specifically, the method excels in addressing depth discontinuity around object boundaries, producing clear disconnections, while competing methods exhibit errors and stretched distortions. Validating normal and depth updating strategies, the paper confirms the effectiveness of processing depth discontinuity in reducing completion errors. The robustness of the method is further affirmed by energy reductions in both normal and position aspects across iterations.

Acknowledgments

I am sincerely grateful to express my deepest appreciation to my supervisor, Dr. Tsz Ho Kwok, for his unwavering support and exceptional academic guidance throughout my master's journey. Dr. Kwok's expertise, patience, and encouragement have played a pivotal role in shaping the trajectory of my research and guiding me through the intricate challenges of academic exploration.

Beyond his academic wisdom, I have found inspiration in the motivational quotes featured on his personal website, particularly "If you really want to do something, you'll find a way. If you don't, you'll find an excuse." and "It is a good day if you have learned something new." These insightful words have been a source of motivation and reflection, contributing significantly to my personal and academic growth.

I would also like to extend my heartfelt gratitude to my parents for their unwavering support and sympathetic ear. Their encouragement, understanding, and unyielding belief in my abilities have been instrumental in reaching this significant milestone. Their unwavering support has provided me with the strength and determination needed to navigate the challenges of academia.

Additionally, I want to express my gratitude to my friends and research colleagues, Christopher Danny-Matte, Eder da Silva Sales, Ankhy Sultana, and Newton Andoh, for their sincere assistance. Engaging in discussions with them and benefiting from their valuable suggestions has deepened my understanding of technical details and enriched the overall quality of my research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Scope of the Research Proposal

Depth measurement plays a pivotal role in a wide range of applications, including autonomous driving, 3D scene reconstruction, augmented reality, and human-computer interaction [1]. These applications have the potential to significantly advance smart manufacturing by integrating vision systems and collaborative robots for Industry 4.0 [2]. Also, they can be divided into indoor and outdoor applications based on the objects being captured, and this work focuses on indoor depth measurement, which always have smaller depth range, such as ScanNet [3], NYUv2 [4] and VOID [5], compared to outdoor scenes. Furthermore, there are some differences in the processing of depth values and the evaluation criteria between indoor scenes and outdoor scenes. Inverse depth representations are used for mixing datasets [6] and real-time performance is more important for outdoor autonomous driving [7].

The proliferation of readily available depth sensors has fueled the rapid growth of applications mentioned above, making their success contingent upon the accuracy and reliability of depth sensors. Traditional depth sensors can be broadly categorized as passive and active. The passive approach relies on rich and distinctive features in paired color (RGB) images to estimate depth for corresponding points, but it tends to perform inadequately in regions with low texture or repetitive patterns. On the other hand, the active approach, encompassing techniques like time-of-flight (ToF) and structured light (SL), overcomes some of the limitations of stereovision by projecting structured

patterns to discern the geometry of the scene. However, active methods may encounter challenges such as non-Lambertian surfaces and inherent constraints [8]. Consequently, both sensor types often yield noisy and incomplete depth maps, which are insufficient for accurate indoor 3D information for localization and reconstruction. Therefore, the completion and refinement of raw depth maps are essential to enhancing their utility.

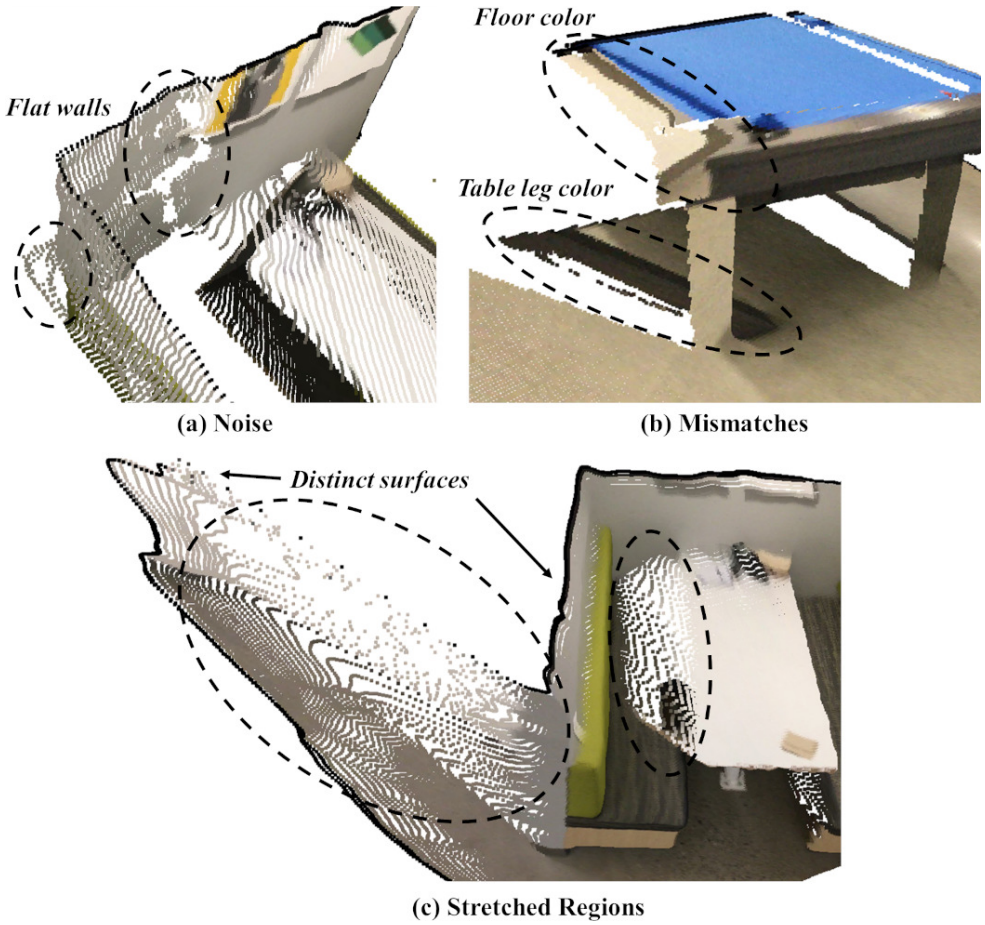


Figure 1.1: Challenges in depth completion. The colors from the RGB image are assigned to the corresponding points in the depth map.

In essence, depth completion is the process of generating a comprehensive and more precise depth map from an initially incomplete or noisy depth map. Its primary objective is to fill the void left by missing depth values and elevate the overall quality of the depth data. Depth completion methods encompass a range of techniques, including interpolation [9], machine learning [10], graph-based optimization [11], and sensor data fusion [12], to estimate and augment the missing

depth values. Sensor fusion, in particular, involves the integration of data from multiple sensors, each possessing its own strengths and weaknesses, to enhance the depth map's quality. This approach proves especially valuable when tackling intricate and challenging scenes. For instance, depth (D) cameras contribute depth information, introducing an additional dimension to the data. However, they may encounter difficulties in achieving precision, especially when dealing with highly reflective surfaces. On the other hand, RGB cameras excel at capturing high-resolution details, patterns, and textures on surfaces. Some work implicitly extracted features representation by designing specific modules and networks [10, 13, 14, 15], while others explicitly extracted geometric information [16, 17, 18], e.g., detected edges and surface normal. These diverse sensors provide complementary insights into the scene, and by leveraging their individual strengths, this approach can yield a more resilient and accurate depth map.

Nevertheless, our study has unearthed certain nuances. Even in cases where dense reconstructions are derived from RGB-D data [3], often referred to as "ground truth" in various papers, certain areas still exhibit noise and mismatches when compared with RGB images. As shown in Fig. 1.1, walls that should be flat have gaps and redundancies, while the point cloud calculated from depth map with camera intrinsics does not semantically match the corresponding RGB images well. Because the reconstruction methods proposed for these "ground truth", i.e. calibration procedure [19, 20], BundleFusion system [21], VoxelHashing [22] and some post-processing methods, still have limitations and make the results inherently different from the real scenes. Therefore, the poor quality of supervised depth maps affects the performance of data-driven methods. Even training with the data acquired in the virtual scenes [23], there is a theoretical difference between the rendered data from virtual scene and the data captured from real scene. [24] focuses on generating high-quality ground truth depth for ScanNet, but aleatoric uncertainty still exists and should be considered.

Besides, stretched regions that appears in some existing work [10, 17, 25] are caused by insufficient depth discontinuity detection and pixel processing. Some work utilized **Canny** operator [16, 17, 26, 27] or **Sobel** filter [28, 29] to detect occlusion boundary and the pixels around boundaries were assigned weights [30] as depth discontinuity cues to generate clear structures. But these detected boundaries were unstable and did not satisfy the definition of occlusion boundary [31].

(a) Example of NYUv2-OC++ [28].

(b) Example of BSDS-RIND [30].

Figure 1.2: It is ambiguous which side the annotated pixels belong to.

Although NYUv2-OC++ dataset used in SharpNet [32] and [33] provided manually annotated occlusion boundaries and the annotation idea of RINDNet [34] was consistent with C_0 discontinuity used in our work, their annotated results were still ambiguous as shown in Fig. 1.2.

1.2 Objective

Therefore, there is a requirement for a more resilient approach capable of adeptly integrating data from multiple sensors to effectively address the challenges mentioned above. We observed that numerous previous efforts aimed at enhancing the depth maps, such as denoising and utilizing surface normal [26], begin by comprehending the structure and form of objects and scenes. In essence, these improvements are rooted in the geometric properties and relationships within the data. Their limitations are that over-reliance on data statistics rather than actual objects' spatial relationships makes them have low interpretability and weak generalization ability. Consequently, we propose applying a geometry-based approach for indoor depth completion to handle noise and inconsistencies while also addressing depth discontinuities.

Our method is inherited from the existing geometric-based Surface-from-Gradients (SfG) method [35], which considers an RGB-D image a quadrangle mesh surface and poses the depth reconstruction problem as a constrained geometric optimization. This constrained geometric optimization problem is formulated and solved using the Shape-Up method [36], a local-global geometric processing method with shape constraints, whereas the shape constraints are formulated as local

operators and the global geometries are obtained through iterative global solving. Shape-up has demonstrated its supreme convergence and robustness in various applications, including support-free design [37], 4D printing [38], soft robotics [39], non-planar slicing [40], etc. In the geometric-based SfG method [35], the local operator shapes each mesh facet by enforcing the facet orientation following the computed normal vector direction, which realizes the reconstruction of the surface through a computed normal field.

Similar to the geometric-based SfG method [35], the essential step of our work is to formulate a local projection operator and proximity functions to effectively fuse depth and RGB data, simultaneously enhancing the quality of the depth data for indoor depth completion. By minimizing the proximity function, a shape that aligns with the specified constraints to the greatest extent is obtained. In addition, depth confidence and surface normal uncertainty are converted into spaces used for updating depth and updating normal respectively to guide the depth completion process [10, 29, 41, 42, 43]. The updatable spaces corresponds to their geometric interpretation and relates to the observed depth and surface normal estimated from RGB data.

1.3 Major Contributions

This paper focuses on the completion of indoor depth data obtained from RGB-D cameras. Firstly, we leverage the raw depth map to establish depth constraints in the form of position operators. Given the inherent noise and uncertainty from the scanning process [29], we use the depth confidence map, acquired from the depth sensors, to define a depth tolerance for each point, allowing for a more accurate positioning within the operator. In regions where depth information is absent, the maximum permissible tolerance is applied with an initial estimated position.

Secondly, we estimate the surface normals for each pixel using the RGB data [44] to create orientation operators. This orientation information not only enhances depth accuracy but also aids in completing areas with missing depth. Similar to the depth information, uncertainty exists in the normal estimation, prompting the utilization of the corresponding confidence map to define a tolerance for each facet of the pixel, enabling reorientation.

Furthermore, depth-discontinuity information is extracted from the RGB data [45]. To address

this, facets are disconnected, enabling the system to optimize distinct surfaces separately. Lastly, the proximity function is defined as a weighted sum of errors from both the position and orientation operators in the camera space, employing the pinhole model. The contributions of this work can be summarized as follows:

- The depth completion task is reformulated as a geometry processing problem, providing a unified methodology for fusing complementary data and addressing the issues in indoor depth completion concurrently.
- Recognizing the presence of data uncertainty, a geometric tolerance is incorporated into the optimization framework for both position and orientation. This enhances the framework's denoising capacity and augments overall accuracy.
- Leveraging features derived from the RGB data, the framework untangles facets belonging to different surfaces and trims the depth data accordingly. This enables the framework to adeptly manage discontinuity and mismatching issues without alterations.

The experimental results indicate that the proposed method outperforms other two state-of-the-art techniques [14, 17], which also leverages RGB data to improve the depth map. Our results exhibit better qualitative performance and exhibit a higher degree of consistency with the RGB data.

1.4 Outline of the thesis

The remainder of the paper is structured as follows: Section 2 provides an overview of related works in the field of depth completion. Section 3 delves into the technical intricacies of the proposed method. Section 4 presents our experimental results along with a detailed analysis. Finally, in Section 5, we conclude by summarizing the proposed method and discussing potential directions for future work.

Chapter 2

Related Work

This study pertains to the field of depth completion, and we will examine related techniques within this domain.

2.1 Sensor Data Fusion

Various sensors offer complementary depth information, enabling the creation of a more comprehensive and precise depth map through data integration. RGB images are a prevalent data source in depth completion. Certain methods have introduced specific modules [13] and network architectures [15] designed to extract transformed features from RGB images. The former proposed depth-adaptive convolution kernel for super-resolution network to filter out the invalid depth values and the sub-pixel operation [46] was applied in up-sampling layer, while the later extracted discriminative features from blurry RGB image in different levels by the proposed repetitive hourglass network to gradually guide depth completion. These methods model the depth distribution through loss functions [47, 48, 49]. Additionally, surface normals can be estimated from RGB images, providing geometric information for depth completion [16]. The proposed pipeline in [16] is a representative to take advantage of the ability of surface normal to describe the local 3D space, and it proved that surface normal and occlusion boundaries can improve the depth completion performance. Based on [16], Huang et al. [17] have enhanced performance by incorporating a self-attention mechanism

and boundary consistency. Self-attention mechanism forced networks pay more attention on critical features in high-dimensions, while boundary consistency preserves occlusion boundaries, and the results showed that there was the significant improvement on evaluation metrics compared with [16]. Ren et al. [18] extended this idea to depth structure completion, introducing surface normal constraints on flat regions and reflecting depth structures through Gaussian weights. A significant challenge lies in sensor alignment and calibration to ensure accurate fusion. The decoder modulation branch [14] utilizes a missing depth map mask to exploit spatially-dependent features, reducing domain shift between RGB images and depth maps through a technique known as spatially-adaptive denormalization (SPADE). RigNet [15] extracts discriminative features from blurry RGB images at different levels using the repetitive hourglass network and models high-frequency responses to generate clear structures near object boundaries. In the fusion process, some approaches incorporated depth confidence for guidance during both training and inference [29, 41]. Wang et al. [10] introduced a constraint network branch that assigns high confidence to accurate raw depth data. The fused confidence map serves as a complementary guide for depth completion. Liu et al. [27] employed a confidence mask to reflect the reliability of depth obtained from Multiple-View Stereo (MVS). They mitigated outliers by reducing their influence through weighted loss functions. Nevertheless, data from different sensors may introduce varying errors [42], resulting in distinct uncertainties and mismatching issues that necessitate further investigation.

2.2 Depth Spatial Propagation

By considering the interrelationships between neighboring pixels, one can disseminate depth information across an image, thus completing the areas lacking depth data. Liu et al. [47] introduced the spatial propagation network (SPN), which aimed to learn an affinity matrix for semantic similarity. Cheng et al. [49] extended this approach to the convolutional spatial propagation network (CSPN), which propagated information within a local area in all directions without sacrificing theoretical support. Subsequently, they further refined the method with CSPN++ [50]. The context-aware CSPN significantly improved depth completion accuracy by distinguishing between simple

planes (e.g., ground and walls) and complex surface boundaries. The resource-aware CSPN optimized kernel size and iterations for each pixel to reduce computational resources while maintaining comparable accuracy. To efficiently incorporate relevant neighbors and suppress unrelated ones, Park et al. [25] introduced learnable parameters that dynamically selected K neighbors for each pixel based on the RGB image and raw depth map. Liu et al. [11] pioneered a more dynamic, graph-based spatial propagation in 3D space. They utilized changeable graphs with estimated affinity patches to propagate information effectively, employing a self-attention mechanism. DySPN [51] further refined the spatial propagation model by generating different affinity matrices during iterations to account for long-range dependencies. This approach also addressed over-smoothing issues by considering the affinity matrix between the current refined result and the previous one. However, it's worth noting that spatial propagation may not excel at handling discontinuities and can sometimes result in stretched regions at the boundary.

2.3 Depth Residual Learning

Assuming the accuracy of the observed depth, depth completion can be seen as a regression problem, employing residual reconstruction mapping to estimate invalid depth [52]. The primary characteristic of depth residual learning involves adding the interpolated dense depth map to the output of the designed network through a global skip connection to calculate the completed depth map. Liao et al. [53] expanded linear interpolation results from filtered laser scan readings along the gravity direction to create a "reference depth" map. Conversely, Chen et al. [54] and Hegde et al. [28] utilized nearest neighbor interpolation for acquiring a dense depth map. Subsequently, the resultant interpolated map was combined with RGB images as the input for the depth residual network, encouraging the network to predict residual depth. However, the hand-crafted interpolation kernels employed in these methods are data-independent and heuristic. To address these limitations, Liu et al. [9] introduced differentiable kernel regression to fully exploit the image through end-to-end learning. The interpolated dense depth map ensured a lower bound on the final results' quality. Nonetheless, most of these approaches considered rendered depth from mesh or inpainted depth as ground truth, such as in ScanNet [3] and NYU Depth v2 [4]. Unfortunately, even the ground truth

depth map includes noisy areas, missing depth regions, and mismatched parts compared to RGB images [55]. Hence, our objective is to create a method that is not reliant on the training data.

Chapter 3

Methodology

3.1 A Geometric-Processing-Based Method

We considered the depth completion problem a mesh deformation in geometric processing. Similar to Xie et. al.'s work [35], the input depth image was converted into a mesh surface M, and at each pixel, a quadrangular facet was constructed. We then generated a normal field, with each of the facets having a normal vector associated with it. The computation of missing depth was formulated as a constrained mesh deformation where the normal vector at each facet and the observed depth information were constraints. Another constraint came from the pin-hole camera model, which constrained the deformation of each facet to follow the perspective projection in the pin-hole model. The details about the pin-hole camera model and the quadrilateral mesh representation of the depth image are as follows.

3.1.1 Pin-hole Camera Model

We adopted the state-of-the-art pin-hole camera model [56] to realize the transformation of a 3D scene into a depth image. In the pin-hole model, a 3D point $\mathbf{p}$ with coordinates (x, y, z) in the camera coordinate system was projected onto the physical imaging plane through a perspective projection, resulting in a point $\mathbf{p}'$ with coordinates $(\frac{fx}{z}, \frac{fy}{z}, -f)$ on the physical imaging plane, where f was the focal length between the optical center $\mathbf{o}$ and the image plane as shown in Fig. 3.1. In fact, the physical imaging plane was always placed in front of the optical center for convenience,

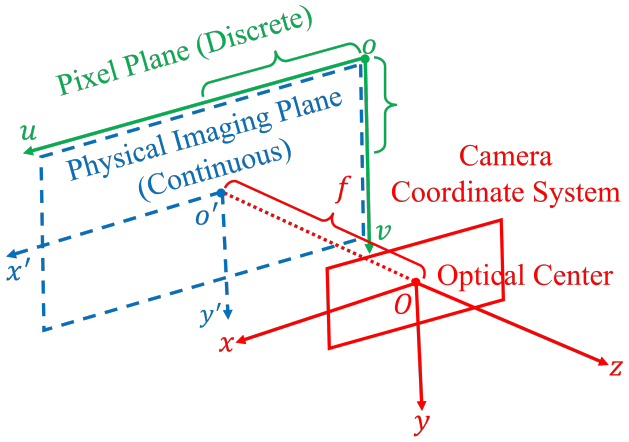


Figure 3.1: The pin-hole model.

which was at the positive direction of the z axis and the coordinates of point $\mathbf{p}'$ is $(\frac{fx}{z}, \frac{fy}{z}, f)$. The pixel coordinates (u, v) of $\mathbf{p}'$ can be further obtained through the scaling α along u axis and β along v axis, and the translation c_x along u and c_y along v, which can be represented as: $u = \alpha x' + c_x$, and $v = \beta y' + c_y$. The whole process can be rewritten into a matrix multiplication formation, as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{z}\mathbf{K}\mathbf{p} \tag{1}$$

where $f_x = \alpha f$, $f_y = \beta f$, and $\mathbf{K}$ was also defined as camera intrinsic matrix.

3.1.2 Quadrilateral Mesh Representation of Depth Image

For each pixel of the depth image with the pixel coordinates (i, j), a quadrilateral facet $f_{i,j}$ was constructed with four vertices $\mathbf{v}_{i,j}$, $\mathbf{v}_{i+1,j}$, $\mathbf{v}_{i+1,j+1}$ and $\mathbf{v}_{i,j+1}$. Each vertex $\mathbf{v}_{i,j}$ was initially positioned on the physical imaging plane with the coordinates $(\frac{(i-0.5)-c_x}{\alpha}, \frac{(j-0.5)-c_y}{\beta}, f)$ based on the pin-hole camera model as shown in Fig. 3.2. The original depth image in the image space was thus converted to a mesh surface $M = (V, F, E)$ in Euclidean space, where V, F, and E represent the sets of vertices, facets, and edges.

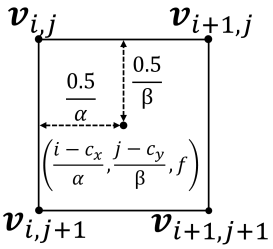


Figure 3.2: Points on the physical imaging plane.

3.1.3 Formulation for Constrained Optimization

To complete the missing depth, the mesh surface M was deformed by moving each vertex $\mathbf{v}_{i,j}$ and therefore reshaping each facet $f_{i,j}$ to fulfill the target normal vector $\mathbf{n}_{i,j}$. At some facets, there were observed depth values associated, which can be considered as spatial anchors to constrain the deformation. The last constraint came from the pinhole camera model, which required that the vertex did not move freely in space but only moved along the ray formed by the camera center $\mathbf{o}$ and the vertex's projection on the imaging plane (see Fig. 3.3). A straightforward formulation of this

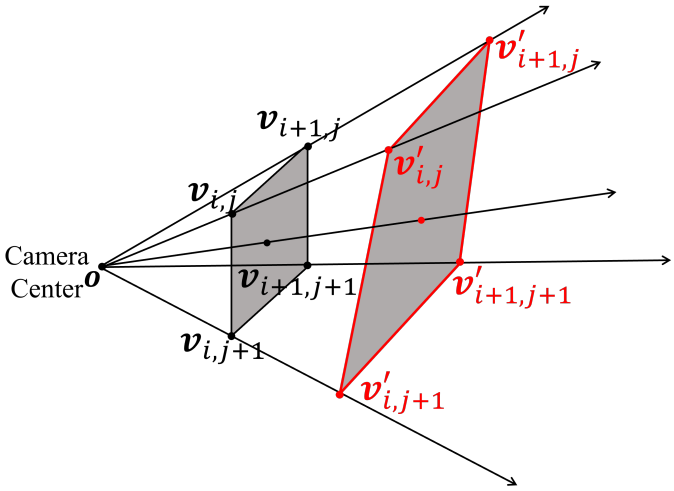


Figure 3.3: Points in the camera coordinate system.

problem was to minimize the shape variation of M and enforced the normal, position, and pin-hole

projection constraints as follows:

$$\min_{\{d_{i,j}\}} E(M) \quad \text{s.t.} \quad \mathbf{n}(f_{i,j}) = \mathbf{n}_{i,j} \quad \forall f \in F$$
$$d(f) = d_{ob} \quad \forall f \in F_{ob} \tag{2}$$
$$\mathbf{ov} \times \mathbf{ov}' = 0 \quad \forall \mathbf{v} \in V$$

where $E(M)$ was a functional measuring the shape variation (and/or smoothness) of M, $\mathbf{n}(\cdots)$ returned the normal vector of a facet, $\mathbf{d}(\cdots)$ returned the depth value of a facet, and F_{ob} was the set of facets with observed depth values. The third constraint enforced that the movement of each vertex of M during the deformation followed the pin-hole camera model's perspective projection by ensuring the vector $\mathbf{ov}$ ($\mathbf{o}$ was the camera center and $\mathbf{v}$ was the initial vertex on the physical imaging plane) and the vector $\mathbf{ov}'$ ($\mathbf{v}'$ was the deformed vertex) were collinear. According to existing research works [57, 58], directly solving this non-linear optimization suffered from slow convergence and large distortions in the resultant shape. Moreover, the initial shape of M did not satisfy the normal constraints at each facet, which made the convergence of the optimization even more challenging. Therefore, we followed previous work [35] to solve the optimization problem in an iterative local-global manner.

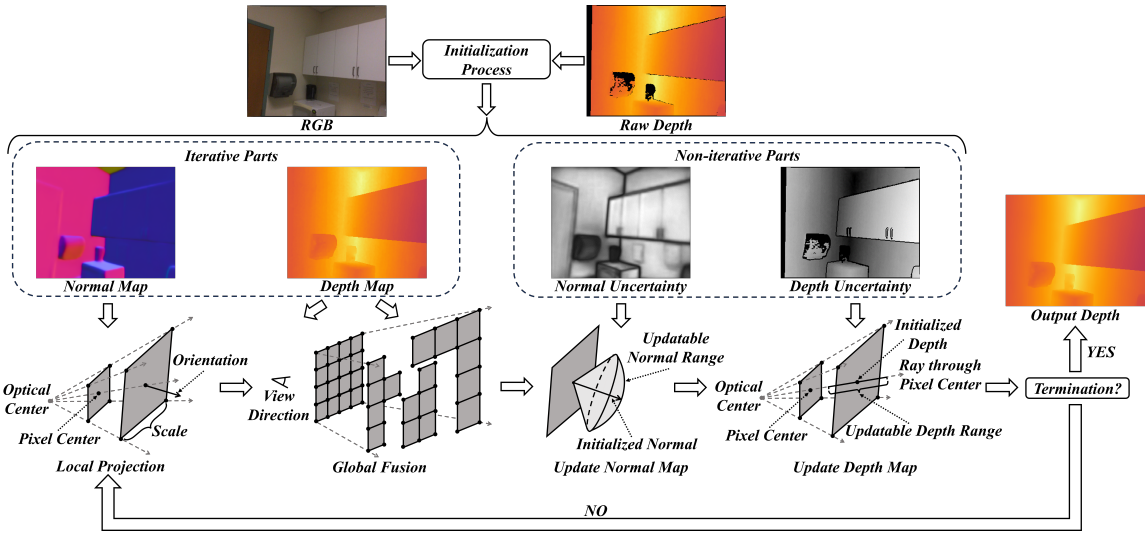


Figure 3.4: Overview of the local-global optimization.

3.1.4 Local/Global Solution

We proposed a local-global optimization approach to iteratively compute the missing depth information. Before starting the iteration, a normal field containing normal vectors associated with each facet $f_{i,j}$ was computed from the RGB image. The mesh edges, which were at the boundary of two regions with two discontinuous depths, were detected and utilized to trim M. Another initialization was a global Laplacian diffusion for generating initial depth values considered as initial scale information at the facets with missing depth values.

In the proposed local-global optimization process, each facet was shaped locally according to the three constraints, including normal, position, and pin-hole perspective projection constraints. Afterward, global linear system solving would be performed to obtain the new shape of M by computing the new position of each vertex. To address the potential uncertainties in the raw depth and input surface normal, two simple and effective strategies were proposed to update depth and normal respectively at each step. Please refer to Fig.3.4 for an overview.

3.2 Formulation and Implementation Details

3.2.1 Initialization

Normal Estimation from RGB Image

The surface normal at each facet was required as the input of our iterative depth completion method. There were a variety of deep learning-empowered surface normal estimation methods [59, 60, 61, 62] with a single RGB image as input, some of which leverage the outstanding performance of CNN [63, 64]. We chose a recent work [44] to estimate the surface normal since it achieved state-of-the-art performance on the ScanNet dataset [3] as well as providing the estimation of aleatoric uncertainty in the dataset as shown in Fig. 3.5. Furthermore, the maximum angle between the estimated normal and corresponding ray direction was 80.475 degree after statistics. Therefore, we did not need to consider the specific threshold for estimated normal to make sure there were always intersections between the facet and the rays formed by the optical center and vertices on the physical imaging plane. It proposed the angular von Mises-Fisher distribution to model the

normal probability density function (PDF), and the derived cumulative probability of angular error is utilized in our proposed update strategy. In our work, we utilized the pre-trained model on ScanNet to estimate the surface normal. Also, the estimated normal was used for the other two works for comparison in experiments. The concentration parameter κ predicted along with the surface normal was taken as uncertainty for each facet. Based on each κ, we proposed an updating strategy for the estimated normal, which would be detailed in Section 3.2.4.

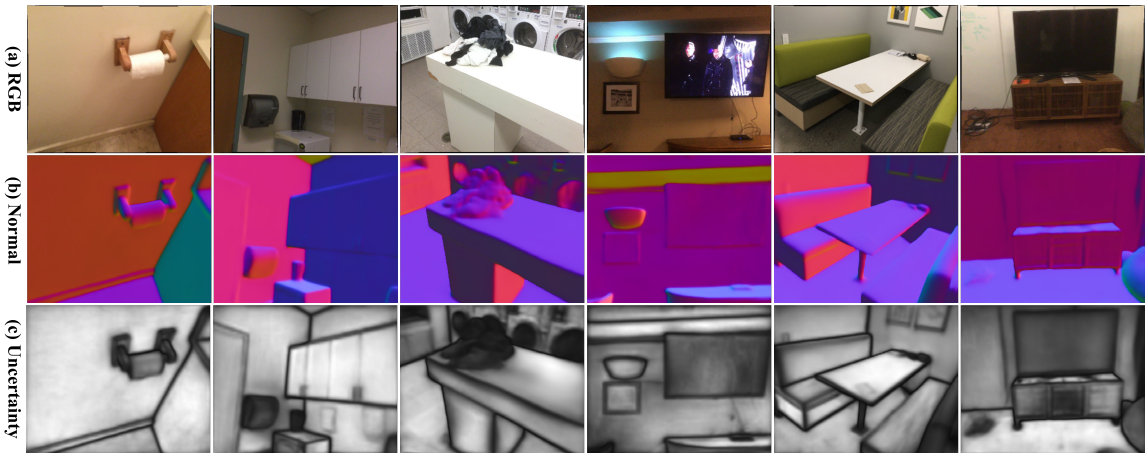


Figure 3.5: Estimated normal maps and uncertainty maps.

Depth-discontinuity Edges Detection and Trimming

In a raw depth image, there were regions with quite distinct depths (e.g., different objects), which indicated the discontinuity of the depth image. Existing methods considered the raw depth image as a whole and therefore cannot handle the discontinuity well (see stretched regions shown in Fig. 1.1). In our geometric processing-based method, we handled the discontinuity through a simple topological process: the boundaries of discontinuous regions, which were edges of mesh surface M, were detected and the neighboring depths were trimmed. In the following computation for surface mesh deformation, the neighboring facets with trimmed edges as boundaries would be decoupled.

Fig. 3.6, which was similar to [45]. Without considering rotation, there were five cases of depth-discontinuity edges (DDEs) in Fig. 3.6, which was similar to [45]. With the recent advancement of transformer-based large models [63, 64], the depth image can be robustly segmented into different regions based on its RGB information, but in this work, the DDEs were manually detected on RGB

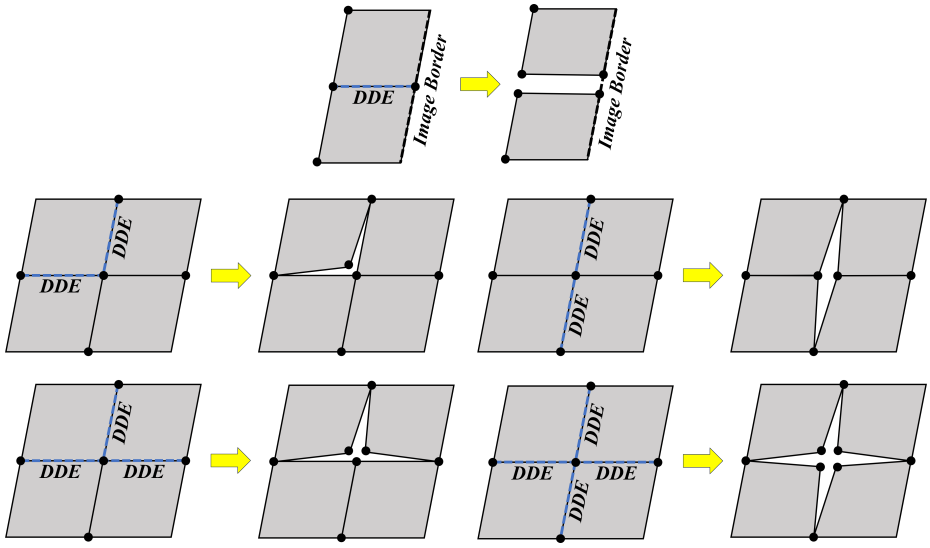


Figure 3.6: Five cases of depth-discontinuity edges (DDEs).

images and annotated by different colors as shown in Fig. 3.7. Theoretically, up to four different colors can represent all DDEs on an RGB image. Trimming neighboring depths on raw depth maps was along the detected DDEs, and its main idea was that the depth values that were between annotated DDEs or between annotated DDEs and invalid depths were deleted. Trimmed depth images were shown in Fig. 3.7.

Figure 3.7: Detected DDEs and the trimmed raw depth maps.

Initial Depth Generation Through Laplacian Diffusion

To start our iterative depth completion, the initial depth values of the depth-missing regions were required to assist shape constraints. We simply applied the Laplacian mesh diffusion method, which has been used in mesh hole filling [65], to generate initial depth values of depth missing regions as shown in Fig. 3.8. For the mesh surface with trimming applied, we applied a global

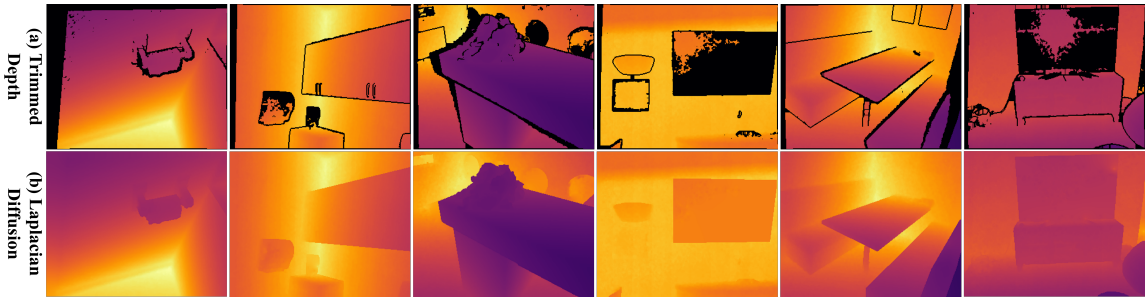


Figure 3.8: Some examples of global Laplacian diffusion.

Laplacian diffusion by solving a linear equation system assembled according to the connectivity of the mesh surface and the known depth values as the boundary conditions (See Fig. 3.9). In spite of more sophisticated methods that could be applied and tested in the future, in our implementation, we found that order-one Laplacian diffusion led to good results.

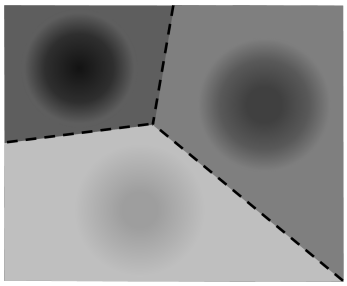


Figure 3.9: An illustration for global Laplacian diffusion with boundary conditions. The dotted lines represent boundary conditions.

3.2.2 Locally Shaping Each Facet

For each facet $f_{i,j}$, the vertices of a quadrangular facet $f_{i,j}$ were projected onto the plane with the normal $\mathbf{n}_{i,j}$, where the plane was supposed to be passing through the *current* projected center, $\mathbf{c}_{i,j}$, of $f_{i,j}$. The projection of a vertex $\mathbf{v}_{k,l}$ ($k \in \{i, i+1\}$ and $l \in \{j, j+1\}$) was along the vector

(i.e. $\widetilde{\mathbf{ov}}^0_{k,l}$ formed by camera center $\mathbf{o}$ and the initial position of the vertex $\mathbf{v}^0_{k,l}$ on the image plane and normalized), which is obtained as

$$p_{i,j}(\mathbf{v}_{k,l}) = \mathbf{c}_{i,j} + (\frac{(\mathbf{c}_{i,j} - \mathbf{o}) \cdot \mathbf{n}_{i,j}}{\widetilde{\mathbf{ov}}^0_{k,l} \cdot \mathbf{n}_{i,j}})\widetilde{\mathbf{ov}}^0_{k,l} \tag{3}$$

A vertex $\mathbf{v}_{k,l}$ of M surrounded by four facets would have four projected positions computed by Eq.(3). Simply assigning $\mathbf{v}_{k,l}$ to the average position of these four points did not lead to a good result. A more sophisticated global-solving method was developed and presented below.

3.2.3 Globally Solving Depth Value

In the global step, following the previous work [35], the vertices of $f_{i,j}$ were desired to move to their projections obtained in the local step, represented by a column vector as:

$$\mathbf{p}(f_{i,j}) = \begin{bmatrix} p_{i,j}(\mathbf{v}_{i,j}) & p_{i,j}(\mathbf{v}_{i+1,j}) & p_{i,j}(\mathbf{v}_{i+1,j+1}) & p_{i,j}(\mathbf{v}_{i,j+1}) \end{bmatrix}^T \tag{4}$$

The unknown depth values of $f_{i,j}$'s four vertices can be represented as

$$\mathbf{z}(f_{i,j}) = \begin{bmatrix} d_{i,j} & d_{i+1,j} & d_{i+1,j+1} & d_{i,j+1} \end{bmatrix}^T \tag{5}$$

The deformation of M was desirable to move the position of each vertex to its projected position in each facet, which can be formulated to the following optimization problem:

$$\Phi_n(\{d_{k,l}\}) = \sum_{f_{i,j}} \|\mathbf{z}(f_{i,j}) - \mathbf{p}(f_{i,j})\|^2 \tag{6}$$

Similarly, to release the degree of freedom of translation in the [36], we subtracted the mean position of the four vertices for both $\mathbf{z}(f_{i,j})$ and $\mathbf{p}_{i,j}$ [35], and therefore, the formulation became

$$\Phi_n(\{d_{k,l}\}) = \sum_{f_{i,j}} \|\mathbf{N}\mathbf{z}(f_{i,j}) - \mathbf{N}\mathbf{p}(f_{i,j})\|^2 \tag{7}$$

with $\mathbf{N} = \mathbf{I}_{4\times4} - \frac{1}{4}\mathbf{1}$, and $\mathbf{1}$ was a 4×4 matrix with all elements equal to 1. Minimizing Eq.7 solely could lead to ambiguity since there were multiple solutions for the facets fulfilling normal constraints under the perspective projection. Therefore, the observed depth information from the raw depth image was utilized as positional constraints to eliminate scale ambiguity. These positional constraints can be formulated as the following optimization problem:

$$\Phi_o(\{d_{k,l}\}) = \sum_{f_{i,j}\in F_o} \|\mathbf{z}(f_{i,j}) - \mathbf{p}(f_{i,j})\|^2 \tag{8}$$

where F_o was the set of facets with observed depth values from the raw depth image. The overall objective function Φ can be obtained simply through the summation of Φ_n and Φ_o (i.e., $\Phi = \Phi_n + \Phi_o$).

Without loss of generality, the mesh surface M was assumed to have $\mathbb{N}$ vertices, $\mathbb{M}$ quadrangular facets, and $\mathbb{M}_o$ facets with depth values observed. Minimizing Φ was straightforward by solving a linear system of $\mathbb{N}$ equations: $\partial\Phi/\partial d_{k,l} = 0$. Since both Φ_n and Φ_o were in quadratic forms, a simpler formulation with a more efficient numerical scheme was developed below.

$$\Phi(\{d_{k,l}\}) = \Phi_n(\{d_{k,l}\}) + \Phi_o(\{d_{k,l}\}) = \|\mathbf{A}_n\mathbf{x} - \mathbf{b}_n\|^2 + \|\mathbf{A}_o\mathbf{x} - \mathbf{b}_o\|^2 \tag{9}$$

where $\mathbf{A}_n$ was a $4\mathbb{M} \times \mathbb{N}$ matrix derived from $\mathbf{N}\mathbf{z}(f_{i,j})$, $\mathbf{b}_n$ was a vector with $4\mathbb{M}$ components derived from $\mathbf{N}\mathbf{p}(f_{i,j})$, $\mathbf{A}_o$ was a $4\mathbb{M}_o \times \mathbb{N}$ matrix from $\mathbf{z}(f_{i,j})$ of those facets in the set F_o, $\mathbf{b}_o$ was a vector with $4\mathbb{M}_o$ depth values obtained from the local projection $\mathbf{p}(f_{i,j})$, and the vector $\mathbf{x}$ contained all the unknown depth values at the vertices of M. Finally, $\Phi(\{d_{k,l}\})$ can be rewritten as $\Phi(\{d_{k,l}\}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$, with $\mathbf{A} = \begin{bmatrix}\mathbf{A}_n \\ \mathbf{A}_o\end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix}\mathbf{b}_n \\ \mathbf{b}_o\end{bmatrix}$. This was a standard least-square formulation, which can be solved by finding out the solution of $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$, where $\mathbf{A}$ was a $4(\mathbb{M} + \mathbb{M}_o) \times \mathbb{N}$ matrix, and $\mathbf{b}$ was a vector with $4(\mathbb{M} + \mathbb{M}_o)$ values.

3.2.4 Uncertain-aware Updating Stragedies

In each iteration, the solved new depth values led to new positions of M's vertices. With these new positions of vertices, the normal vectors associated with each facet would be updated. For

a facet, its four vertices after deformation were generally not on the same plane. Therefore, a plane was fitted based on the computed new positions of the four vertices using a Singular Value Decomposition (SVD) to solve a least-squares problem. For details about the plane fitting from four vertices, please refer to the state-of-the-art paper [66]. The depth value of each facet $f_{i,j}$ can be computed by finding the intersection of the ray $\mathbf{oc}^0_{i,j}$ (formed by the camera center $\mathbf{o}$ and the center point $\mathbf{c}^0_{i,j}$ of $f_{i,j}$ on the physical imaging plane) and the newly fitted plane.

Since the local shaping step relied on the normal and position constraints, an essential question to ask was: should we update these constraints with the newly computed normal vectors and depth values? The initial normal constraints were estimated from RGB images [44], and positional constraints were from the observed depth values. Nevertheless, due to the limitations of their methods, the noise in the training dataset, and the instability of sensor performance, there were potential uncertainties in the initial normal and positional inputs. These uncertainties could result in increasingly larger errors in the depth of information throughout the iterations. Therefore, we proposed two simple and effective strategies to constrain the trusted update ranges for normal vectors and depth values based on the uncertainty modelings.

Normal Updating Strategy

The method we adopted to generate the initial normal input [44], provided the modeling of the noise distribution existing in the ScanNet dataset it was trained on. Specifically, an angular von Mises-Fisher distribution described the normal Probability Density Function (PDF) for each pixel with pixel coordinates i, j:

$$\mathfrak{p}_{i,j}(\mathbf{n}_{i,j}|\boldsymbol{\mu}_{i,j}, \kappa_{i,j}) = \frac{(\kappa^2_{i,j} + 1)\exp(-\kappa_{i,j}\arccos(\boldsymbol{\mu}^T_{i,j}\mathbf{n}_{i,j}))}{2\pi(1 + \exp(-\kappa_{i,j}\pi))}, \tag{10}$$

where $\boldsymbol{\mu}_{i,j}$ was the estimated mean normal, $\kappa_{i,j}$ was the concentration parameter estimated along with the estimated normal, and $\mathbf{n}_{i,j}$ was the ground truth normal. The larger κ_i was, the more concentrated the PDF of $\mathbf{n}_{i,j}$ was around $\boldsymbol{\mu}_{i,j}$. Accordingly, the cumulative probability, $\mathfrak{P}$, of

angular error between $\mathbf{n}_{i,j}$ and $\boldsymbol{\mu}_{i,j}$ was derived as:

$$\begin{aligned}\mathfrak{P}(\arccos(\boldsymbol{\mu}_{i,j}^T\mathbf{n}_{i,j}) \leq \theta_{i,j}|\boldsymbol{\mu}_{i,j}, \kappa_{i,j}) &= \int_0^{2\pi}\int_0^{\theta_{i,j}} \mathfrak{p}_{i,j}(\phi)\sin\phi d\phi d\omega \\ &= \frac{1-\exp(-\kappa_{i,j}\theta_{i,j})(\cos\theta_{i,j}+\kappa_{i,j}\sin\theta_{i,j})}{1+\exp(-\kappa_{i.j}\pi)},\end{aligned} \tag{11}$$

where $\theta_{i,j}$ was angular uncertainty between estimated normal $\boldsymbol{\mu}_{i,j}$ and ground truth $\mathbf{n}_{i,j}$. With an angle $\theta_{i,j}$, there was a probability of $\mathfrak{P}$ that the angle between $\boldsymbol{\mu}_{i,j}$ and $\mathbf{n}_{i,j}$ fell into the range $[0, \theta_{i,j}]$. In principle, a larger $\mathfrak{P}$ led to a larger $\theta_{i,j}$, and vice versa.

We utilized Eq. 11 to constrain the normal updating. By setting up a $\mathfrak{P}$, an range $[0, \theta_{i,j}]$ was obtained. In each iteration, we calculated $\theta_{i,j}$ for each facet $f_{i,j}$, and compared it with the angle $\theta_{i,j}^{'}$ between the newly calculated normal vector and the initial normal estimated from RGB images. If the $\theta_{i,j}^{'}$ was within $[0, \theta_{i,j}]$, the newly calculated normal vector would be the new target normal as the constraint in the next iteration. Otherwise, we rotated the initial normal towards the newly generated normal by the angle $\theta_{i,j}^{'}$, and updated the target normal for the next iteration with the rotated normal.

Note that in principle, $\mathfrak{P}$ for each facet could be different, but we decided to choose the same value for all facets. In the next section, we would present the experiment about how we decide the value of $\mathfrak{P}$. In the future, more sophisticated methods to determine $\mathfrak{P}$ could be explored.

Depth Updating Strategy

The uncertainty of depth came mainly from the depth sensor's measurement precision and working range. Since we tested our method on the ScanNet dataset, the camera information of the Structure Sensor used for ScanNet was of interest. Based on Structure Sensor's fact sheet [67], the precision was 1% of the measured depth. Therefore, we can come up with a range of $[0.99\hat{d}_i^c, 1.01\hat{d}_i^c]$, where $\hat{d}_i^c$ was the measured depth. For the facet with the observed depth, in each iteration, if the newly computed depth fell into the range, we updated the depth value with the new depth value. Otherwise, we need to project the new depth value to the range and updated the depth value with either $0.99\hat{d}_i^c$ or $1.01\hat{d}_i^c$. For facets without the observed depth, we updated the depth value to the newly computed one freely, unless it is beyond the minimum and maximum depth. To be more

specific, they were set to 0.1 meters and 10.0 meters respectively based on the prior knowledge of the selected examples.

Chapter 4

Results

4.1 Dataset and Benchmark

In this work, we evaluated the proposed method on 30 selected examples with distinct depth values across different objects from ScanNet dataset. ScanNet provided raw RGB-D images, camera intrinsics and extrinsic, and high-quality reconstructed mesh. For comparison experiments with Huang et al.'s method [17] and Senushkin et al.'s method [14], the resolution of RGB-D images and rendered depth was resized to 320×256, although our proposed method was theoretically not affected by the resolution. Why were these two work chosen for comparison? At first, they were designed for indoor scenes and formulated the depth completion tasks at the individual frame level without considering temporary information [68]. Secondly, Huang et al.'s method [17] had the similar idea to our work to utilize estimated surface normal and raw depth, while Senushkin et al.'s method [14] was relatively new work and had competitive generalization capability on ScanNet. Finally, we selected some examples from ScanNet rather than the whole dataset for testing and the source code of these two work were available and easily to be implemented without modifying. To make the comparison as fair as possible, we use the same normal map predicted from the trained model of [44] as input for our work and the other two methods.

4.2 Evaluation Metrics

For evaluating the depth completion results, there were five metrics used to compare the completion results with the rendered depth map. It should be noted that the rendered depth map still has pixels without valid depth. Therefore, these pixels were discarded from both of completion results and the rendered depth map during evaluation, and $pix \in val$ was denoted as a pixel with valid depth on the rendered depth map. Given rendered depth D^* and completion result D, the first metric was Root Mean Square Error (RMSE):

$$\sqrt{\frac{1}{|val|}\sum_{pix \in val} \|D(pix) - D^*(pix)\|^2} \tag{12}$$

The second metric was Mean Absolute Error (MAE):

$$\frac{1}{|val|}\sum_{pix \in val} \|D(pix) - D^*(pix)\| \tag{13}$$

The third metric was Absolute Relative Error (Abs. Rel):

$$\frac{1}{|val|}\sum_{pix \in val} \frac{|D(pix) - D^*(pix)|}{|D^*(pix)|} \tag{14}$$

The forth metric was the percentage of pixels δ_i, within the relative error range $e_r \in \{1.05, 1.10, 1.25, 1.25^2, 1.25^3\}$ and the counted pixels should satisfy the following inequality:

$$max(\frac{D^*(pix)}{D(pix)}, \frac{D(pix)}{D^*(pix)}) < e_r, pix \in val \tag{15}$$

The fifth metric was the Structural Similarity Index Measure (SSIM) [69]. It was useful to measure the structural similarity between two corresponding local areas centered in x_i and y_i on two images, X and Y, respectively, and mean SSIM (MSSIM) was used to evaluate the overall image quality.

$$MSSIM(X, Y) = \frac{1}{\mathfrak{N}}\sum_{i=0}^{\mathfrak{N}-1} SSIM(x_i, y_i) \tag{16}$$

where $\mathfrak{N}$ was the number of pixels on the image. Before evaluating depth completion with SSIM, the invalid values of the rendered depth map were assigned to 0, and the same pixels of completion results were assigned to 0 as well. According to [69], the definition of SSIM(x_i, y_i) was:

$$SSIM(x_i, y_i) = \frac{(2\mu_{x_i}\mu_{y_i} + C_1)(2\sigma_{x_i y_i} + C_2)}{(\mu_{x_i}^2 + \mu_{y_i}^2 + C_1)(\sigma_{x_i}^2 + \sigma_{y_i}^2 + C_2)} \tag{17}$$

where, μ_{x_i} and μ_{y_i} were mean values of the local square window centered in x_i and y_i respectively, and $C_1 = 0.01^2$, $C_1 = 0.03^2$. σ_{x_i} and σ_{y_i} were standard deviations:

$$\begin{cases} \sigma_{x_i} = \sqrt{\dfrac{1}{\mathfrak{M}-1}\displaystyle\sum_{j=0}^{\mathfrak{M}-1}(x_j - \mu_{x_i})^2} \\ \sigma_{y_i} = \sqrt{\dfrac{1}{\mathfrak{M}-1}\displaystyle\sum_{j=0}^{\mathfrak{M}-1}(y_j - \mu_{y_i})^2} \end{cases} \tag{18}$$

where $\mathfrak{M}$ was the number of pixels in the local square window, and x_j and y_j belonged to the corresponding local square windows. In practice, the size of the square local window was 11×11, and a Gaussian filter with a standard deviation of 1.5 samples was applied.

4.3 Implementation Details

The proposed method was implemented totally with Python, and the function, *cholesky*, was imported from *sksparse.cholmod* package for large matrix factorization. The code ran on the Ubuntu 20.04 server equipped with AMD Ryzen Threadripper PRO 3955WX@4.3GHz and 128GB RAM.

The termination condition of the local-global iteration depended on the normal energy difference $\mathfrak{E}_n$ and depth energy difference $\mathfrak{E}_o$, where $\mathfrak{E}_n$ measured the absolute difference of $\|\mathbf{A}_n\mathbf{x} - \mathbf{b}_n\|^2$, and $\mathfrak{E}_o$ measured the absolute difference of $\|\mathbf{A}_o\mathbf{x} - \mathbf{b}_o\|^2$, between the current $(\mathfrak{i}+1)$ and previous $(\mathfrak{i})$ iterations. When both $\mathfrak{E}_d$ and $\mathfrak{E}_n$ were less than or equal to 10^{-4} as shown below, the iteration

terminated.

$$\begin{cases} \mathfrak{E}_n = \left| \left\| \mathbf{A}_n^{i+1}\mathbf{x}^{i+1} - \mathbf{b}_n^{i+1} \right\|^2 - \left\| \mathbf{A}_n^{i}\mathbf{x}^{i} - \mathbf{b}_n^{i} \right\|^2 \right| \leq 10^{-4} \\ \mathfrak{E}_d = \left| \left\| \mathbf{A}_o^{i+1}\mathbf{x}^{i+1} - \mathbf{b}_o^{i+1} \right\|^2 - \left\| \mathbf{A}_o^{i}\mathbf{x}^{i} - \mathbf{b}_o^{i} \right\|^2 \right| \leq 10^{-4} \end{cases} \tag{19}$$

4.4 Performance

Methods	*RMSE*↓	*REL*↓	*MAE*↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25^1}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑	*SSIM*↑
Huang [17]	0.1292	0.0183	0.0372	0.9298	0.9625	0.9853	**0.9967**	**0.9989**	0.9401
	(0.3482)	(0.1053)	(0.2097)	(0.4285)	(0.6858)	(0.8825)	(0.9822)	(**0.9941**)	(0.9941)
Senushkin [14]	0.1289	0.0181	0.0339	0.9529	0.9666	0.9798	0.9882	0.9960	0.9457
	(0.3581)	(0.1129)	(0.1888)	(0.6127)	(0.7054)	(0.8159)	(0.8834)	(0.9589)	(0.9905)
Ours	**0.1233**	**0.0106**	**0.0227**	**0.9797**	**0.9859**	**0.9921**	0.9963	0.9984	**0.9573**
	(**0.2687**)	(**0.0338**)	(**0.0707**)	(**0.9164**)	(**0.9350**)	(**0.9660**)	(**0.9837**)	(0.9916)	(**0.9965**)

Table 4.1: Our method, Huang et al.'s [17] and Senushkin et al's [14] were evaluated over all pixels, and only the ones without observed depth values (values in the parenthesis), and the bold font indicated the best one in a whole column.

We tested our method over the 30 selected depth images and measured the metric values for 1) all pixels and 2) only the ones without observed depth values (see the values in parenthesis in Table. 4.1). The same set of tests was applied to Huang et al.'s [17] and Senushkin et al.'s [14] as well, and the results are shown in Table. 4.1. From the results shown in Table. 4.1, our method outperformed two other methods on most metrics. If we only considered those pixels without observed depth, our method was even better. It was reasonable to pay more attention to the errors on pixels without observed depth values instead of all pixels since pixels without observed depth values were the target of the depth completion. Moreover, the average error over all pixels would dilute the error, as those pixels with observed depth values had very low errors in general. Therefore, some results from Huang et al. and Senushkin et al. showed low quality when visualized on the point cloud, even with reasonable error values (see Figs. 4.1 and 4.2). Considering this, in our other results, we only measured the errors over the pixels without observed depth values. Moreover, the results in Table. 4.1 showed that our method had a particularly better performance compared with the other two over the metric $\delta_{1.05}$, which measured the relative depth accuracy at a near distance.

We also visualized the depth images and the point clouds for a qualitative evaluation in Figs. 4.1

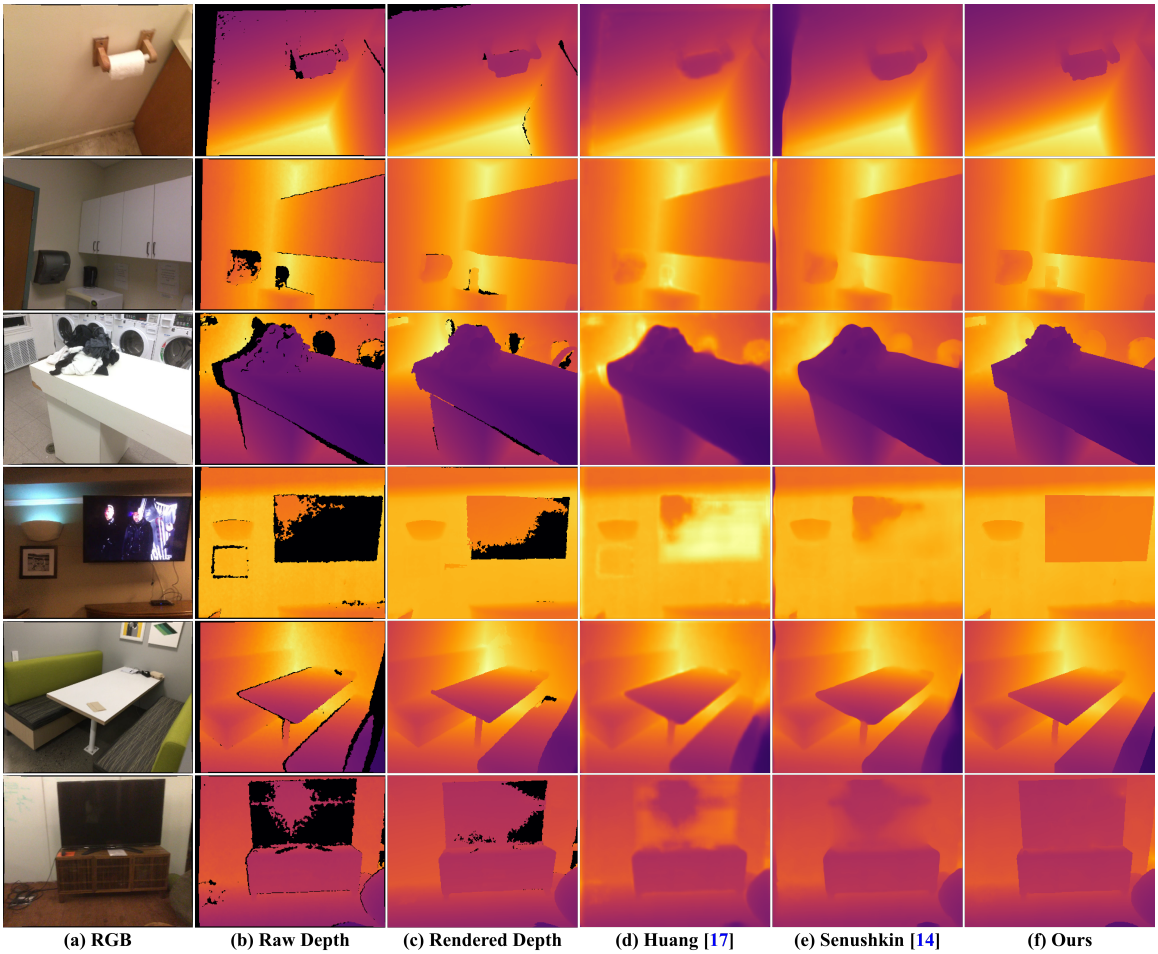


Figure 4.1: Qualitative results of the depth map.

and 4.2. From the depth images, our method successfully generated the missing depth values and preserved the clear, sharp boundaries between different objects. The main reason was that we explicitly consider the discontinuity of depth across different objects and processed it through mesh-based geometric processing with topological modification. Huang et al.'s and Senushkin et al's methods cannot handle the depth discontinuity well and introduced large errors around the boundaries, as shown in the depth images (see Fig. 4.1). The point clouds further indicated the stretched distortions around the boundaries between distinct objects (shown in Fig. 4.2).

For reducing the impact of manually detected depth-discontinuity edges in our work and thus having fairer comparison, Fig. 4.3 showed the depth completion results for some specific areas. The first row and second row were the depth completion results at non-discontinuity areas, while the

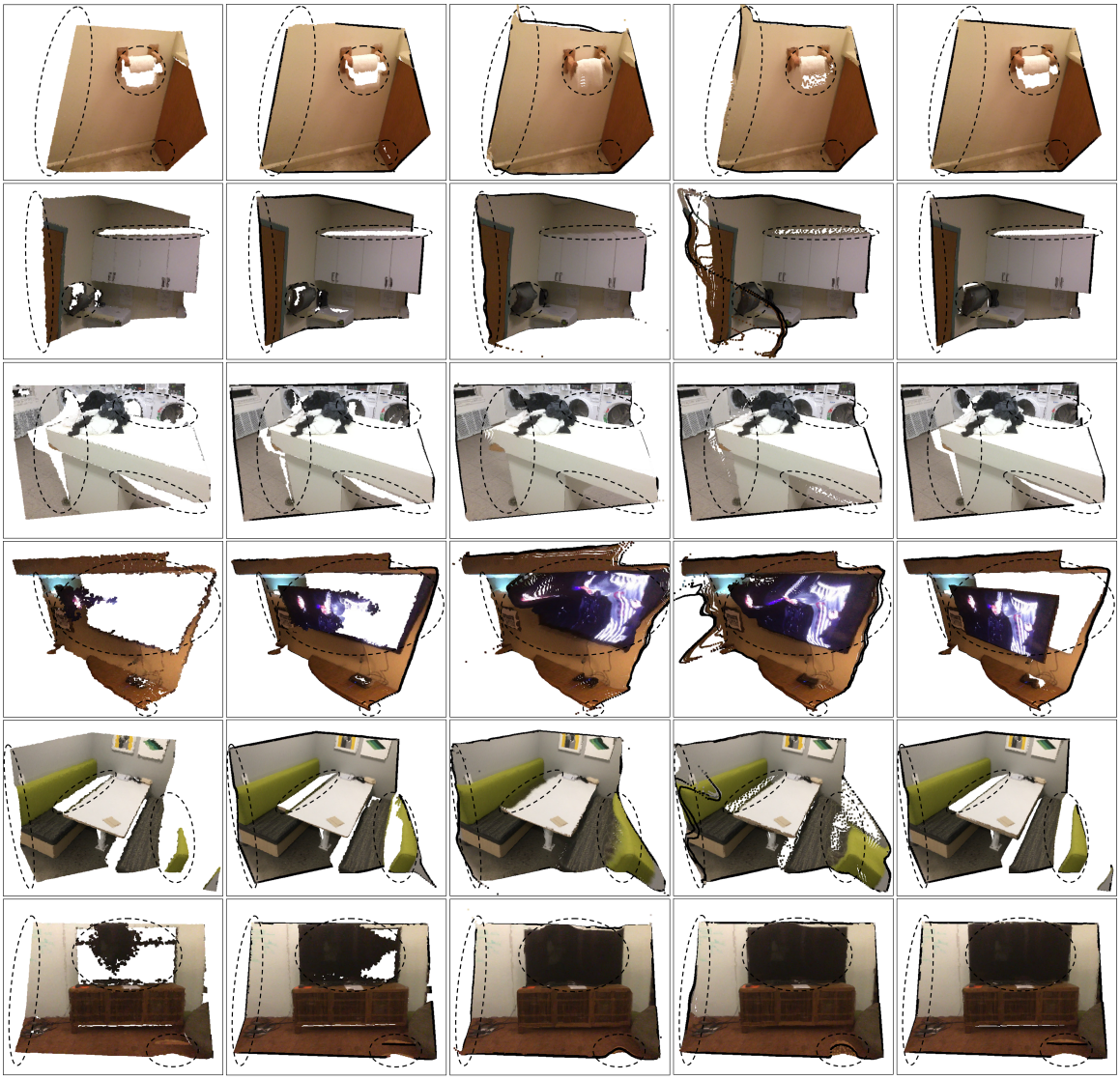


Figure 4.2: Qualitative results of the point cloud.

third and forth rows mainly reflected the noise or even error of rendered depth used as ground truth in other work. The fifth row was to illustrate the mismatching issue existing in rendered depth and other work. In our work, the topological modification based on RGB image and depth trimming were essential to avoid mismatching issue.

To verify our depth and normal updating strategies, we set up an experiment with different conditions. For updating depth, there were two strategies: updating the depth or not. For normal updating, the key was to figure out the angular uncertainty $\theta_{i,j}$, such that an updating range $[0, \theta_{i,j}]$

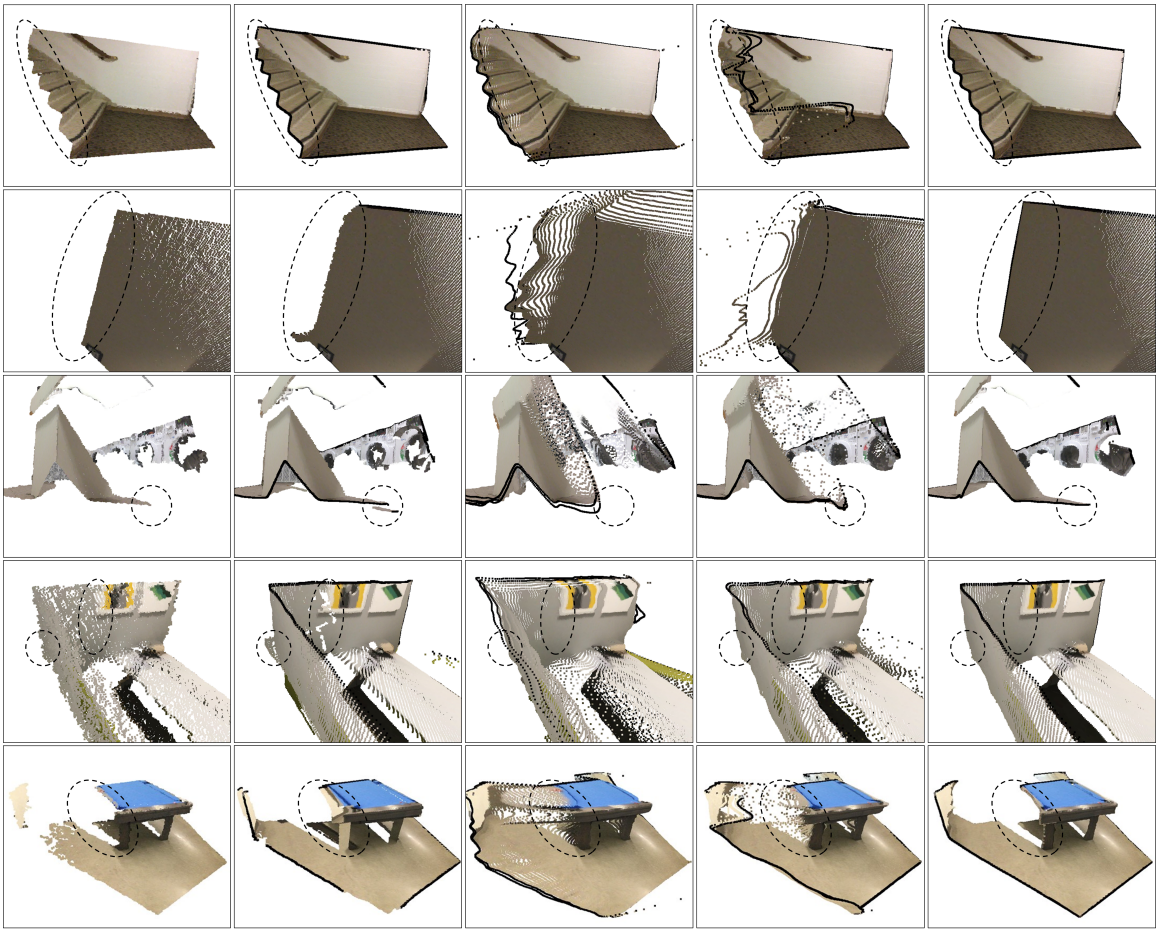


Figure 4.3: Depth completion results for some specific areas.

was obtained. $\theta_{i,j}$ can be calculated using Eq. 11 by setting a cumulative probability P. Therefore, what was the value of P was the essential question to answer. We uniformly sampled the value range of P (i.e., $[0, 1]$) by 0.25 and calculated the updating range with these values (i.e., 0, 0.25, 0.5, and 0.75). By combining different conditions of depth updating and normal updating, we had a total of eight cases. Table. 4.2 included all the results in different cases. The results indicated that the combination of updating depth and $P = 0.5$ led to the smallest error, and therefore, this setup was adopted in all our experiments.

One assumption of our method was that the depth discontinuity processed through a topological

Depth	P	*RMSE*↓	*REL*↓	*MAE*↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25^1}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑	*SSIM*↑
×	0.75	0.2690	0.0344	0.0719	0.9133	0.9332	**0.9660**	**0.9837**	**0.9916**	0.9964
✓	0.75	0.2689	0.0341	0.0713	0.9137	0.9333	0.9659	**0.9837**	**0.9916**	0.9964
×	0.5	0.2689	0.0344	0.0717	0.9132	0.9336	**0.9660**	**0.9837**	**0.9916**	0.9964
✓	0.5	**0.2687**	**0.0338**	**0.0707**	**0.9164**	**0.9350**	**0.9660**	**0.9837**	**0.9916**	**0.9965**
×	0.25	0.2688	0.0343	0.0717	0.9130	0.9333	**0.9660**	**0.9837**	**0.9916**	0.9964
✓	0.25	0.2689	0.0339	0.0709	0.9139	0.9344	**0.9660**	**0.9837**	**0.9916**	**0.9965**
×	0	0.2688	0.0344	0.0718	0.9115	0.9324	0.9657	0.9836	**0.9916**	0.9964
✓	0	0.2691	0.0341	0.0711	0.9120	0.9328	0.9658	**0.9837**	**0.9916**	0.9964

Table 4.2: The quantitative evaluation of different conditions for normal and depth updating strategies. Bold font indicated the best for the whole column.

modification and taken into account in the following optimization would reduce the errors, especially around the boundaries across different objects with very distinct depths. Therefore, we conducted an experiment to validate this assumption. We applied our method to the same set of depth images twice, with all other setups the same, but one with depth discontinuity processed and considered and one not. The results shown in Fig. 4.4 suggested that without discontinuity processed and considered, the regions around boundaries had large errors and appear stretched distortions, which validated our assumption.

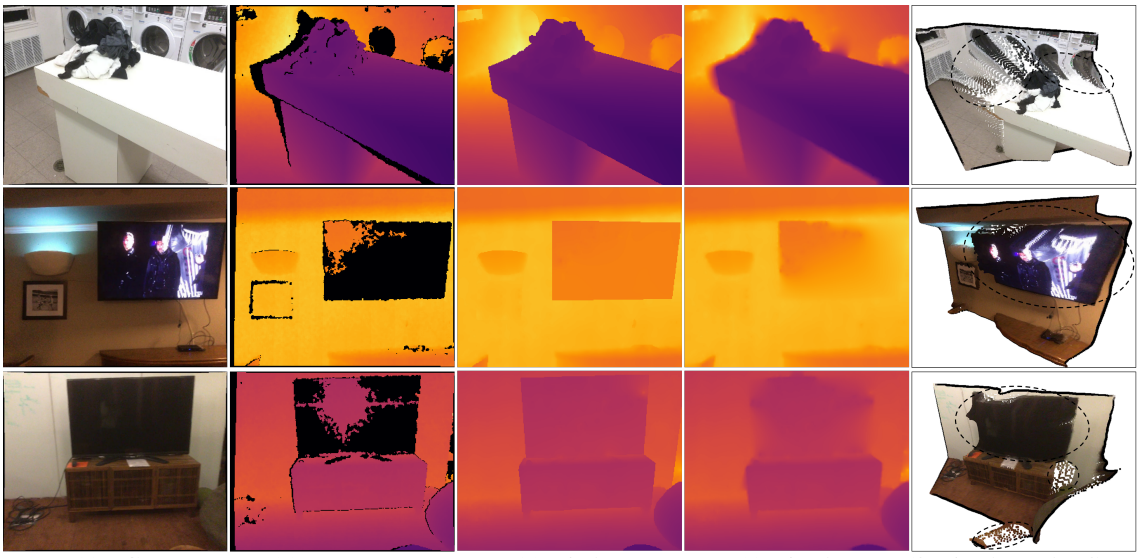
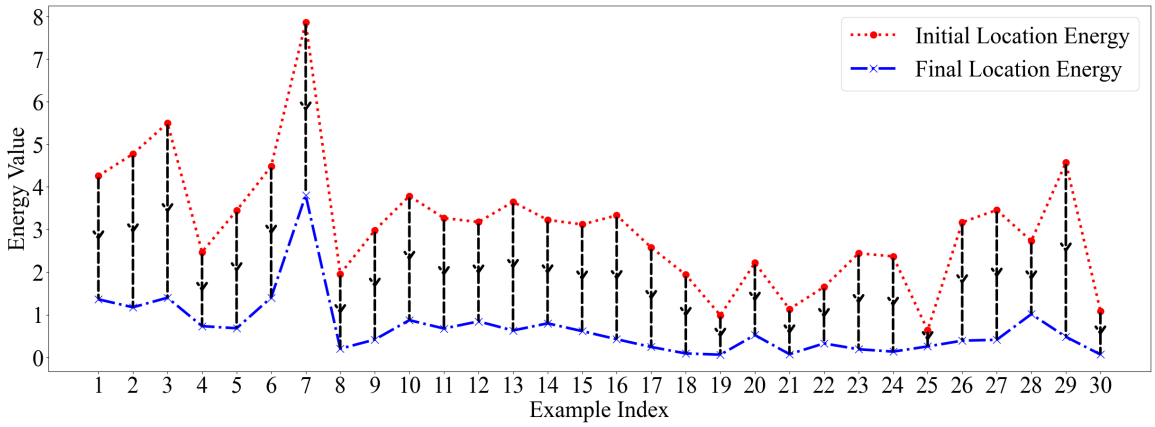


Figure 4.4: Qualitative validation results for discontinuity.

Besides, both the position and normal energies of the first and last iterations for all 30 examples were plotted (see Fig. 4.5). For all 30 tested examples, both position and normal energies had

a reduction in the last iteration compared with the ones in the first iteration, which indicated the robustness of our local-global optimization scheme.

(a) Normal Energy Changes



(b) Location Energy Changes

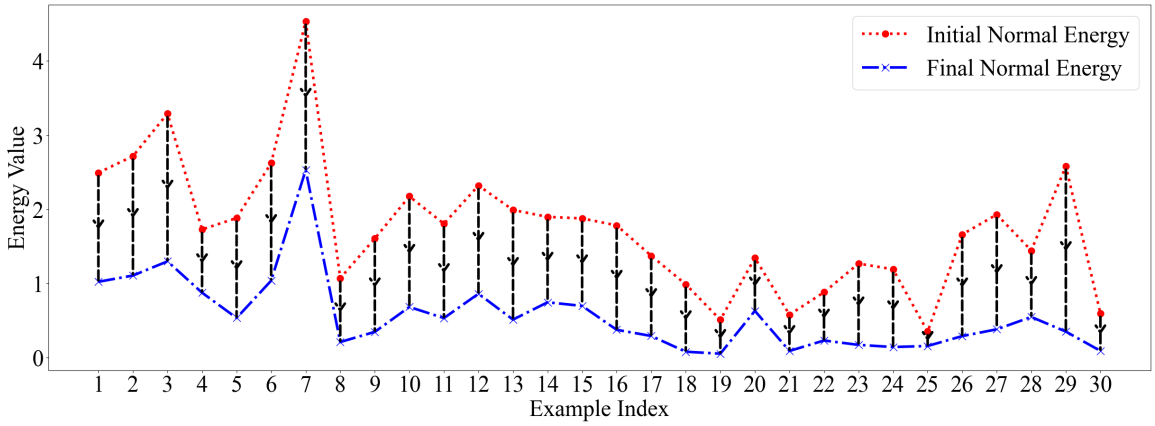


Figure 4.5: Energy Changes.

Although the proposed method paid more attention on depth completion accuracy, time consumption was an important metric to evaluate the proposed method as well and was recorded in Table. 4.3. "Initialization" contained all steps that only need to be calculated once, such as generating the variables for depth discontinuity, processing raw depth map, checking the direction of estimated normal and calculating prefactor matrix by *cholesky*, but the most of time was spent on calculating angular error used in updating normal for each pixel because Eq. 11 was solved by numerical method.

"Local Step", "Global Step" and "Updating Step" referred to the average time spent in an iteration for each example. In each iteration, the reasons why "Updating Step" needs more time were using SVD to fit the new plane for all facets and performing direction checking on target normals for next iteration, while "Local Step" and "Global Step" took much less time to generate projection vector $\mathbf{b}$ and solved the matrix equation.

Currently, the proposed method was still far from achieving real-time performance, and parallel computing would be helpful to further reduce time consumption.

Example	Initialization	Local Step	Global Step	Updating Step	Iterations	Total
01	110.02s	6.29s	0.47s	19.95s	065	1854.50s
02	109.90s	6.17s	0.56s	20.77s	102	2924.77s
03	111.23s	6.15s	0.61s	18.30s	051	1395.68s
04	110.55s	6.16s	0.68s	19.72s	045	1311.96s
05	109.13s	6.29s	0.59s	20.11s	068	1952.35s
06	108.94s	6.21s	0.55s	18.43s	060	1629.00s
07	108.26s	6.26s	0.59s	19.84s	083	2334.09s
08	110.98s	6.21s	0.47s	19.85s	085	2374.95s
09	111.59s	6.13s	0.50s	17.93s	031	0878.38s
10	109.72s	6.23s	0.25s	21.02s	097	2787.33s
11	107.78s	6.27s	0.56s	20.86s	087	2526.62s
12	109.82s	6.28s	0.51s	19.70s	055	1575.81s
13	106.40s	6.31s	0.40s	18.45s	060	1625.60s
14	108.60s	6.28s	0.50s	20.79s	099	2845.83s
15	107.45s	6.62s	0.47s	21.70s	042	1328.88s
16	107.27s	6.76s	0.52s	20.04s	056	1653.43s
17	109.57s	6.74s	0.66s	20.36s	051	1541.54s
18	110.01s	6.70s	0.43s	18.17s	047	1312.75s
19	110.05s	6.74s	0.28s	20.50s	060	1780.78s
20	111.55s	6.68s	0.47s	21.39s	041	1292.32s
21	109.60s	6.78s	0.18s	21.50s	035	1114.92s
22	112.47s	6.81s	0.35s	20.81s	076	2260.77s
23	110.10s	6.82s	0.58s	22.15s	073	2286.20s
24	111.62s	6.77s	0.50s	20.30s	074	2169.17s
25	109.65s	6.75s	0.28s	20.22s	071	2063.86s
26	109.33s	6.69s	0.30s	20.17s	067	1945.26s
27	110.07s	6.78s	0.49s	21.84s	087	2664.99s
28	111.51s	6.74s	0.37s	21.74s	059	1827.41s
29	110.47s	6.69s	0.34s	20.24s	060	1760.52s
30	118.16s	6.67s	0.44s	20.03s	088	2517.26s
Avg.	109.85s	6.50s	0.46s	20.23s	066	1779.79s

Table 4.3: Time consumption for selected examples. The index of examples was same as in Fig. 4.5, and the units of time consumption were seconds(s). The last line recorded the average time spent on each example.

Chapter 5

Conclusion

We presented a depth image completion method based on discrete geometric processing. Our proposed method first converted the input raw depth image to a quadrilateral mesh surface and then, completed the missing depth information under a mesh deformation framework, with raw depth information, a generated normal field [3], and the depth discontinuity information as input. This mesh deformation framework processed the depth discontinuity through a topological modification of the mesh surface.

An optimization question was formulated to fulfill the normal constraints, position constraints, and perspective projection constraints from the pin-hole model and solved using a local shaping and global solving strategy. We also considered the uncertainties in the input normal and observed depth information and developed normal and depth updating strategies based on the geometric meaning of uncertainties.

To tested our method, we conducted experiments on 30 examples from the ScanNet dataset, and five different metrics were adopted to measure the results. The experimental results showed that our method outperformed the two existing methods, Huang et al.'s [17] and Senushkin et al.'s [14] both quantitatively and qualitatively. Especially, in the regions around boundaries across different objects, our method appeared to have clear and sharp disconnections due to the explicit processing and consideration of discontinuity, which provided a reliable and reasonable direction for future research. In contrast, Huang et al.'s [17] and Senushkin et al.'s [14] cannot handle the discontinuity well due to their design principles and processing methods for pixels around depth discontinuity.

Thus, their results had large errors around the boundaries between different objects and appeared as stretched regions.

We verified the normal and depth updating strategies and figured out that updating depth in each iteration and setting the cumulative probability of angular error to be P, led to the smallest errors. For facets with observed depth, the updatable depth range was related to the measurement precision, while the minimum depth and maximum depth for facets without raw depth were set to $0.1m$ and $10m$ respectively according to prior knowledge.

This work also indicated the validity of our assumption that processing and considering depth discontinuity would reduce the errors in depth completion through an experiment with two conditions (with and without depth discontinuity). The explicit topological modification was more beneficial than implicit processing by assigning weights to generate clear and sharp boundaries at depth discontinuity.

Besides, we compared the initial normal and position energies in the first iteration and the ones in the last iteration for all 30 examples. The results showed reductions for both energies over all 30 examples, which confirmed the robustness of our method and was essential for optimization problem.

Also, we recorded the time consumption of each component of the proposed method to illustrate that the current method was still far away from real-time performance. The efficiency bottleneck was not solving large sparse matrices but other necessary steps, and parallel computing can further reduce time consumption.

While our proposed method showed promising results, it was still in the preliminary stage. Our current local shaping with normal constraints was still coupled with the observed depth from the raw depth image. In principle, local shaping with normal constraints should be only related to the shape of objects, not the position (the depth in this problem). The scale of the entire reconstruction scene should be independent of the relative position of points within the scene. Moreover, there could be a large error introduced if the observed depth has a large error, which affects the surrounding objects. To address these problems, a new formulation for local shaping with normal constraints, which is position-independent, is planned to be proposed.

Although we proposed strategies to handle the uncertainties in the raw depth and normal field,

our strategies were still simple and did not consider the noise or even errors existing in the uncertainties. Another question that has not yet been answered is: do the uncertainties in normal and depth weigh the same? Because raw depth and normal field belongs to different domains, their forms should be unified into a same framework for balancing their effects. More comprehensive studies and more sophisticated methods are expected in future work.

Besides, discontinuity detection is still a precondition in the current proposed method, which is worthy of further study to achieve automation. Different methods are expected to be explored, especially those with a semantic understanding of the objects or surfaces of the objects. This information has the potential to further improve the depth completion result.

As for time efficiency, parallel computing and reserving memory in advance are from the perspective of code implementation. The acceleration at the algorithm level is required as well, such as using analytical solutions to replace matrix calculations.

At the same time, we found that in datasets such as ScanNet, the rendered depth, which was considered as the ground truth in many research works relying on ScanNet, presented a large error or incompletion (see column (b) of Fig. 4.2). A dataset with a rendered depth of much higher quality would be in demand from a long-term perspective.

Bibliography

[1] Yunbo Zhang and Tsz-Ho Kwok. Design and interaction interface using augmented reality for smart manufacturing. *Procedia Manufacturing*, 26:1278–1286, 2018.

[2] Tsz Ho Kwok and Tom Gaasenbeek. Dynamic computer-aided process control with computer vision for industry 4.0. In *Flexible Automation and Intelligent Manufacturing: The Human-Data-Technology Nexus: Proceedings of FAIM 2022, June 19–23, 2022, Detroit, Michigan, USA, Volume 2*, pages 510–518. Springer, 2023.

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[4] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.

[5] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2):1899–1906, 2020.

[6] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

[7] Ming Wei, Ming Zhu, Yaoyuan Zhang, Jiarong Wang, and Jiaqi Sun. Real-time depth completion based on lidar-stereo for autonomous driving. *Frontiers in Neurorobotics*, 17:1124676, 2023.

[8] Jian Wang and Yize Liang. Generation and detection of structured light: a review. *Frontiers in Physics*, 9:688284, 2021.

[9] Lina Liu, Yiyi Liao, Yue Wang, Andreas Geiger, and Yong Liu. Learning steering kernels for guided depth completion. *IEEE Transactions on Image Processing*, 30:2850–2861, 2021.

[10] Haowen Wang, Mingyuan Wang, Zhengping Che, Zhiyuan Xu, Xiuquan Qiao, Mengshi Qi, Feifei Feng, and Jian Tang. Rgb-depth fusion gan for indoor depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6209–6218, 2022.

[11] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcns. In *European Conference on Computer Vision*, pages 90–107. Springer, 2022.

[12] Rizhao Fan, Zhigen Li, Matteo Poggi, and Stefano Mattoccia. A cascade dense connection fusion network for depth completion. In *The 33rd British Machine Vision Conference*, volume 1, page 2, 2022.

[13] Chuhua Xian, Dongjiu Zhang, Chengkai Dai, and Charlie CL Wang. Fast generation of high-fidelity rgb-d images by deep learning with adaptive convolution. *IEEE Transactions on Automation Science and Engineering*, 18(3):1328–1340, 2020.

[14] Dmitry Senushkin, Mikhail Romanov, Ilia Belikov, Nikolay Patakin, and Anton Konushin. Decoder modulation for indoor depth completion. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2181–2188. IEEE, 2021.

[15] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *European Conference on Computer Vision*, pages 214–230. Springer, 2022.

[16] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018.

[17] Yu-Kai Huang, Tsung-Han Wu, Yueh-Cheng Liu, and Winston H Hsu. Indoor depth completion with boundary consistency and self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[18] Dongran Ren, Meng Yang, Jiangfan Wu, and Nanning Zheng. Surface normal and gaussian weight constraints for indoor depth structure completion. *Pattern Recognition*, 138:109362, 2023.

[19] Maurilio Di Cicco, Luca Iocchi, and Giorgio Grisetti. Non-parametric calibration for depth sensors. *Robotics and Autonomous Systems*, 74:309–317, 2015.

[20] Alex Teichman, Stephen Miller, and Sebastian Thrun. Unsupervised intrinsic calibration of depth sensors via slam. In *Robotics: Science and systems*, volume 248, page 3. Citeseer, 2013.

[21] 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(4):1, 2017.

[22] 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):1–11, 2013.

[23] Zhanghao Sun, Wei Ye, Jinhui Xiong, Gyeongmin Choe, Jialiang Wang, Shuochen Su, and Rakesh Ranjan. Consistent direct time-of-flight video depth super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5075–5085, 2023.

[24] Jiwan Kim, Minchang Kim, Yeong-Gil Shin, and Minyoung Chung. Accurate ground-truth depth image generation via overfit training of point cloud registration using local frame sets. *arXiv preprint arXiv:2207.07016*, 2022.

[25] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 120–136. Springer, 2020.

[26] Xiaojuan Qi, Zhengzhe Liu, Renjie Liao, Philip HS Torr, Raquel Urtasun, and Jiaya Jia. Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):969–984, 2020.

[27] Hongmin Liu, Xincheng Tang, and Shuhan Shen. Depth-map completion for large indoor scene reconstruction. *Pattern Recognition*, 99:107112, 2020.

[28] Girish Hegde, Tushar Pharale, Soumya Jahagirdar, Vaishakh Nargund, Ramesh Ashok Tabib, Uma Mudenagudi, Basavaraja Vandrotti, and Ankit Dhiman. Deepdnet: Deep dense network for depth completion task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2190–2199, 2021.

[29] Yongjin Lee, Seokjun Park, Beomgu Kang, and Hyunwook Park. Confidence guided depth completion network. *arXiv preprint arXiv:2202.03257*, 2022.

[30] Yasuhiro Yao, Menandro Roxas, Ryoichi Ishikawa, Shingo Ando, Jun Shimamura, and Takeshi Oishi. Discontinuous and smooth depth completion with binary anisotropic diffusion tensor. *IEEE Robotics and Automation Letters*, 5(4):5128–5135, 2020.

[31] Chaohui Wang, Huan Fu, Dacheng Tao, and Michael J Black. Occlusion boundary: A formal definition & its detection via deep exploration of context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2641–2656, 2020.

[32] Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[33] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14648–14657, 2020.

[34] Mengyang Pu, Yaping Huang, Qingji Guan, and Haibin Ling. Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6879–6888, 2021.

[35] Wuyuan Xie, Yunbo Zhang, Charlie CL Wang, and Ronald C-K Chung. Surface-from-gradients: An approach based on discrete geometry processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2195–2202, 2014.

[36] Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. Shape-up: Shaping discrete geometry with projections. In *Computer Graphics Forum*, volume 31, pages 1657–1667. Wiley Online Library, 2012.

[37] Kailun Hu, Shuo Jin, and Charlie CL Wang. Support slimming for single material based additive manufacturing. *Computer-Aided Design*, 65:1–10, 2015.

[38] Tsz-Ho Kwok and Yong Chen. Gdfe: geometry-driven finite element for four-dimensional printing. *Journal of Manufacturing Science and Engineering*, 139(11):111006, 2017.

[39] Guoxin Fang, Christopher-Denny Matte, Rob BN Scharff, Tsz-Ho Kwok, and Charlie CL Wang. Kinematics of soft robots by geometric computing. *IEEE Transactions on Robotics*, 36(4):1272–1286, 2020.

[40] Tianyu Zhang, Guoxin Fang, Yuming Huang, Neelotpal Dutta, Sylvain Lefebvre, Zekai Murat Kilic, and Charlie CL Wang. S3-slicer: A general slicing framework for multi-axis 3d printing. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022.

[41] Andrea Conti, Matteo Poggi, Filippo Aleotti, and Stefano Mattoccia. Unsupervised confidence for lidar depth maps and applications. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8352–8359. IEEE, 2022.

[42] Wang Zhao, Shaohui Liu, Yi Wei, Hengkai Guo, and Yong-Jin Liu. A confidence-based iterative solver of depths and surface normals for deep multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6168–6177, 2021.

[43] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th international conference on machine vision applications (MVA)*, pages 1–6. IEEE, 2019.

[44] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13137–13146, 2021.

[45] Wuyuan Xie, Miaohui Wang, Mingqiang Wei, Jianmin Jiang, and Jing Qin. Surface reconstruction from normals: A robust dgp-based discontinuity preservation approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5328–5336, 2019.

[46] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[47] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[48] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[49] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 103–119, 2018.

[50] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10615–10622, 2020.

[51] Yuankai Lin, Hua Yang, Tao Cheng, Wending Zhou, and Zhouping Yin. Dyspn: Learning dynamic affinity for image-guided depth completion. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[52] Ilya Makarov, Vladimir Aliev, and Olga Gerasimova. Semi-dense depth interpolation using deep convolutional neural networks. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1407–1415, 2017.

[53] Yiyi Liao, Lichao Huang, Yue Wang, Sarath Kodagoda, Yinan Yu, and Yong Liu. Parse geometry from a line: Monocular depth estimation with partial laser observation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 5059–5066. IEEE, 2017.

[54] Zhao Chen, Vijay Badrinarayanan, Gilad Drozdov, and Andrew Rabinovich. Estimating depth from rgb and sparse sensing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 167–182, 2018.

[55] Yifan Zuo, Qiang Wu, Ping An, and Jian Zhang. Explicit measurement on depth-color inconsistency for depth completion. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 4037–4041. IEEE, 2016.

[56] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[57] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, 2008.

[58] Y. Liu, H. Pottmann, J. Wallner, Y.L. Yang, and W. Wang. Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics*, 25(3):681–689, 2006.

[59] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 539–547, 2015.

[60] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas J Guibas. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8638–8647, 2019.

[61] Rui Wang, David Geraghty, Kevin Matzen, Richard Szeliski, and Jan-Michael Frahm. Vplnet: Deep single view normal estimation with vanishing points and lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 689–698, 2020.

[62] Quewei Li, Jie Guo, Yang Fei, Qinyu Tang, Wenxiu Sun, Jin Zeng, and Yanwen Guo. Deep surface normal estimation on the 2-sphere with confidence guided semantic attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 734–750. Springer, 2020.

[63] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[64] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*, 2023.

[65] Peter Liepa. Filling holes in meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 200–205, 2003.

[66] V Schomaker, J Waser, RE t Marsh, and G Bergman. To fit a plane or a line to a set of points by least squares. *Acta crystallographica*, 12(8):600–604, 1959.

[67] Washington and Lee University. Structure 3d sensor, 2024. https://my.wlu.edu/iq-center/equipment/prototyping/scanners/structure-sensor [Accessed: January-11, 2024].

[68] Sriram Krishna and Basavaraja Shanthappa Vandrotti. Deepsmooth: Efficient and smooth depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3357–3366, 2023.

[69] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.