

**HUMAN-ALIGNED HIGH-FIDELITY 3D SHAPE EVALUATION AND HAND
RECONSTRUCTION**

by

Tianyu Luan

April 2025

A dissertation submitted to the
Faculty of the Graduate School of
the University at Buffalo, State University of New York
in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy

Department of Computer Science and Engineering

Copyright by
Tianyu Luan
2025

Acknowledgments

First, I want to express my deepest gratitude to my family—my wife, Manqing, and my parents. Pursuing a PhD in a foreign country has been a personal journey, but it also placed considerable challenge on my family. I am truly thankful for their support, which allowed me to follow my dreams. Throughout my PhD studies, their love and encouragement have been my driving force, and I feel incredibly lucky to have such support and blessing.

I would also like to thank my collaborators—Zhongpai, Ziyang, Terrence, Zhong, Yi, Luyuan, Haoxiang, and Xuan. You witnessed my growth from a novice researcher to someone with much more confidence and experience in research today. Without your collaboration and help, I would not have achieved what I have. It has been an honor to work alongside you.

My thanks to my labmates—Sheng, Liangchen, Jialian, Lin, Sudhir, Yuanhao, Nan, Zixin, and Xuelu. You have given me both research support and daily companionship. You’ve become an irreplaceable part of my PhD journey rather than just my PhD research. I will miss our lunch and dinner chats about research and gossip, and I hope we can continue sharing those lunches and dinners and conversations in the future.

I am deeply grateful to my committee members: Prof. Chen Wang, Prof. Kenneth Joseph, and Prof. Ying Wu. Thank you for taking time from your busy schedules to serve on my committee and offering your invaluable advice. Your insights and feedback have given me fresh perspectives and inspiring ideas not only to my thesis, but also to my future research.

Finally, I want to thank my advisor, Prof. Junsong Yuan. I truly believe he is the best advisor in the world. Before working with Prof. Yuan, I thought that if an advisor could provide (a) detailed research guidance, (b) big-picture direction and high-level advice, (c) freedom to explore my own interests, or (d) room to make mistakes along with emotional support, any one of these would make for a very good advisor. However, Prof. Yuan has done all of the above and more, which makes him the best mentor a junior researcher like me could imagine. If I ever have the chance to become a professor or a team leader in the future, I hope I can be someone like him.

Table of Contents

Acknowledgments	ii
List of Tables	viii
List of Figures	x
1 Introduction	1
1 Background	1
2 Problem Formulation	5
2.1 Fidelity	5
2.2 High-fidelity Reconstruction	6
3 Disertation Overview	6
3.1 Fidelity Dataset and Metric	6
3.2 Fully-supervised High-fidelity Hand Reconstruction	7
3.3 Self-supervised High-fidelity Hand Reconstruction	8
2 Human-aligned Fidelity Metric	9
1 Introduction	9
2 Related Works	11
3 Proposed Method	13
3.1 Overview	13
3.2 Mesh spectrum analysis	13
3.3 Spectrum AUC Difference	16
3.4 Human-adjusted Spectrum AUC Difference	18
3.5 Discretization of Spectrum AUC Difference	18
3.6 Discretization of Human-adjusted SAUCD	20
4 Dataset	22
4.1 Dataset Design	22
4.2 Human Scoring Procedure Overview	23
4.3 Swiss System Tournament for Human Scoring	24

4.4	Outlier Detection	24
4.5	Dataset Error Analysis	25
5	Experiments	25
5.1	Dataset	25
5.2	Implementation Details	26
5.3	Evaluation Methods	27
5.4	Human-adjusted SAUCD Training	28
5.5	Quantitive and Qualitative Results	28
6	Conclusions	35
3	Fully-supervised High-fidelity Hand Reconstruction	38
1	Introduction	38
2	Related Work	43
3	Proposed Method	45
3.1	High-fidelity 3D Hand Model	46
3.2	Hierachical Graph Convolution Network	47
3.3	Graph Frequency Decomposition	48
3.4	Image-Graph Ring Frequency Mapping	50
3.5	Frequency Decomposition Loss	54
4	Datasets and Annotation Generation	56
4.1	High-fidelity Hand Dataset	56
4.2	Topology-correct Annotation Generation	57
5	Experiments	61
5.1	Implementation Details	61
5.2	Quantitative Evaluation	62
5.3	Ablation Studies	65
4	Self-supervised High-fidelity Hand Reconstruction	71
1	Introduction	71
2	Related Works	74
3	Fully-supervised High-fidelity Hand Reconstruction	76
3.1	Coarse Hand Reconstruction	77
3.2	Detail Enhancement Network	78
3.3	Differentiable Rendering	79
3.4	Loss Functions	79
3.5	Training and Inference	83
4	Experiments	83

4.1	Training Dataset	83
4.2	<i>HandScan</i> Benchmark	83
4.3	Implementation Details	85
4.4	Quantitive and Qualitative Results	86
5	Conclusion	90
Appendix A	A Counterexample of the Original Cotan Formula not being Positive Semidefinite	92
Appendix B	Proof of Positive-semidefiniteness of Revised Cotan Formula . . .	96
Appendix C	Proof of SAUCD Satisfies Metric Definition in Spectrum Domain .	98
Reference	115

List of Tables

2.1	Correlations between different metrics and human annotation. “SAUCD” is our basic version metric. “Adjusted SAUCD” is the human-adjusted version of our metric. The ranges of all three correlation coefficients are $[-1, 1]$, and the higher the better.	21
2.2	Dataset statistics and error analysis.	22
2.3	Results when building metrics using each frequency band separately. The bottom row is our proposed metric.	30
2.4	Results with different pruning portions. The metric achieves better results with pruning portion to be 0.1% or 1%. We use pruning portion as 0.1% in our design.	31
2.5	Module replacement. We replace each module of our metric with alternative designs to verify the design of each module.	31
2.6	Distortions in our provided <i>Shape Grading</i> dataset.	36
3.1	Joint and mesh errors (Chamfer distance) of topology-correct mesh annotations of 3 resolution levels on InterHand2.6M. For joint error and Chamfer distance, lower is better.	61
3.2	Module effectiveness Results on the InterHand2.6M [91] dataset. Bold number means the best. For MPJPE and Chamfer distance (CD), lower is better. For MSNR, higher is better. The proposed method improves the accuracy of hand surface details compared to previous methods and our conference version (Conf-level 1-3). While our method generates better shape details in a scalable manner, the general accuracy (MPJPE and CD) of overall shape also increases.	62

3.3	Ablation study on the feature skip connection design, new registration strategy, and the effect of loss functions. Bold number means the best. The 2nd-4th lines show the effectiveness of our Image-Graph Ring Frequency Mapping (IGRFM). The 5th line shows the result of using the previous registration from the conference version. The 6th and 7th lines show the effectiveness of our loss functions.	63
3.4	Quantitative results of removing high-frequency features in IGRFM. We remove the high-frequency features and retain 10% and 1% of the lowest-frequency features in IGRFM frequency rings, and show 3 levels of the quantitative results comparison.	63
3.5	Comparison of different IGRFM ring segmentation strategies. From top to bottom: Radius(proposed method): Keeping the radius difference between adjacent rings to be the same. Area: Keeping the same area difference between adjacent rings. Graph frequency: Keeping the radius difference to be the same as the graph frequency Λ in Eq. (3.7). Square root of graph frequency: Keeping the radius difference to be the same as the square root of graph frequency. Random segmentation: Randomly segment the image frequency band. Our results show that our proposed radius segmentation has the best MSNR result.	68
3.6	The mesh sizes and the resources needed for generating different resolution levels of meshes for both our proposed method and conference version. We observe that despite our performance exceeding that of our conference version, the computational costs barely increase.	68
4.1	<i>HandScan</i> and 11k Hands dataset attributes. Compared with 11k Hands covers more subjects, <i>HandScan</i> provides high-resolution 3D scans for 16 subjects with 3D scanned shape ground-truth and MANO registration. . . .	84
4.2	SOTA comparison of our result with previous works. We evaluate the hand’s general shape using Chamfer Distance (CD) and the details fidelity using FSNR [17]. As shown in the table below, our methods outperform SOTA methods, especially in detail measurements. We also report the average inference time. For CD and Inference Time, a lower number means better performance. For FSNR, the higher the number, the better the performance. The inference time is measured on a single NVIDIA A40 GPU.	84
4.3	We do an ablation study on the fast inference part of our method. The result demonstrates the importance of each loss term. Removing any single loss, including perceptual loss, Laplacian loss, silhouette loss, and frequency loss, it degrades our method’s reconstruction performance, underscoring its contribution to the final result.	85

List of Figures

1.1	The motivation of acquiring high-fidelity in 3D virtual world.	2
1.2	An example of how previous spatial domain 3D shape metrics (Chamfer Distance [13] and UHD [14]) deviate from human evaluation. We create Mesh A by adding a small pose error to the ground truth mesh, and by applying a large smoothing kernel to ground truth, we create Mesh B . Contrary to human perception, previous spatial domain metrics evaluate Mesh B better than Mesh A . This indicates that while they are sensitive to general shape differences, they tend to overlook high-frequency details. Note that different metrics use different units of measurement.	4
2.1	Our SAUCD metric is designed as follows: <i>A</i> . We use mesh Fourier Transform to analyze the spectrums of test and ground truth mesh. <i>B</i> . We compare the difference between two spectrum curves by calculating the Area Under the Curve (AUC) difference. <i>C</i> . We further extend our metric by multiplying the AUC difference with a learnable weight to capture human sensitivity in each frequency band.	12
2.2	Variables defined in our discrete Laplace-Beltrami operator design.	14
2.3	Spectrum Area Under the Curve Difference. We design our metric using the AUC difference of the spectrums. The blue curve and red curve are the test and ground truth mesh spectrum, respectively. The purple area in the last graph is the Spectrum AUC Difference. Please find details in Sec. 3.3. . . .	14
2.4	Objects in our provided <i>Shape Grading</i> dataset and what the object numbers correspond to in Tab. 2.1.	23
2.5	Examples of distorted meshes of different distortion levels in our provided <i>Shape Grading</i> dataset.	23

2.6	The panel of our online user study system. The instruction on the left contains simple instructions for the subjects. On the right side of the page, the top two videos are rendered from distorted meshes. The lower video is rendered from groundtruth mesh.	25
2.7	An example of mesh spectrum curve: We do mesh Fourier transform on the “Origin” mesh and show the spectrum in the left graph. The λ -axis is the eigenvalues of the DLBO matrix, the larger the higher frequency. We also show how mesh changes when gradually removing high-frequency information (mesh A to G). The frequency bands of the meshes are shown as the colored arrows in the left graph.	26
2.8	Learned spectrum weights on all 12 folds. The name of colorful thin lines means the test object name of that fold. The bold purple line is the average weights of all folds. We also show some examples of mesh shape information in different frequency bands. Frequency band A is $[0, 0.0075)$, B is $[0, 0.03)$, and C is $[0, 0.05)$	29
2.9	Counterintuitive low-frequencies information if some of the mesh frequencies are negative. We can see if we remove the high-frequency part of the mesh (resulting in “Filtered mesh 1” and “Filtered mesh 2”) using the original Cotan formula, the mesh’s low-frequency parts show artifacts (sharp shapes). The red circles show the artifacts in the left object. The right object shows a case when these artifacts occur much more often. These artifacts do not occur using our revised Cotan formula DLBO.	30
2.10	We Adapt SAUCD into a loss function and use it in monocular-image-based 3D hand reconstruction. From left to right: input images, reconstruction result w/o SAUCD loss, reconstruction result w/ SAUCD loss, and ground truth mesh. We can see that the enhancement of SAUCD loss in mesh details is clearly noticeable.	32
2.11	AUC normalization. We normalize the spectrum of origin mesh with factor $s = 2$. The blue curve is the resulting curve. We transform the blue curve back to $s = 2$ scaled mesh . As we see on the right side, the mesh’s general shape is kept the same, but the scales increased to twice the size of the original mesh.	33
2.12	Pair-wise distortion type comparisons. s is the percentage difference of the inverse-order pairs compared to groundtruth. Blue color means s is larger than 0, which shows that our metric is better than compared metric among the meshes of distortion pair (d_1, d_2) . Red color means s is smaller than 0, which means the compared metric is better.	33
2.13	Network architecture used when adapting SAUCD to training loss.	34

2.14	Failure cases. We show a case in which our metric does not provide accurate evaluations aligned with the human evaluation.	34
2.15	Examples in our dataset and their evaluation results using different metrics. ↓ means lower is better. ↑ means higher is better. For each object, the mesh on the top-left is the groundtruth mesh, and the rest meshes are distorted meshes. The table below the meshes contains the scores they get from different metrics or from our user study. As shown in the figure, our metric aligns better with user study scores and human perception.	37
3.1	An exemplar hand mesh with sufficient details and its graph frequency decomposition. The x-axis shows frequency components from low to high. The y-axis shows the amplitude of each component on a logarithm scale. At the frequency domain, the signal amplitude generally decreases as the frequency increases.	39
3.2	To map the image features to graph features, previous methods use a simple pooling strategy (first row) or a projection-interpolation strategy (second row), but the detailed high-frequency features would be easily damaged by pooling or small projection errors. In this paper, we propose an Image-Graph Ring Frequency Mapping (IGRFM) (third row) to map the image features to the graph features via the frequency domain.	41
3.3	An example of the topology error of a scanned hand mesh. The red part on the top-right is the missing or ambiguous topology. To better train our network, we use a bidirectional registration strategy to generate valid ground-truth meshes.	42
3.4	We design our scalable hand modeling network in a U-net manner. First, we generate a MANO mesh from image features (light gray block). Then, based on the MANO mesh, we use a multilevel GCN to recover 3 levels of personalized meshes (green blocks from shallow to dark). In order to obtain high-frequency hand details, we use Image-Graph Ring Frequency Mapping (IGRFM) skip-connected image features (yellow blocks) from different layers of the backbone network as parts of the GCN input. At inference, our network can stop at any resolution level, but still provides reasonable high-fidelity results at that resolution.	45
3.5	We design a new high-fidelity hand mesh with 12,337 vertices. Our new model inherits the advantages of the parametric hand model and provides reliable 3D shape estimation with fewer flaws when hand poses change. . .	48

3.6	Frequency decomposition of a 3D hand mesh. Cumulative frequency components start from frequency 0. The range shows the frequency band. For example, [0,20] means the signal of the first 21 frequencies (lowest 21) added together. We can see how the mesh shape changes when we gradually add higher frequency signals to the hand mesh. In general, the hand details increase as higher frequency signals are included.	51
3.7	We use Image-Graph Ring Frequency Mapping (IGRFM) to map image features to graph features through frequency. (a) Input image spatial feature map. (b) Image frequency feature after Fast Fourier Transform (FFT). (c) Frequency feature rings. The inner rings are low-frequency features and the outer rings are high-frequency features. (d) Graph frequency features after ring average pooling. (e) Graph spatial features transformed from graph frequency features using Graph Inverse Fourier Transform (Graph IFT). . . .	52
3.8	Example of Topology-correct hand mesh registration. (a) Scanned mesh with topology flaws (red circle). (b) We use an optimization-based coarse pose registration to get the coarse pose. (c) We then use bidirectional vertex picking to get the topology-correct part of mesh vertices. (d) We finally use 3D Poisson editing to inpaint the topology-incorrect part of the mesh vertices.	55
3.9	Evaluations using Euclidean distance and MSNR under different noise amplitudes in every frequency band. Each line of a different color indicates a frequency band. The maximum and minimum frequencies are shown in the legend. On each line, every dot means adding a random amplitude noise to the mesh. The noise amplitude of each dot is evenly distributed over the ranges shown on the x-axis. The result validates that Euclidean distance is more sensitive to error in low-frequency bands, and MSNR is more sensitive to noise in high-frequency bands. Thus, compared to Euclidean distance, MSNR can better measure the errors in high-frequency details.	56
3.10	We show examples of Noisy Meshes. The meshes from left to right are meshes with a noise maximum amplitude of 0.6 and the frequency band changed from [60,119] to [7680,12336]. For easier visualization, we magnify the vertices location changes by a factor of 5.	57
3.11	Visualized comparison with the conference version. We compare our results with those of our conference version. Our results have better high-fidelity details (first row). Moreover, our proposed method can solve some failure cases of the previous conference version (second row).	64

3.12	Visualization results of “w/o frequency decomposition loss” and “w/o per-vertex error loss” in Sec. 5.3. As shown, if we do not use frequency decomposition loss, the mesh result we get tends to be smoother with fewer personalized details. If we do not use the per-vertex error loss, the mesh’s low-frequency information is not well learned. The mesh we generate exhibits overall shape deformation.	65
3.13	Visualized results of removing high-frequency features in IGRFM. (Best viewed at magnification.) We remove the high-frequency feature rings and retain 10% and 1% of the lowest-frequency features in IGRFM frequency rings, and show the highest-resolution visualized results comparison. As shown in the figure, removing the high-frequency feature rings will cause a loss of the shape details.	66
3.14	Comparison of registered mesh annotations. For each case, the left meshes are our conference version results, and the meshes on the right are our proposed registration method results. We can observe some topology flaws in the conference version results (abnormal triangle faces in the red circles), while our proposed registration does not have such flaws.	67
3.15	Qualitative reconstruction results. The columns, from left to right, are input images, our level 1-3 output meshes, MANO mesh, MANO mesh subdivided to 200k vertices (<i>i.e.</i> the same number of vertices as our mesh), and the ground-truth, respectively. We can see that even if we upsample MANO into the same number of vertices as our mesh, it still does not provide personalized details comparable to our results.	69
4.1	(a) Existing high-fidelity 3D hand reconstruction methods typically rely on specialized 3D scan ground-truth data, which require expensive hardware, time-consuming procedures, and controlled environments. (b) Our self-supervised approach reconstructs high-fidelity 3D hands directly from image inputs, leveraging general shape and detail priors without requiring 3D annotations. This method reduces reliance on specialized 3D-scanned data and broadens applicability across diverse subjects.	72
4.2	Overview of our self-supervised pipeline for high-fidelity 3D hand reconstruction. From a single RGB image, we obtain a coarse MANO [146]-based mesh, subdivide it for higher resolution, and refine it with per-vertex displacements predicted by a detail enhancement network. A differentiable renderer projects the refined mesh for image-space supervision using multiple loss terms (e.g., perceptual l_{perc} , silhouette l_{sil} , Laplacian color l_{color} and normal l_{normal} , frequency-based l_{freq}). This framework allows end-to-end training without requiring 3D ground-truth scans.	75

4.3	Example data of 11k Hands (a) and our benchmark <i>HandScan</i> (b). For <i>HandScan</i> , the top row is the input images, and the bottom row is the hand scan data. As shown in the figure, our dataset scanning has good hand shape details for evaluation.	86
4.4	Example of decomposing a hand mesh into different frequency bands. We accumulate frequency components from low to high, resulting in 12 hand meshes (M_1 to M_{12}). The boundary between low and high frequencies is marked at \mathbf{K} , roughly between M_9 and M_{10} . The central figure illustrates the overall frequency decomposition of the hand shape.	87
4.5	Visualization of the normal maps (2nd column from the left), general shape mesh, and 2D keypoints (3rd column from the left) used in our method. In the third column, green dots represent 2D keypoints from an off-the-shelf estimator, while red dots are the projections of 3D hand joints. We can observe the details and general shape alignment of the details provided by the normal map and general shape provided by conventional hand mesh reconstruction.	88
4.6	We visualize our results for both fast inference (a) and direct inference (b) on <i>HandScan</i> . For direct inference, we also visualize our results on 11k hands (bottom row). As shown in the figure, our result has a better detail shape and fidelity than the baseline approach for both fast inference approach and direct inference approach.	89
A.1	A simple mesh example to show that the original Cotan formula does not guarantee to be positive semidefinite.	93

Abstract

Augmented Reality (AR) and Virtual Reality (VR) technologies have rapidly advanced in diverse fields, from education and remote work to retail and entertainment, offering immersive experiences that boost user engagement, collaboration, and overall satisfaction. Central to these immersive experiences is the concept of fidelity, which profoundly influences users’ sense of immersion, presence, co-presence, and emotional responses. Despite growing interest in creating high-fidelity 3D environments, there remains a pressing need for standardized methods to evaluate fidelity in a way that accurately reflects human perception. Traditional geometric metrics, such as Chamfer Distance, often fail to capture nuanced visual differences, especially those tied to local details that humans readily discern, underscoring the importance of robust, human-aligned fidelity measures.

In this dissertation, we address this gap through a comprehensive approach encompassing both data collection and method development. First, we introduce our user study benchmark dataset, *Shape Grading*, which compiles human-assigned quality scores for a broad range of distorted 3D meshes. Spanning twelve ground-truth objects and incorporating seven common distortion types, each with four levels of severity, our dataset contains 1,008 short video renderings, offering a rich resource for understanding how people perceive realism in synthetic 3D content. By analyzing the correlation between these human evaluations and various automated metrics, we present valuable insights into the strengths and weaknesses of current fidelity assessment approaches, thereby clarifying how well they mirror genuine human perception.

Building on these findings, we propose a novel analytic metric, Spectrum Area Under the Curve Difference (SAUCD), to more effectively gauge fidelity. Our method transforms 3D meshes into the spectrum domain via the discrete Laplace-Beltrami operator and Fourier transform, ensuring that both global structure and finer surface details receive appropriate emphasis. We further refine this metric by learning frequency-specific weights to align closely with human judgment, improving correlations between algorithmic outputs and subjective evaluations. This metric therefore provides a practical yet powerful tool for both researchers and practitioners aiming to create high-fidelity 3D objects.

To demonstrate the real-world applicability of these ideas, we present two high-fidelity reconstruction pipelines on human hand reconstruction, one fully supervised and one self-supervised. In the fully supervised setting, our frequency-split network reconstructs hand meshes in a coarse-to-fine manner, preserving both overall shape and intricate detail. In contrast, our self-supervised framework (FlipFlop) tackles the inherent data-collection challenges by requiring only two RGB images (front and back views) for training, yet still recovers textured, high-frequency hand geometry. Both methods demonstrate the advantages of explicitly modeling high-frequency information, markedly improving realism in 3D hand reconstruction tasks.

Overall, this dissertation lays a foundational framework for understanding, measuring, and generating high-fidelity 3D worlds. By combining large-scale human perception studies with a spectrum-domain metric and specialized reconstruction techniques, we offer a holistic approach that addresses the core challenge of realism in AR/VR. In doing so, we pave the way for more immersive virtual experiences, advancing education, remote collaboration, interactive entertainment, and beyond.

Chapter 1

Introduction

1 Background

Augmented Reality (AR) and Virtual Reality (VR) technologies have shown their vast potential across various domains. In education, VR enables immersive learning environments that enhance students' comprehension and memory through virtual field trips and scientific experiments. For remote work, VR is used to replicate office settings, allowing team members to engage in face-to-face meetings and collaborative efforts in virtual spaces, thus boosting communication efficiency and team bonds. In retail, AR allows consumers to try on clothing or visualize how furniture will look in their homes before purchasing, using smartphones or specialized glasses. Moreover, VR is employed in the real estate sector to offer virtual tours of properties, enabling potential buyers to remotely explore and experience the layout of homes. Finally, AR and VR are popular in the entertainment and gaming industries, providing unparalleled gaming experiences and interactive opportunities.

In AR & VR applications, fidelity is a hard-to-evaluate yet crucial factor. Previous works show evidence that high fidelity enhances user immersion, presence, and co-presence, and hence brings emotional impact on the user such as emotional arousal, enhancement, and emotional interaction.

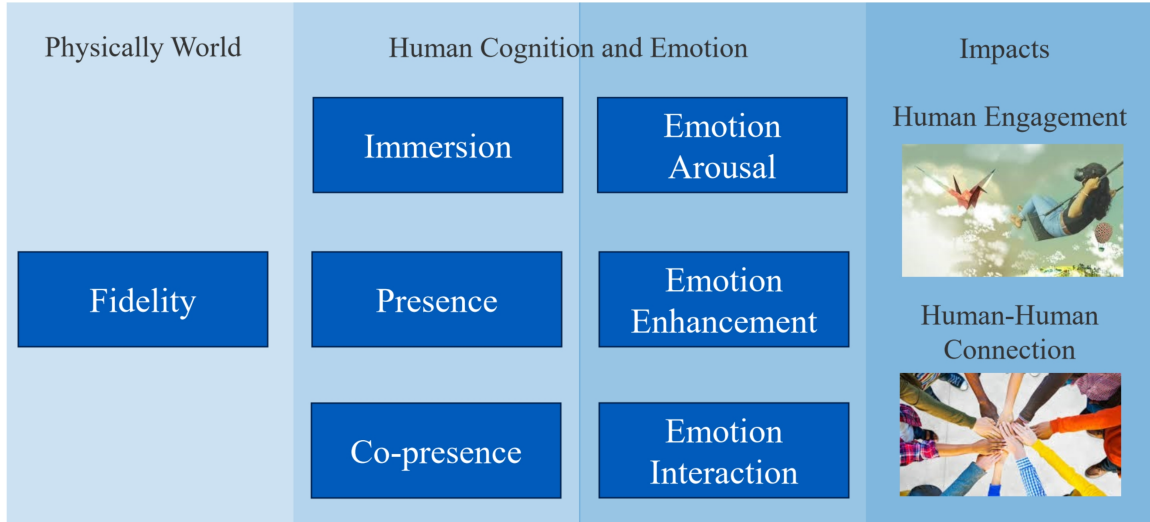


Figure 1.1: The motivation of acquiring high-fidelity in 3D virtual world.

Fidelity, immersion, and emotional arousal. In psychology, immersion is defined as the degree to which an individual feels absorbed by or engrossed in a particular experience [1]. For immersions, experiments in [2] found that looking at the unrealistic scene significantly lowers the immersion feeling score of the subjects compared with looking at the realistic scene. Subsequently, the sense of immersion also impacts people's emotional arousal. Very recent research [3] analyze previous works including [4, 5, 6, 7, 8] and found that higher immersions of nature exposure would significantly decrease the arousal of fatigue in the subjects.

Fidelity, presence, and emotional enhancement. Presence is defined as an experience of being in one place or environment, even when one is physically situated in another [1] (sometimes also called *situated immersion*). [9] proposed a framework named *Servotte-Ghuysen framework* to analyze the sense of presence. In the Servotte-Ghuysen framework, fidelity serves as a crucial system factor for users to have a sense of presence, and a high-fidelity environment can enhance users' sense of presence. Research on VR-related psychology also found that the sense of presence enhances the intensity of people's emotional feelings. For instance, [10] shows that the intensity of the subjects' happiness feeling shows a positive correlation with the sense of presence in a relaxing environment, and a negative

correlation with the sense of presence in an anxious environment, while the intensity of the subjects' anxious and sadness gives the opposite results.

Fidelity, co-presence and emotional interaction. Co-presence exists when people sense that they are able to perceive others and that others are able to actively perceive them [11] found the sense of co-presence increases when the level of realism increases. The relation between co-presence and emotional interaction is not evidently clear yet. In psychology, people would empirically use human interaction behavior to measure the sense of co-presence, such as in [12]. Thus, it is highly possible that the co-presence would strongly affects the emotional interaction among people.

Although fidelity is so important for people's engagement in the VR-world and human-human connection, existing research on high-fidelity 3D worlds remains at a trivial exploratory stage. High-fidelity mesh reconstruction is a widely studied direction, and significant efforts have been made to enhance the details of 3D meshes. But we cannot answer if these efforts truly enhanced fidelity. These works lack rigorous standards to demonstrate whether they have indeed enhanced fidelity. Their evaluations rely on traditional metrics based on L2 distance, such as chamfer distance, and visualization. These evaluation standards have clear flaws when assessing realism. For instance, Figure 1 shows an example concerning chamfer distance, illustrating the misalignment between this metric and human perception of fidelity. Specifically, when we remove the wrinkles from the ground truth mesh (resulting in Mesh *B*), the errors detected by previous metrics are not as significant as when we slightly change the pose of the hand (Mesh *A*). However, humans tend to sense a significant difference between ground truth and Mesh *B*, but barely recognize the difference between ground truth and Mesh *A*. Other previous works evaluated the results through visualization. These measurements may vary for different people, failing to capture a statistical understanding of the realism perceived by the population, that is, how realistic their results appear to the entire population. Moreover, visualizing a few results does not reflect the method's performance, especially since most of these methods do not claim to

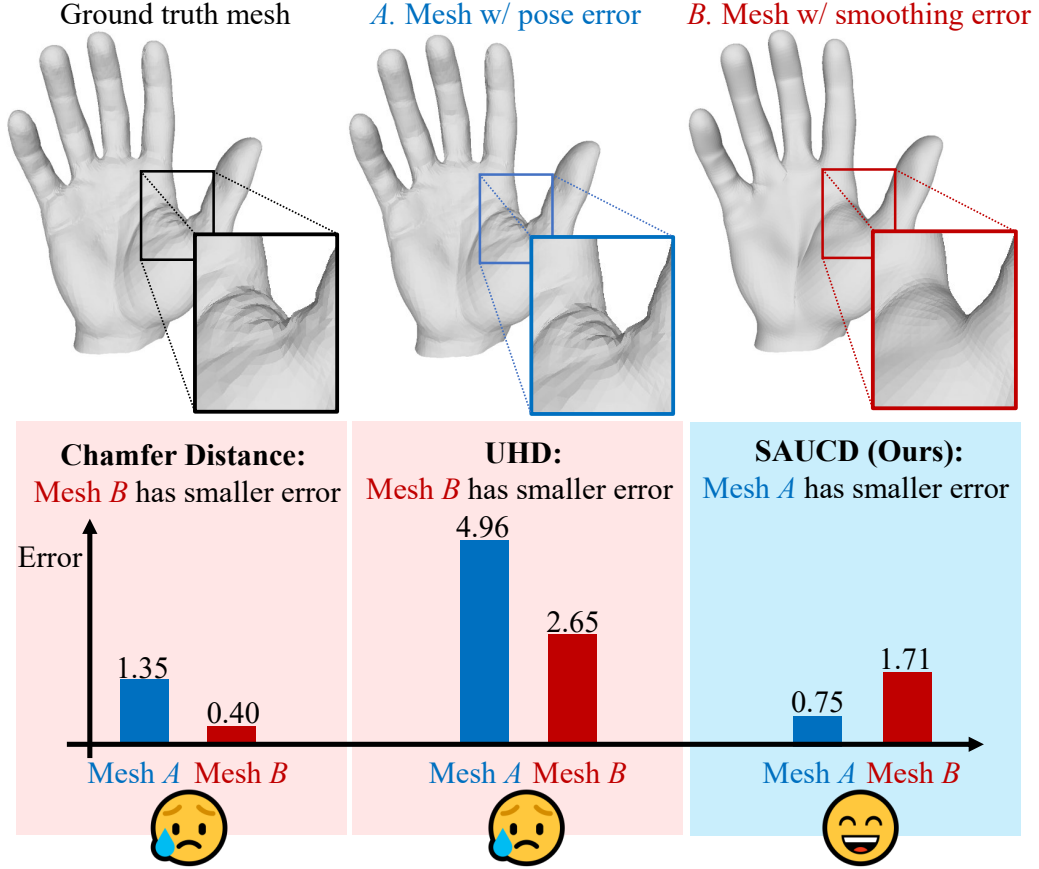


Figure 1.2: An example of how previous spatial domain 3D shape metrics (Chamfer Distance [13] and UHD [14]) deviate from human evaluation. We create **Mesh A** by adding a small pose error to the ground truth mesh, and by applying a large smoothing kernel to ground truth, we create **Mesh B**. Contrary to human perception, previous spatial domain metrics evaluate **Mesh B** better than **Mesh A**. This indicates that while they are sensitive to general shape differences, they tend to overlook high-frequency details. Note that different metrics use different units of measurement.

be randomly selected. These issues are not only present in the 3D reconstruction field. As generative AI evolves, the demand for generating high-fidelity 3D worlds is increasing, and more attempts are being made to produce high-fidelity works, yet no one has answered what fidelity is, how to measure fidelity, and what methods can truly reconstruct and generate fidelity.

We are trying to address these challenges. Initially, we discovered that human perception of fidelity follows predictable patterns. Various papers have studied human sensations of realism and immersion, revealing statistical consistencies in these perceptions. [15]

found that the fidelity scores a group of subjects have on a certain object or behavior have an obvious consistency. For most objects, the standard derivation of scoring distribution would be less than 1 on a scoring scale of 1 to 5. This consistency among subjects enables us to model human’s sense of fidelity. Specifically, it is necessary to collect data on people’s reactions to various 3D objects and scenes first. Based on this data, we then plan to establish a standard for assessing fidelity—a standard that reflects the statistical expectation of evaluations from the entire dataset, not just individual responses. With this established standard, we would further utilize it as a loss function to pursue high-fidelity 3D reconstruction and generation. The detailed contribution can be found in Sec. 3.

2 Problem Formulation

2.1 Fidelity

Fidelity is a personalized concept. The fidelity of the same object can be different in different people’s eyes. Thus, the mathematical definition of fidelity must be based on the statistical result of a group of people’s opinions. Thus, in our research, we defined the fidelity of a 3D shape as:

$$F_r(x, x_0) = E_{s \in S} f_r(x, x_0; s), \quad (1.1)$$

where x_0 is the reference shape and x is the input shape. In this dissertation, we represent the 3D shapes in terms of 3D mesh. S is a collection of subjects. $F_r(\cdot)$ and $f_r(\cdot)$ are the fidelity of subject set S and subject s , respectively. $F_r(\cdot)$ and $f_r(\cdot)$ are both the lower the better.

To build a fidelity metric, we need to design a function to fit $F_r(x, x_0)$ in Eq. (1.1) as

$$\hat{F}_r(x, x_0) = \arg \min_{\hat{F}_r'} \Sigma_x (\hat{F}_r'(x, x_0) - F_r(x, x_0))^2, \quad (1.2)$$

where $\hat{F}_r(\cdot)$ is the fitting function of $F_r(\cdot)$. In practice, $\hat{F}_r(\cdot)$ could be designed as an analytic-based function or a neural network.

2.2 High-fidelity Reconstruction

In high-fidelity reconstruction, we aim to reconstruct a 3D shape that has lower fidelity to the real ground-truth shape. Specifically, high-fidelity reconstruction can be defined as:

$$\hat{x}_r = \arg \min_x \hat{F}_r(x, x_0), x = R(I), \quad (1.3)$$

where \hat{x}_r is the high-fidelity reconstructed shape, $R(\cdot)$ is a reconstruction function, which is typically and neural network. x_0 is the ground-truth shape. Similar to Sec. 2.1, we also represent the 3D shapes in terms of 3D mesh here. I is the input, which could be an image, a coarse shape, or other input data structures. Other variables are defined the same as in Eq. (1.1).

3 Dissertation Overview

This dissertation includes the following four parts: a human-aligned fidelity dataset and metric, fully-supervised high-fidelity hand reconstruction, and self-supervised high-fidelity hand reconstruction. The overview of each part is as follows:

3.1 Fidelity Dataset and Metric

¹Existing 3D mesh shape evaluation metrics mainly focus on the overall shape but are usually less sensitive to local details. This makes them inconsistent with human evaluation, as human perception cares about both overall and detailed shape. In this paper, we propose an analytic metric named Spectrum Area Under the Curve Difference (SAUCD) that demonstrates better

¹This chapter is published in [16].

consistency with human evaluation. To compare the difference between two mesh shapes, we first transform the 3D mesh to the spectrum domain using the discrete Laplace-Beltrami operator and Fourier transform. Then, we calculate the Area Under the Curve (AUC) difference between the two spectrums, so that each frequency band that captures either the overall or detailed shape is equitably considered. Taking human sensitivity across frequency bands into account, we further extend our metric by learning suitable weights for each frequency band which better aligns with human perception. To measure the performance of SAUCD, we build a 3D mesh evaluation dataset called *Shape Grading*, along with manual annotations from more than 800 subjects. By measuring the correlation between our metric and human evaluation, we demonstrate that SAUCD is well aligned with human evaluation, and outperforms previous 3D mesh metrics

3.2 Fully-supervised High-fidelity Hand Reconstruction

²Despite the impressive performance obtained by recent single-image hand modeling techniques, they lack the capability to capture sufficient details of the 3D hand mesh. This deficiency greatly limits their applications when high-fidelity hand modeling is required, *e.g.*, personalized hand modeling. To address this problem, we design a frequency split network to generate 3D hand meshes using different frequency bands in a coarse-to-fine manner. To capture high-frequency personalized details, we transform the 3D mesh into the frequency domain, and proposed a novel frequency decomposition loss to supervise each frequency component. By leveraging such a coarse-to-fine scheme, hand details that correspond to the higher frequency domain can be preserved. In addition, the proposed network is scalable, and can stop the inference at any resolution level to accommodate different hardware with varying computational powers. To feed the scalable frequency network with frequency split image features, we proposed an image-graph ring feature mapping strategy. To train our network with per-vertex supervision, we use a bidirectional registration strategy to generate

²This chapter is published in [17].

a topology-fixed ground-truth. To quantitatively evaluate the performance of our method in terms of recovering personalized shape details, we introduce a new evaluation metric named Mean-frequency Signal-to-Noise Ratio (MSNR) to measure the mean signal-to-noise ratio of mesh signal on each frequency component. Extensive experiments demonstrate that our approach generates fine-grained details for high-fidelity 3D hand reconstruction, and our evaluation metric is more effective than traditional metrics for measuring mesh details.

3.3 Self-supervised High-fidelity Hand Reconstruction

High-fidelity 3D hand reconstruction is essential for immersive AR/VR applications, where users strongly prefer realistic hand representations over simplified meshes. However, acquiring detailed 3D hand data remains challenging, typically requiring complex, expensive equipment that severely limits dataset diversity and scalability. To address these limitations, we propose FlipFlop, a novel self-supervised method that reconstructs textured, high-fidelity 3D hands using only two RGB images (front and back views) without requiring any 3D ground-truth annotations. Our approach seamlessly integrates both general shape information and fine details by combining priors from off-the-shelf models through a frequency-based regulation loss. We also introduce a color regulation loss that encourages the model to represent appearance variations through geometric surface changes rather than merely altering color values. For practical deployment, we offer two complementary workflows: a direct inference pipeline requiring no prior training, and a fast inference approach that delivers quick results after pre-training. To evaluate hand reconstruction quality, we introduce a new benchmark dataset where FlipFlop demonstrates superior performance compared to state-of-the-art methods, particularly in capturing fine surface details while maintaining accurate overall hand structure.

Chapter 2

Human-aligned Fidelity Metric

1 Introduction

With the recent progress of 3D reconstruction and processing techniques, 3D mesh shapes have increasing applications in fields such as video games, industrial design, 3D printing, etc. In these applications, assessing the visual quality of the 3D mesh shape is a crucial task. To meet the requirements of various applications, a promising evaluation metric should not only reflect the geometry measurement but also align with human visual perception. Considering that human beings perceive 3D meshes in both overall shape and local details, it is a challenging task to find an evaluation metric that can align well with humans.

Previous metrics have the following disadvantages in this scenario. Traditional spatial domain measurements such as Chamfer Distance [13] which calculates the mean distance between a vertex on one mesh and its nearest vertex on the other mesh, can accurately measure the spatial distance. However, it does not guarantee capturing all shape details. In fact, such measurements in the spatial domain often overlook finer shape details, as the details tend to get overwhelmed by the overall shape. Fig. 1.2 illustrates the discrepancy between spatial measurements and human evaluation as mesh details change. Specifically, When we remove the wrinkles from the ground truth mesh (resulting in **Mesh B**), the errors

detected by previous metrics are not as significant as when we slightly change the pose of the hand (**Mesh A**). However, humans tend to sense a significant difference between ground truth and **Mesh B**, but barely recognize the difference between ground truth and **Mesh A**. To mitigate this problem, previous works propose learning-based approaches, such as Single Shape Fréchet Inception Distance (SSFID) [18] based on learnable features from 3D shape. They compare the difference between the test mesh and the ground truth mesh in the latent feature space, and the design is expected to better align human perception. However, such learning-based methods would require a large amount of data to train the network. Their accuracy and generalizability are limited by the size of the dataset, data distribution, and annotation quality, not to mention the potential bias in collecting human perception feedback, which could mislead the learned metrics. An analytic metric that can better explain the shape difference is thus preferred.

To address the above limitations, we design an analytic-based 3D shape evaluation metric named Spectrum Area Under the Curve Difference (SAUCD). Our metric measures mesh shape differences with a balanced consideration of both overall and detailed shape, making it better aligned with human evaluation. To allow our metric to capture detail variations, we leverage the 3D shape spectrum to decompose different levels of shape details from the overall shape, with details corresponding to higher-frequency components. The advantage of transforming the shape signal into the spectrum domain is that the high-frequency details are explicitly separated from the low-frequency overall shape. Therefore, it provides appropriate consideration to the information in different frequency bands, not just the low-frequency information of the overall shape in the dominant place. Thus, the details that human perception cares about will be better represented. Besides, the frequency analysis method allows the metric to be mostly analytic and better explained.

We design SAUCD following the above inspiration. To begin with, both the test mesh and the ground truth mesh are transformed from the spatial to the spectrum domain using the discrete Laplace-Beltrami operator (DLBO), which encodes the mesh geometry information

into a semidefinite Laplacian matrix. Once in the spectrum domain, we compare the regions under the two spectrums. Our Spectrum Area Under the Curve Difference metric is defined as the area of the non-overlapping region under the two spectrums – a larger area indicates a greater difference. Moreover, to better align with human evaluation, we further extend our design by learning a spectrum weight for SAUCD. However, different from previous learning-based approaches that use deep networks, large datasets, and extensive learning processes, our learning-based method requires the training of a weight vector. This vector measures the sensitivity of human perception across frequency bands, making the learned metric better aligned with human perception. We then evaluated the effectiveness of SAUCD on our provided user study benchmark dataset named *Shape Grading*. Using *Shape Grading*, we compare our metrics with previous metrics by calculating the correlation between each metric and human scoring.

In summary, our contributions are listed as follows.

- We design an analytic-based 3D mesh shape metric named Spectrum AUC Difference (SAUCD), which evaluates the difference between a 3D mesh and its ground truth mesh. Our metric considers both the overall shape and intricate details, to align more closely with human perception.
- We further extend our design to a learnable metric. The extended metric explores the human perception sensitivity in different frequency bands, which further improves this metric.

Our experiments show that both SAUCD and its extended version outperform previous methods with good generalizability to different types of objects.

2 Related Works

Metrics in 3D mesh reconstruction. Chamfer Distance [13] is a popular metric used in 3D mesh reconstruction tasks such as those in [19, 20, 21, 22, 23, 24, 25, 26]. Other

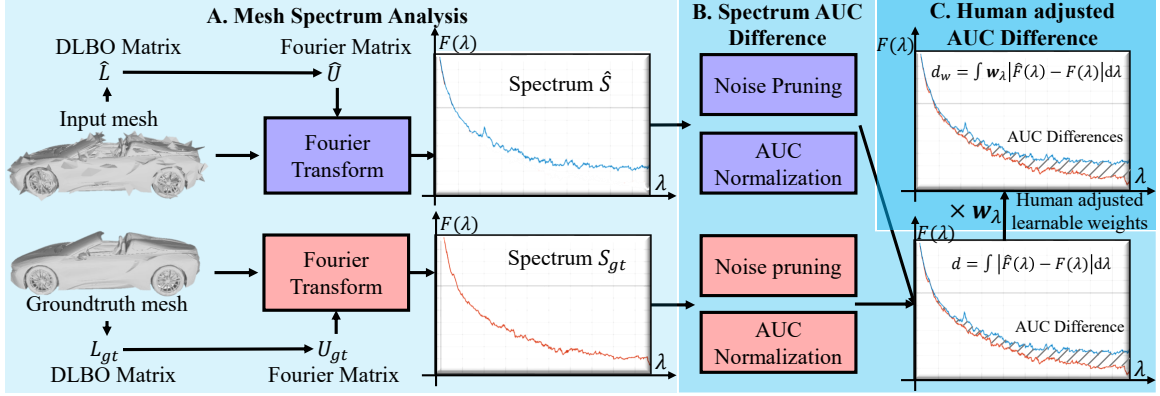


Figure 2.1: Our SAUCD metric is designed as follows: A. We use mesh Fourier Transform to analyze the spectrums of test and ground truth mesh. B. We compare the difference between two spectrum curves by calculating the Area Under the Curve (AUC) difference. C. We further extend our metric by multiplying the AUC difference with a learnable weight to capture human sensitivity in each frequency band.

spatial domain metrics, such as 3D Intersection over Union (IoU) in [27, 28, 29, 30, 31, 32]. F-score in [33, 34, 35, 36], and Unidirectional Hausdorff distance (UHD) in [14] are commonly focused on the geometry accuracy of mesh shapes. These metrics can provide accurate geometry measurements, but they are not designed to align with human evaluation. Deep-learning-based methods such as Single Shape Fréchet Inception Distance [18] are also used in 3D reconstruction. While these metrics have the capacity to adapt from human evaluation, they are more like black boxes, with performances subject to dataset size and annotation bias. Moreover, most previous works miss out on user study validation to verify if their metrics align with human evaluation.

3D shape generation metrics. Multiple metrics have been used in 3D shape generation, such as Minimal Matching Distance (MMD) [37], Jensen-Shannon Divergence (JSD) [38], Total Mutual Difference (TMD) [14], Fréchet Pointcloud Distance (FPD) [39], *etc.*. These metrics are designed to measure the differences between the generated distributions, while our task is to build a metric to compare the shape of two meshes.

3D mesh compression and watermarking metrics. Previous works [40, 41, 42, 43] focused on evaluating mesh errors in mesh compression and watermarking. Since compression and watermarking pursue mesh errors that cannot be detected by humans, they

mainly focus on barely noticed errors. However, our task is to build a metric that can handle generally occurring errors that happen in 3D reconstruction tasks and applications.

3 Proposed Method

Our task is to design a metric aligned with human evaluation to measure the shape difference between a test triangle mesh and its corresponding ground truth triangle mesh. Specifically, given a test mesh \hat{M} and its ground truth mesh M_{gt} , Spectrum AUC Difference (SAUCD) can be abstracted as

$$d = D(\hat{M}, M_{gt}). \quad (2.1)$$

d is the distance between the test and the ground truth mesh. In this section, we will elaborate on how the distance function $D(\cdot)$ is designed.

3.1 Overview

As shown in Fig. 2.1, our metric is calculated via the following steps: First, we use mesh Fourier transform to analyze the spectrums of the test and ground truth mesh (in Sec. 3.2). Then we leverage each frequency band by calculating the Area Under the Curve (AUC) difference of the spectrum curves (in Sec. 3.3). Moreover, we further extended our metric by multiplying the AUC difference with a learnable weight to capture the human sensitivity on each frequency band (in Sec. 3.4). We will discuss each step in detail.

3.2 Mesh spectrum analysis

In order to capture the overall shape as well as shape details, we choose to decompose the mesh signal into a spectrum. Considering the mesh as a function on a discretized manifold space, we can calculate the spectrum using the manifold space Fourier transform. In Hilbert space, the Fourier operator is defined as the eigenfunctions of the Laplacian operator [44].

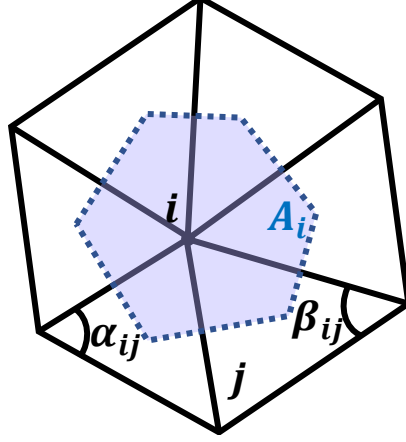


Figure 2.2: Variables defined in our discrete Laplace-Beltrami operator design.

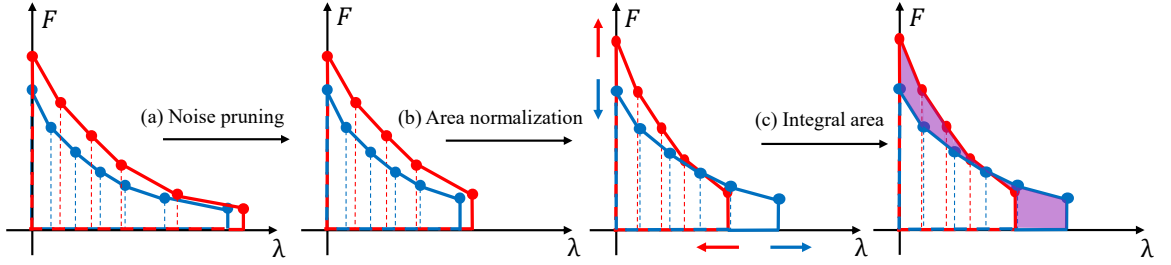


Figure 2.3: Spectrum Area Under the Curve Difference. We design our metric using the AUC difference of the spectrums. The blue curve and red curve are the test and ground truth mesh spectrum, respectively. The purple area in the last graph is the Spectrum AUC Difference. Please find details in Sec. 3.3.

The same definition and similar theories are extended to continuous and discrete manifold space by [45, 46]. The Laplacian operator on discrete manifold spaces, *i.e.* mesh space in our task, is named the Discrete Laplace-Beltrami operator (DLBO). Similar to the Laplacian operator in image space that encodes the pixel information by capturing the local pixel differences [47, 48, 49, 50, 51, 52], DLBO encodes the mesh shape information by capturing the local shape fluctuation. The “Cotan formula” defined in [53] is the most widely used discretization, which can be represented in matrix form as

$$L_{ij} = \begin{cases} \sum_{j \in N(i)} \frac{1}{2A_i} (\cot \alpha_{ij} + \cot \beta_{ij}), & i = j \\ -\frac{1}{2A_i} (\cot \alpha_{ij} + \cot \beta_{ij}), & i \neq j \wedge j \in N(i) \\ 0, & i \neq j \wedge j \notin N(i), \end{cases} \quad (2.2)$$

where $L \in \mathbb{R}^{N \times N}$ is the DLBO matrix, with N the vertex number of the mesh. L_{ij} indicates its entry in i th row and j th column, which represents the edge weight between vertex v_i and v_j . A_i is the mixed Voronoi area of vertex v_i on the mesh. As shown in Fig. 2.2, the v_i 's mixed Voronoi area is defined as the area of the polygon in which the vertices are the circumcenters of v_i 's surrounding faces. $N(i)$ is the index set of v_i 's adjacent vertices. If v_i and v_j are adjacent, α_{ij} and β_{ij} are the opposite angles of edge (v_i, v_j) in each of the edge's two neighbor triangle faces, respectively (shown in Fig. 2.2). If not, α_{ij} and β_{ij} are not defined and L_{ij} is 0. As shown in Fig. 2.1, the DLBO matrix is used for mesh Fourier transform to get mesh spectrum. We calculate the Fourier operator U^\top , which is the eigenfunction of L as

$$L = U \Lambda U^\top, \quad (2.3)$$

where Λ is a diagonal matrix whose diagonal elements are Fourier mesh frequencies.

To ensure the mesh frequencies are non-negative, we need the DLBO matrix L to be positive semidefinite. Our experiment in Fig. 2.9 gives an example of the counterintuitive results when there are negative frequencies. However, the Cotan formula in Eq. (2.2) does not guarantee to be positive semidefinite. We provide a simple example in Chapter A in which L is not positive semidefinite when the mesh is not Delaunay triangulated and the mixed Voronoi areas are not all equal to each other. In our metric design, we made two small changes to the original Cotan formula to make it positive semidefinite. a) Inspired by the symmetric normalization of the topology Laplacian matrix in [54], we make L symmetric by changing the normalization parameter A_i into a symmetric normalized manner $A_i^{\frac{1}{2}} A_j^{\frac{1}{2}}$. b) We replace $\cot \alpha_{ij} + \cot \beta_{ij}$ with $|\cot \alpha_{ij} + \cot \beta_{ij}|$. This ensures all the edge weights in the Laplacian matrix to be non-negative. Thus, our revision of DLBO is defined as

$$L_{ij} = \begin{cases} \frac{1}{2} \sum_{j \in N(i)} A_i^{-\frac{1}{2}} A_j^{-\frac{1}{2}} |\cot \alpha_{ij} + \cot \beta_{ij}|, & i = j \\ -\frac{1}{2} A_i^{-\frac{1}{2}} A_j^{-\frac{1}{2}} |\cot \alpha_{ij} + \cot \beta_{ij}|, & i \neq j \wedge j \in N(i) \\ 0, & i \neq j \wedge j \notin N(i). \end{cases} \quad (2.4)$$

We prove that our revision of the Cotan formula is positive semidefinite in Chapter B. In Tab. 2.5, our experiments show that our DLBO matrix design outperforms the origin Cotan formula in [53], and the topology Laplacian matrix defined in [54].

Finally, we obtain the mesh spectrum by acting Fourier operator on the mesh vertices

$$F_i = \sqrt{G_{i,x}^2 + G_{i,y}^2 + G_{i,z}^2}, G = U^\top v, \quad (2.5)$$

where $v \in \mathbb{R}^{N \times 3}$ indicates the 3D coordinates of N mesh vertices. The result spectrum $F \in \mathbb{R}^N$. Fig. 2.7 shows an example of the mesh spectrum (left side) and how the meshes look in different frequency bands (right side). This provides an illustration of the information contained in different frequency bands of the mesh spectrum.

3.3 Spectrum AUC Difference

To reduce the noise and normalize the mesh scale, we also design noise pruning and AUC normalization procedures before calculating the Spectrum AUC Difference.

Noise pruning. As shown in Fig. 2.3 process (a), we prune a small portion of the highest frequency information to reduce the interference of noise. From the observation of the first two meshes (A and B) in Fig. 2.7, we can see that humans can barely notice the shape differences when the highest frequency parts are removed. Thus, if we try to evaluate the mesh shape that aligns with human perception, it is reasonable to remove high-frequency noise without losing much information that humans care about. Empirically, we choose to prune the highest 0.1% frequency information as noise. Our experiments in Tab. 2.4 show that this portion is more consistent with human perception while preserving good mesh quality.

AUC normalization. Given a spectrum $F(\lambda)$, its Area Under the Curve (AUC) can be defined as $\int_{\infty} F(\lambda) d\lambda$. AUC normalization means using spectrum AUC to normalize the mesh scale. If the mesh scale increases by s times in length, the mixed Voronoi area, *i.e.* A_i

in Eq. (2.4), will decrease by s^2 times. Thus, each element in the DLBO matrix L will also decrease by s^2 times. Then, according to Eq. (2.3), the frequency λ will decrease s^2 times to λ/s^2 , and according to Eq. (2.5), the spectrum amplitude F will change to sF because v is increased by s time and U^\top is still orthonormal. Then the area under the spectrum curve (the area boxed with red or blue lines in Fig. 2.3) changes as $A' = \int sF(\lambda)d\lambda/s^2 = \frac{1}{s}A$.

In our approach, we normalize the area under the spectrum curve A to 1 to resolve the scale difference, which means $s = A$, λ decreases by A^2 times, and spectrum amplitude F increases by A times (Fig. 2.3 process (b)). AUC normalization fairly normalizes the scale of objects in different shapes by only changing the scale, not shape details. It normalizes the spectrum AUC of all mesh to 1, making the mesh spectrums differ only in distributions. Our experiments in Tab. 2.5 demonstrate this design can bring a fairer comparison of the spectrums and improve the human consistency of the metric. The experiment also demonstrates that this design outperforms the spatial domain scale normalization.

Spectrum AUC Difference. In order to capture the difference between two mesh shapes in the spectrum domain, we design Spectrum AUC Difference (SAUCD) on the spectrum analysis results after noise pruning and AUC normalization:

$$d = D(\hat{M}, M_{gt}) = \int_{\lambda} |\hat{F}(\lambda) - F_{gt}(\lambda)| d\lambda, \quad (2.6)$$

where \hat{F} and F_{gt} are the test and groundtruth mesh spectrum. As shown in Fig. 2.3 process (c), our metric is defined as the AUC difference of the two spectrum curves (the purple area). In Tab. 2.5, we compare our design with an alternative design which changes the amplitude difference $|\hat{F}(\lambda) - F_{gt}(\lambda)|$ to energy difference $|\hat{F}(\lambda)^2 - F_{gt}(\lambda)^2|$. The result shows our design is more consistent with human evaluation. Besides, experiments in Tab. 2.1 show our SAUCD metric aligns well with human evaluation, and outperforms SOTA metrics under multiple evaluation methods. Experiments in Fig. 2.10 show SAUCD has the capability to improve mesh detail qualities in 3D reconstruction when adapted into training loss.

3.4 Human-adjusted Spectrum AUC Difference

We also provide an extended metric version, in which we design a learnable weight parameter along the frequency bands. The weight parameter indicates the adjustment of human sensitivity to each frequency band. Specifically, we design the extended metric as

$$d_w = D_w(\hat{M}, M_{gt}) = \int_{\lambda} w(\lambda) |\hat{F}(\lambda) - F_{gt}(\lambda)| d\lambda. \quad (2.7)$$

$w(\lambda)$ is weight parameters indicating human sensitivity along frequency bands. Our training loss is defined as

$$\mathcal{L} = \lambda_p \mathcal{L}_{plcc} + \lambda_{sr} \mathcal{L}_{srocc} + \lambda_r \mathcal{L}_{regu}, \quad (2.8)$$

where \mathcal{L}_{plcc} and \mathcal{L}_{srocc} are Pearson correlation loss and Spearman rank order loss. They are defined the same as Pearson’s linear correlation [55] and Spearman’s rank order correlation [56]. $\mathcal{L}_{regu} = 1/N \sum_i (w_i - 1)^2$ is the regularization loss, which regularizes weight w_i close to 1. λ_p , λ_{sr} , and λ_r are the loss weights of \mathcal{L}_{plcc} , \mathcal{L}_{srocc} , and \mathcal{L}_{regu} . Our experiments in Sec. 5.5 show that after adjustment, the consistency between our metric output and human-annotated ground truth is improved.

3.5 Discretization of Spectrum AUC Difference

Our Spectrum AUC Difference (SAUCD) is defined in Eq. (2.7) as

$$d = D(\hat{M}, M_{gt}) = \int_{\lambda} |\hat{F}(\lambda) - F_{gt}(\lambda)| d\lambda, \quad (2.9)$$

where $\hat{F}(\lambda)$ and $F_{gt}(\lambda)$ are the test and groundtruth mesh spectrum, respectively. To discretize Eq. (2.6) in the experiments, we let $\{\hat{\lambda}_i\}$ to be the discretized frequencies of $\hat{F}(\lambda)$ and $\{\lambda_{gt,i}\}$ to be the discretized frequencies of $F_{gt}(\lambda)$. We sort the two sets $\{\hat{\lambda}_i\}$ and $\{\lambda_{gt,i}\}$ into one array from low to high, resulting in a sorted array $\{\lambda_i\}$ with $N_{gt} + \hat{N}$ frequencies,

where N_{gt} is the vertex number of the groundtruth mesh and \hat{N} is the vertex number of the test mesh. The $N_{gt} + \hat{N}$ frequencies discretize Eq. (2.6) into the sum of the area of $N_{gt} + \hat{N} - 1$ segments as:

$$d = \sum_{i=1}^{N_{gt} + \hat{N} - 1} s_i, \quad (2.10)$$

where the area of each segment

$$s_i = \begin{cases} \frac{1}{2}|H_i + H_{i-1}|(\lambda_i - \lambda_{i-1}), & H_i H_{i-1} \geq 0 \\ \frac{H_i^2 + H_{i-1}^2}{2|H_i + H_{i-1}|}(\lambda_i - \lambda_{i-1}), & H_i H_{i-1} < 0, \end{cases} \quad (2.11)$$

is either a trapezoid when $H_i H_{i-1} \geq 0$ or two triangles when $H_i H_{i-1} < 0$. Here,

$$H_i = \hat{F}(\lambda_i) - F_{gt}(\lambda_i) \quad (2.12)$$

is the amplitude difference between $\hat{F}(\lambda)$ and $F_{gt}(\lambda)$ at λ_i . If λ_i is originally from the test mesh spectrum, then

$$\hat{F}(\lambda_i) = \hat{F}(\hat{\lambda}_i), \quad (2.13)$$

and $F_{gt}(\lambda_i)$ is calculated using interpolation as

$$F_{gt}(\lambda_i) = \frac{(\lambda_{gt,i+} - \lambda_i)F_{gt}(\lambda_{gt,i+}) + (\lambda_i - \lambda_{gt,i-})F_{gt}(\lambda_{gt,i-})}{\lambda_{gt,i+} - \lambda_{gt,i-}}, \quad (2.14)$$

where $\lambda_{gt,i-}$ and $\lambda_{gt,i+}$ are the left and right nearest frequencies of λ_i in the groundtruth frequency set $\{\lambda_{gt,i}\}$. Similarly, if λ_i is originally from the groundtruth mesh spectrum, then

$$F_{gt}(\lambda_i) = F_{gt}(\lambda_{gt,i}), \quad (2.15)$$

and $\hat{F}(\lambda_i)$ is calculated using interpolation as

$$\hat{F}(\lambda_i) = \frac{(\hat{\lambda}_{i+} - \lambda_i)\hat{F}(\hat{\lambda}_{i+}) + (\lambda_i - \hat{\lambda}_{i-})\hat{F}(\hat{\lambda}_{i-})}{\hat{\lambda}_{i+} - \hat{\lambda}_{i-}}, \quad (2.16)$$

where $\hat{\lambda}_{i-}$ and $\hat{\lambda}_{i+}$ are the left and right nearest frequencies of λ_i in the test frequency set $\{\hat{\lambda}_i\}$.

In summary, to calculate the area of the region between the two curves (*i.e.* AUC difference), we first sort the frequencies from the test and groundtruth spectrum in one array, and interpolate the test and groundtruth spectrum using the frequencies from the other spectrum. Then, we calculate each AUC difference in the range between two adjacent frequencies and add them together. When $H_i H_{i-1} \geq 0$, the region between the two curves is a trapezoid; when $H_i H_{i-1} < 0$ the region is two triangles and we calculate the sum area of the two triangles. Finally, the sum of the areas between adjacent frequencies is our Spectrum AUC Difference metric.

3.6 Discretization of Human-adjusted SAUCD

Our Human-adjusted SAUCD is defined in Eq. (2.8) as

$$d = D(\hat{M}, M_{gt}) = \int_{\lambda} w(\lambda) |\hat{F}(\lambda) - F_{gt}(\lambda)| d\lambda. \quad (2.17)$$

Similar to SAUCD discretization, Human-adjusted SAUCD can be discretized as

$$d = \sum_{i=1}^{N_{gt} + \hat{N} - 1} w_i s_i, \quad (2.18)$$

where s_i is defined the same as in Eq. (2.10), and w_i is the human-adjusted weight at λ_i in Eq. (2.11). Since the weight vector \mathbf{w} we use is only 20-dimensional to avoid overfitting, we get each w_i by interpolating \mathbf{w} at each λ_i . Specifically, the 20 elements of \mathbf{w} represent the weights at frequencies uniformly distributed in the range from 0 to 0.05. We denote those 20 frequencies as $\{\lambda_{\mathbf{w},k}\}$ on which the weights \mathbf{w} are explicitly defined, which means $0 \leq k < 20$, $\lambda_{\mathbf{w},0} = 0$, and $\lambda_{\mathbf{w},19} = 0.05$. The last frequency location 0.05 is picked empirically. Note that we use a revised version of Discrete Laplace-Beltrami

Object No.	1	2	3	4	5	6	7	8	9	10	11	12	Overall
Metrics													
Chamfer Distance [13]	0.54	0.15	-0.10	0.57	-0.06	-0.12	-0.20	0.07	0.04	0.30	-0.20	0.17	0.097
Point-to-Surface	0.45	0.19	-0.04	<u>0.66</u>	-0.08	-0.25	-0.32	-0.20	0.01	0.13	-0.21	-0.12	0.017
Normal Difference	0.46	0.11	0.06	0.28	0.11	0.21	0.29	0.47	0.27	0.39	0.11	0.27	0.253
IoU [30]	0.60	<u>0.63</u>	0.01	0.51	0.30	0.02	-0.07	0.20	0.14	0.47	-0.09	-0.01	0.225
F-score [33]	0.58	0.09	0.05	0.33	0.03	0.06	0.16	0.34	0.27	0.25	0.01	<u>0.34</u>	0.208
SSFID [18]	0.71	0.74	-0.04	0.74	0.39	0.24	0.13	0.32	0.25	0.64	0.25	-0.02	0.363
UHD [14]	0.29	0.22	0.11	0.15	-0.04	0.18	0.41	0.55	0.13	0.18	0.25	0.33	0.231
SAUCD (Ours)	<u>0.73</u>	0.21	0.60	0.63	0.31	<u>0.51</u>	0.83	<u>0.65</u>	0.77	0.80	0.69	0.08	<u>0.567</u>
Adjusted SAUCD (Ours)	0.79	0.19	<u>0.56</u>	0.64	<u>0.36</u>	0.54	<u>0.79</u>	0.76	<u>0.75</u>	<u>0.77</u>	<u>0.67</u>	0.36	0.598

a. Pearson's linear correlation coefficient.

Object No.	1	2	3	4	5	6	7	8	9	10	11	12	Overall
Metrics													
Chamfer Distance [13]	0.33	0.14	-0.09	0.43	-0.08	-0.06	-0.15	0.17	-0.04	0.24	-0.16	0.22	0.079
Point-to-Surface	0.42	0.39	0.14	0.59	0.11	0.05	-0.10	0.20	0.18	0.40	-0.11	0.18	0.205
Normal Difference	0.44	0.22	0.33	0.42	0.19	0.29	0.33	0.56	0.33	0.32	0.21	0.34	0.331
IoU [30]	0.57	<u>0.61</u>	0.28	0.50	0.36	0.21	0.12	0.31	0.262	0.56	0.03	0.30	0.342
F-score [33]	0.47	0.25	0.20	0.52	0.21	0.11	0.07	0.36	0.30	0.42	-0.01	0.35	0.27
SSFID [18]	0.63	0.81	0.28	0.70	0.33	0.23	0.10	0.33	0.32	0.65	0.16	0.34	0.407
UHD [14]	0.38	0.20	0.11	0.32	0.13	0.35	0.41	0.60	0.06	0.27	0.37	<u>0.35</u>	0.296
SAUCD (Ours)	<u>0.79</u>	0.25	0.57	0.59	<u>0.36</u>	<u>0.56</u>	0.83	<u>0.79</u>	<u>0.69</u>	0.69	0.83	0.24	<u>0.598</u>
Adjusted SAUCD (Ours)	0.83	0.21	<u>0.55</u>	<u>0.59</u>	0.38	0.60	<u>0.82</u>	0.80	0.69	<u>0.68</u>	<u>0.75</u>	0.42	0.611

b. Spearman's rank order correlation coefficient.

Object No.	1	2	3	4	5	6	7	8	9	10	11	12	Overall
Metrics													
Chamfer Distance [13]	0.25	0.14	-0.08	0.31	-0.04	-0.02	-0.09	0.15	0.013	0.19	-0.07	0.22	0.080
Point-to-Surface	0.33	0.30	0.07	<u>0.45</u>	0.10	0.08	-0.03	0.17	0.13	0.30	-0.01	0.16	0.171
Normal Difference	0.34	0.16	0.17	0.31	0.18	0.22	0.26	0.44	0.25	0.23	0.16	0.27	0.250
IoU [30]	0.42	<u>0.44</u>	0.24	0.37	<u>0.28</u>	0.22	0.14	0.26	0.20	0.41	0.10	0.23	0.275
F-score [33]	0.37	0.17	0.14	0.42	0.15	0.11	0.09	0.28	0.23	0.34	0.01	0.30	0.216
SSFID [18]	0.48	0.62	0.24	0.51	0.25	0.24	0.12	0.29	0.26	0.48	0.17	0.23	0.322
UHD [14]	0.27	0.13	0.07	0.22	0.09	0.26	0.29	0.42	0.048	0.19	0.28	0.24	0.209
SAUCD (Ours)	<u>0.60</u>	0.16	0.42	0.41	0.27	<u>0.45</u>	0.65	<u>0.57</u>	0.55	<u>0.47</u>	0.60	0.19	<u>0.445</u>
Adjusted SAUCD (Ours)	0.64	0.14	<u>0.40</u>	0.41	0.29	0.48	<u>0.63</u>	0.59	<u>0.55</u>	0.45	<u>0.57</u>	<u>0.29</u>	0.453

c. Kendall's rank order correlation coefficient.

Table 2.1: Correlations between different metrics and human annotation. “SAUCD” is our basic version metric. “Adjusted SAUCD” is the human-adjusted version of our metric. The ranges of all three correlation coefficients are $[-1, 1]$, and the higher the better.

Operator (DLBO) as in Eq. (2.4) to make sure $\lambda_i \geq 0$, then to calculate weight w_i whose corresponding $\lambda_i \notin \{\lambda_{w,k}\}$, we only consider when $\lambda_i > 0$. We use interpolation to calculate λ_i as

$$w_i = \begin{cases} \frac{(\lambda_{w,i+} - \lambda_i)w(\lambda_{w,i+}) + (\lambda_i - \lambda_{w,i-})w(\lambda_{w,i-})}{\lambda_{w,i+} - \lambda_{w,i-}}, & 0 < \lambda_i < \lambda_{w,19} \\ \lambda_{w,19}, & \lambda_i > \lambda_{w,19}, \end{cases} \quad (2.19)$$

where $\lambda_{w,i-}$ and $\lambda_{w,i+}$ are the left and right nearest element to λ_i in $\{\lambda_{w,k}\}$.

Having w_i , we can calculate Human-adjusted SAUCD following Eq. (2.18).

dataset	Raw	w/ IQR removal
number of valid scores	24304	23775
Scoring range	[0, 6]	[0, 6]
95% confidence interval	0.318	0.303
Relative 95% confidence interval	5.33%	5.04%

Table 2.2: Dataset statistics and error analysis.

4 Dataset

We build a user study benchmark dataset *Shape Grading* to evaluate whether our metric is aligned with human evaluation. The dataset contains the human evaluation scores for a variety of distorted meshes. Using this dataset, we can calculate the correlation between metric outputs and human evaluation scores to see how aligned the test metrics are to human evaluation.

4.1 Dataset Design

We choose 12 objects as ground truth 3D triangle mesh from public object/scene/human mesh datasets such as [57, 58, 59] and commercial datasets such as [60, 61]. These objects are picked from different categories including humans, animals, buildings, plants, *etc.*. For each object, we synthesize 7 different types of distortions which commonly occur in 3D reconstruction. For each distortion type we synthesize 4 distortion levels, which gives us $7 \times 4 = 28$ distorted objects for every ground truth object. We rotate and render each distorted object into 3 videos using different materials for the mesh. In total, we generate $12 \times 28 \times 3 = 1,008$ distorted mesh videos.

Fig. 2.4 shows the objects in our proposed dataset *Shape Grading* and what the object numbers later experiments in Tab. 2.1. We also show the distortion types that we used in our dataset and how we generate them in Tab. 2.6. Fig. 2.5 shows examples of distorted meshes of different distortion levels in our dataset.

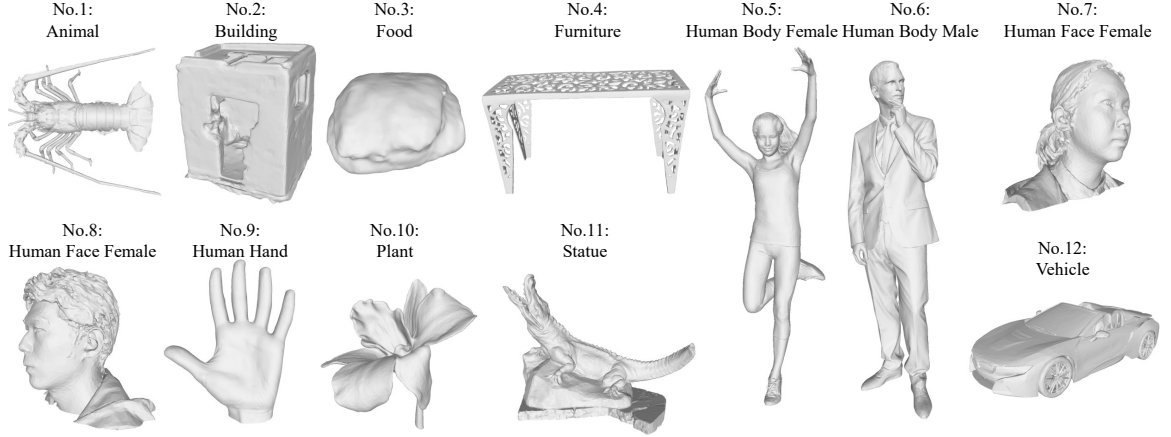


Figure 2.4: Objects in our provided *Shape Grading* dataset and what the object numbers correspond to in Tab. 2.1.

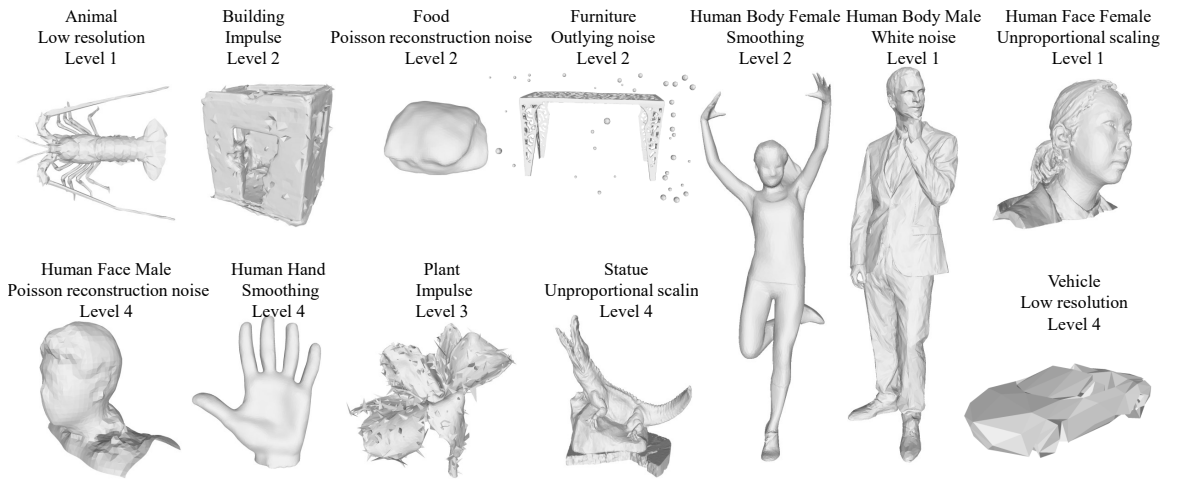


Figure 2.5: Examples of distorted meshes of different distortion levels in our provided *Shape Grading* dataset.

4.2 Human Scoring Procedure Overview

We use a pairwise comparison scoring process similar to [62]. Each subject will evaluate all 28 distorted objects of one ground truth object with a certain material. The scoring follows a Swiss system tournament principle used in [62], in which each subject takes 6 rounds of pairwise comparison to score the distorted meshes. After 6 rounds of scoring, the meshes are scored from 0 to 6. 0 means the object loses in every round and 6 means it wins in every round. This process will largely reduce the biases among subjects, since the subjects are compelled to distribute an equal amount of points to the 28 distorted objects. The process

will take about 15 minutes for each subject, avoiding the fatigue problem in [63]. For every object rendered with every material, we have 24 to 25 subjects scoring it. In total, we have 868 subjects (536 males, 316 females, and 16 others) who give us $868 \times 28 = 24304$ scores.

4.3 Swiss System Tournament for Human Scoring

We do a Swiss system tournament for human scoring. The tournament has 6 rounds. To begin with, all 28 meshes are set to 0 points. In the first round, the 28 meshes are randomly sorted and we form the adjacent meshes into pairs (the 1st and 2nd meshes form a pair, the 3rd and 4th meshes form another pair, *etc.*). Together, we have 14 pairs. For each pair, we ask the subject which one is closer to the groundtruth. The mesh that the subject picked will be added 1 point. From the 2nd to the 6th round, for each round, we first sorted the meshes by their current score from low to high, and we also make pairs with adjacent meshes in the sorted mesh array, like what we did in the first round. The mesh closer to groundtruth will be added 1 point. The scores of the meshes after 6 rounds is their score graded by this subject. Fig. 2.6 shows the panel of our online human scoring page.

4.4 Outlier Detection

We use the interquartile range (IQR) method [64] which is widely used in statistics to detect and remove outliers. For each distorted mesh, we first find the 25 percentile and the 75 percentile of the scores. The score range in between is called the IQR range. We remove the scores that are 1.5IQR smaller than the 25 percentile or 1.5IQR larger than the 75 percentile. Our dataset error analysis in Tab. 2.2 shows, that by removing 2.2% of the scores using IQR, we can decrease the uncertainty of the final scoring result by nearly 6%.

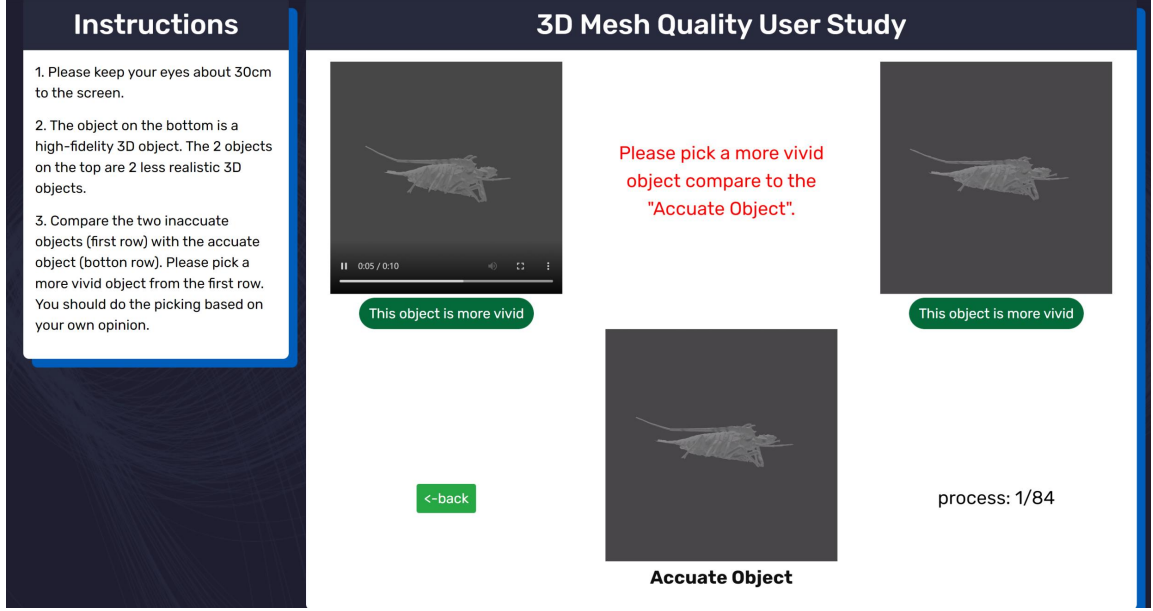


Figure 2.6: The panel of our online user study system. The instruction on the left contains simple instructions for the subjects. On the right side of the page, the top two videos are rendered from distorted meshes. The lower video is rendered from groundtruth mesh.

4.5 Dataset Error Analysis

We analysis the average 95% confidence interval of our dataset scores in Tab. 2.2. The confidence interval of score x can be calculated as $\sigma_{\bar{x}} = z_{0.95} \times \sigma / \sqrt{N}$ where σ is the standard derivation of x , N is the number of valid scores, and $z_{0.95} \approx 1.96$. We report the average 95% confidence interval and the relative 95% confidence interval (which is the confidence interval divided by the scoring range). The result shows that dataset scoring is accurate with a 5% error range with IQR outlier removal.

5 Experiments

5.1 Dataset

We build a user study benchmark dataset *Shape Grading* to evaluate whether our metric is aligned with human evaluation. The dataset contains the human evaluation scores for a

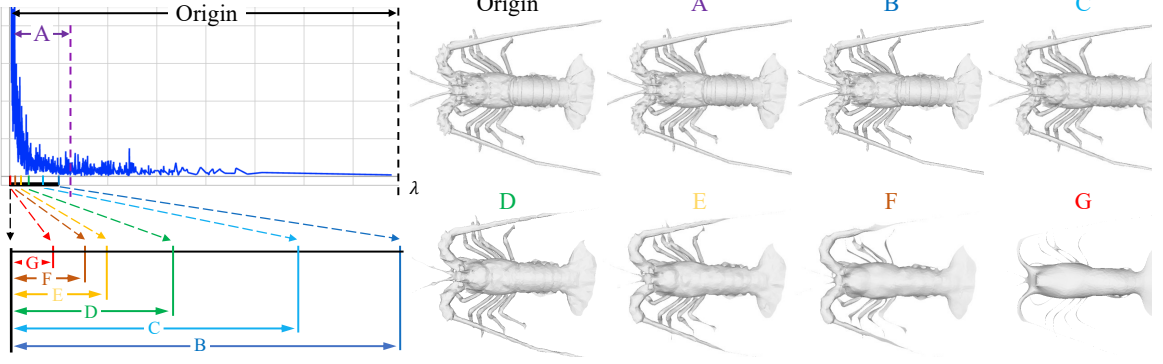


Figure 2.7: An example of mesh spectrum curve: We do mesh Fourier transform on the “Origin” mesh and show the spectrum in the left graph. The λ -axis is the eigenvalues of the DLBO matrix, the larger the higher frequency. We also show how mesh changes when gradually removing high-frequency information (mesh A to G). The frequency bands of the meshes are shown as the colored arrows in the left graph.

variety of distorted meshes. Using this dataset, we can calculate the correlation between metric outputs and human evaluation scores to see how aligned the test metrics are to human evaluation. Details of the dataset can be found in Sec. 4.

5.2 Implementation Details

We implement our basic version metric following Eq. (2.6). $\hat{F}(\lambda)$ and $F_{gt}(\lambda)$ in Eq. (2.6) are both piece-wise functions, so we implement the integration by simply adding every piece area together. We implement our human-adjusted version following Eq. (2.7). We use a 20-dimensional weight $w(\lambda)$ to avoid overfitting. We interpolate w to all frequencies of the ground truth and test meshes and element-wisely multiply them to the spectrums. In spectrum weight training, SROCC and PLCC are used as part of the loss function as Eq. (2.8). KROCC is not used in training but only for testing. We use a k-fold strategy for training the human-adjusted weight. Each time we choose 1 object for testing and the rest 11 objects for training, which means $k = 12$.

5.3 Evaluation Methods

We use 3 different evaluation methods to evaluate the correlation between our metrics and human scoring (groundtruth) on our provided *Shape Grading* dataset.

Pearson’s linear correlation coefficient (PLCC). Pearson’s correlation [55] evaluates the linear alignment between our metrics and human evaluation. It is defined as

$$p = \frac{\sum_{i=1}^N (h_i - \bar{h}_i)(m_i - \bar{m}_i)}{\sqrt{\sum_{i=1}^N (m_i - \bar{m}_i)^2} \sqrt{\sum_{i=1}^N (h_i - \bar{h}_i)^2}}, \quad (2.20)$$

where m_i is the score of mesh i given by the tested metric and h_i is the groundtruth score (human scoring) of mesh i . \bar{h}_i and \bar{m}_i are the average score of h_i and m_i , respectively.

Spearman’s rank order correlation coefficient (SROCC). SROCC [56] is one of the most commonly used metrics to measure rank correlations. It is defined as

$$r_s = 1 - \frac{6 \sum (R(m_i) - R(h_i))^2}{n(n^2 - 1)}, \quad (2.21)$$

where m_i and h_i is are defined the same as in Eq. (2.20). $R(m_i)$ and $R(h_i)$ are the rankings of m_i and h_i , and n is the amount of data. In our paper, n is the number of meshes scored by one subject.

Kendall’s rank order correlation coefficient (KROCC). KROCC [65] is also a rank order correlation. It is defined as

$$\tau = 1 - \frac{2}{n(n^2 - 1)} \sum_{i < j} \text{sgn}(m_i - m_j) \text{sgn}(h_i - h_j), \quad (2.22)$$

where m_i , h_i , and n is the same with Eq. (2.21), and $\text{sgn}(\cdot)$ is the sign function, which means $\text{sgn}(x) = 1$ when $x > 0$, $\text{sgn}(x) = -1$ when $x < 0$, and $\text{sgn}(x) = 0$ when $x = 0$. The difference between SROCC and KROCC is, SROCC considers the actual amount of rank order difference of input data, while KROCC only counts the number of inverse pairs.

The possible ranges of all 3 metrics are $[-1, 1]$. Higher numbers mean stronger correlations.

5.4 Human-adjusted SAUCD Training

During training, Pearson’s correlation loss \mathcal{L}_{plcc} and Spearsman’s rank order loss \mathcal{L}_{srocc} in Eq. (2.8) are defined the same as Eq. (2.20) and Eq. (2.21), respectively. Note that, since the rank part of SROCC is not naturally differentiable, we used a differentiable ranking approach provided in [66] to make Eq. (2.21) differentiable. We set $\lambda_p = 0.1$, $\lambda_{sr} = 10$, and $\lambda_{regu} = 1$ for Eq. (2.8). The training process took about 1 minute on a 14-core Intel Xeon CPU. The training code is implemented using PyTorch [67].

5.5 Quantitive and Qualitative Results

SOTA comparison. Tab. 2.1 shows our results compared to previous 3D mesh shape metrics. We evaluated the correlation between each metric and the human scoring via three different evaluation methods. We observe that **a)** without any learning-based design, our metric outperforms the SOTA learning-based (SSFID) and non-learning-based metrics (Chamfer Distance, IoU, F-score, and UHD), **b)** our extended version metric with learned weights has better linearity and slightly better ranking order correction with human evaluation, and **c)** our results on different objects show that our metrics have good generalizability.

Spectrum example. We first show an example of mesh spectrum in Fig. 2.7. We decompose the “origin” mesh using the Fourier Transform and get the resulting spectrum (top-left graph). The meshes on the right (from mesh **A** to **G**) are generated by gradually removing high-frequency information. The frequency bands of the meshes are shown as colored arrows in and under the graph. As we see, the details gradually disappear as we remove high-frequency information.

Frequency band separation. We explored the consistency between human perception and the information obtained from every frequency band. Specifically, we separate

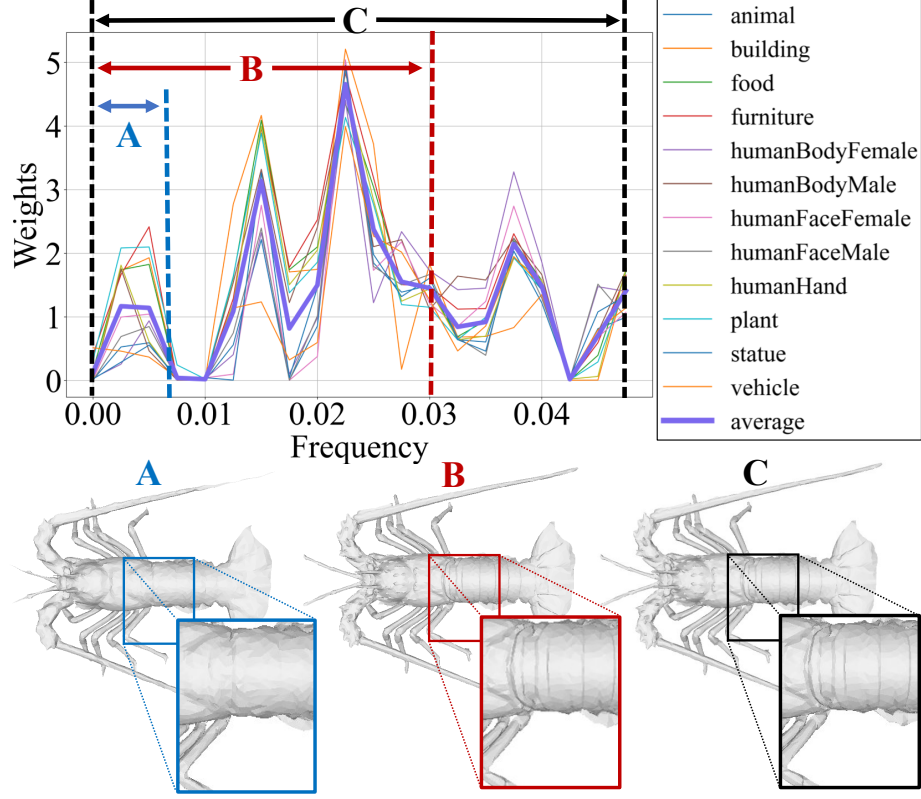


Figure 2.8: Learned spectrum weights on all 12 folds. The name of colorful thin lines means the test object name of that fold. The bold **purple** line is the average weights of all folds. We also show some examples of mesh shape information in different frequency bands. Frequency band **A** is $[0, 0.0075)$, **B** is $[0, 0.03)$, and **C** is $[0, 0.05)$.

the frequency band exponentially and build metrics only using information from that frequency band. The results are shown in Tab. 2.3, we find the frequency bands $[0, 0.001]$ and $[0.01, 0.03]$ have the best consistency with human perception. Moreover, it shows that if we put all frequencies together, they can achieve better results.

Trained weight. We show our trained weights in Human-adjusted metric in Fig. 2.8. Different lines represent different folds, and the bold **purple** line is the average weight. We can see the weights trained on each fold have similar patterns. We also observe that the weight curves have a small peak in the range **A** and two much larger peaks between **A** and **B**, which means our extended metric relies more on the information between **A** and **B**. We show an example of mesh shapes in the range **A**, **B**, and **C** at the bottom of Fig. 2.8. Mesh **A** obviously has fewer details than Mesh **B**, and the weight curve shows that this difference is

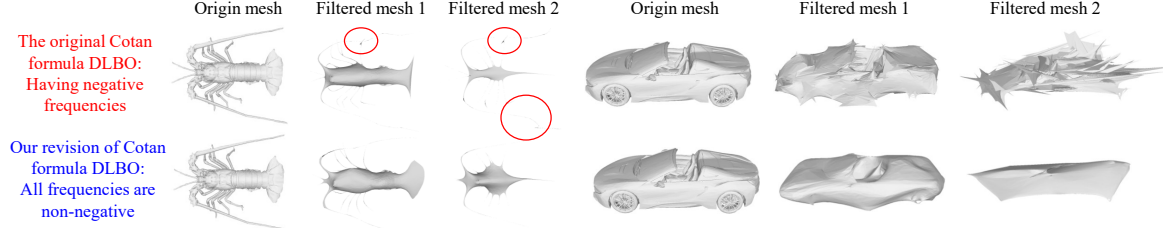


Figure 2.9: Counterintuitive low-frequencies information if some of the mesh frequencies are negative. We can see if we remove the high-frequency part of the mesh (resulting in “Filtered mesh 1” and “Filtered mesh 2”) using the original Cotan formula, the mesh’s low-frequency parts show artifacts (sharp shapes). The **red** circles show the artifacts in the left object. The right object shows a case when these artifacts occur much more often. These artifacts do not occur using our revised Cotan formula DLBO.

Table 2.3: Results when building metrics using each frequency band separately. The bottom row is our proposed metric.

Frequency band	PLCC \uparrow	SROCC \uparrow	KROCC \uparrow
$[0, 0.001)$	0.434	0.515	0.376
$[0.001, 0.003)$	0.240	0.409	0.281
$[0.003, 0.01)$	0.255	0.455	0.340
$[0.01, 0.03)$	0.421	0.528	0.391
$[0.03, 0.1)$	0.287	0.351	0.250
$[0.1, \infty)$	0.318	0.192	0.155
$[0, \infty)$	0.567	0.598	0.445

what the learning process tries to emphasize.

Negative frequencies. In Fig. 2.9 we illustrate how our revised Cotan formula DLBO in Eq. (2.4) improves frequency analysis compared to the original Cotan formula in Eq. (2.2) [53]. The first and second rows are the results of the original Cotan formula DLBO and our revised Cotan formula DLBO, respectively. The original Cotan formula can yield negative frequencies due to its lack of positive semidefiniteness, whereas our revision ensures all frequencies are non-negative. For both objects in the figure, we remove different portions of high-frequency information and show the remaining low-frequency parts (resulting in “Filtered mesh 1” and “Filtered mesh 2”). For the left object, notice the counterintuitive sharp shapes in the **red** circle when using the Cotan formula. The right object is a much more severe case. Sharp shapes in low-frequency parts show improper decomposition and high-frequency aliasing with low-frequency shapes, making the Cotan

Table 2.4: Results with different pruning portions. The metric achieves better results with pruning portion to be 0.1% or 1%. We use pruning portion as 0.1% in our design.

Pruning Portion	PLCC \uparrow	SROCC \uparrow	KROCC \uparrow
0%	0.513	0.549	0.393
0.1%	0.567	0.598	0.445
1%	0.554	0.602	0.462
10%	0.517	0.581	0.442
20%	0.503	0.587	0.445

Table 2.5: Module replacement. We replace each module of our metric with alternative designs to verify the design of each module.

Modules	PLCC \uparrow	SROCC \uparrow	KROCC \uparrow
Topology Laplacian [54]	0.298	0.327	0.235
Cotan formula [53]	0.417	0.470	0.340
Energy difference	0.268	0.315	0.215
w/o normalization	0.257	0.507	0.353
Spatial normalization	0.269	0.542	0.392
Ours	0.567	0.598	0.445

formula unsuitable for spectral mesh comparison. In contrast, our revised formula yields smooth low-frequency components without these artifacts.

Noise pruning portion. Tab. 2.4 shows our SAUCD metric performance by changing the noise pruning portion (Sec. 3.3). The metric achieves better results when the pruning portion is 0.1% or 1%. In our proposed metric, we choose the pruning portion to be 0.1% to best avoid possible high-frequency information loss.

Module replacement. Tab. 2.5 shows our SAUCD metric performance by replacing some modules with alternative designs. First, we replace our revision of the discrete Laplace-Beltrami operator in Eq. (2.4) with topology Laplacian matrix in [54] and “Cotan formula” in [53]. Second, we change the AUC difference defined in Eq. (2.6) into the energy difference, which means changing $|\hat{F}(\lambda) - F_{gt}(\lambda)|$ in Eq. (2.6) into $|\hat{F}(\lambda)^2 - F_{gt}(\lambda)^2|$. In the third experiment, we replace AUC normalization (in Sec. 3.3) with spatial normalization, where we normalize the meshes by their maximum range along all 3 spatial axes. We also removed the AUC normalization module for another comparison. Our experiments show SAUCD has better performance than alternative designs.

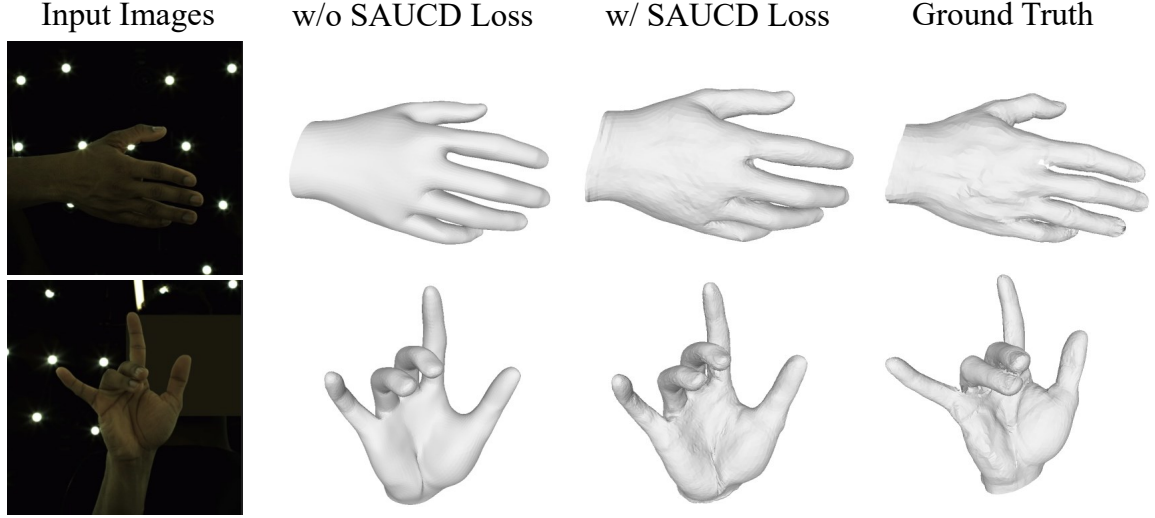


Figure 2.10: We Adapt SAUCD into a loss function and use it in monocular-image-based 3D hand reconstruction. From left to right: input images, reconstruction result w/o SAUCD loss, reconstruction result w/ SAUCD loss, and ground truth mesh. We can see that the enhancement of SAUCD loss in mesh details is clearly noticeable.

Adapting SAUCD to loss function. We adapted our metric into a loss function to enhance the visual quality of 3D mesh reconstructions, as evident from the hand reconstruction results in Fig. 2.10. Specifically, we adapt SAUCD to a topology Laplacian version. Specifically, we replace the Laplacian matrix defined in the main paper Eq.(4) to $L = D - A$ defined in [54], where D is the degree matrix of the mesh graph, and A is the adjacency matrix of the mesh graph. By making the change, we can avoid calculating a different SVD decomposition in every training iteration when mesh vertex locations change. Our network is designed as Fig. 2.13. The input image first goes through a feature extraction CNNs network to get image features, and using that feature to generate MANO [68] mesh. Then, we use features from CNNs network and 3 resolution levels of Graph Convolution Networks (GCN) to reconstruct the mesh details. In the main paper Fig. 8, we compare the results using only MVPE loss (w/o SAUCD loss column) and using both MVPE and SAUCD loss (w/ SAUCD loss column). In this experiment, we use EfficientNet [69] and GCN similar to [70].

How mesh scale changes using AUC Normalization We use AUC normalization on

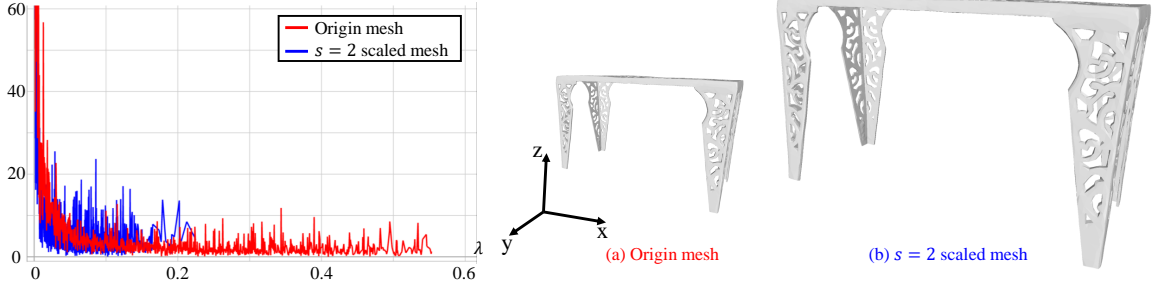


Figure 2.11: AUC normalization. We normalize the spectrum of **origin mesh** with factor $s = 2$. The **blue** curve is the resulting curve. We transform the blue curve back to $s = 2$ **scaled mesh**. As we see on the right side, the mesh’s general shape is kept the same, but the scales increased to twice the size of the original mesh.

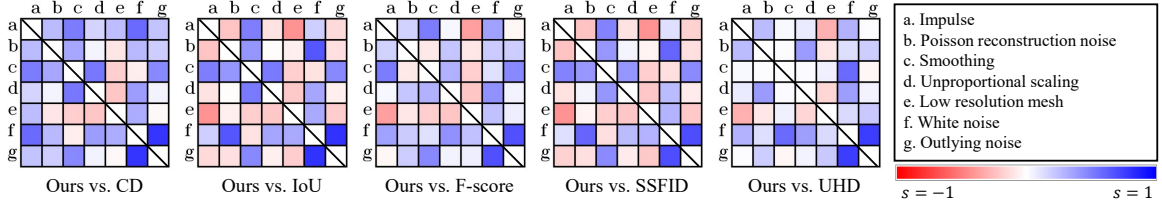


Figure 2.12: Pair-wise distortion type comparisons. s is the percentage difference of the inverse-order pairs compared to groundtruth. **Blue** color means s is larger than 0, which shows that our metric is better than compared metric among the meshes of distortion pair (d_1, d_2) . **Red** color means s is smaller than 0, which means the compared metric is better.

the spectrum curve to normalize the mesh scale. In Fig. 2.11, we show an example of how the mesh changes when we normalize the spectrum with factor $s = 2$ in Eq. (2.6). As we see, the mesh shape is kept the same, but the scales are changed proportionally with the normalization factor s .

Pair-wise distortion comparison. We explore how our metric performs when leveraging different types of distortions. As shown in Fig. 2.12, we compare our metric with previous metrics considering only a pair of distortion types. The score in the boxes is calculated as:

$$s = \frac{\Delta_{ours}(d_1, d_2)}{2N^2} - \frac{\Delta_{prev}(d_1, d_2)}{2N^2}. \quad (2.23)$$

Here, $\Delta_{ours}(d_1, d_2) = \sum_{i,j} \text{sgn}(m_{ours,d_1,i} - m_{ours,d_2,j}) \text{sgn}(h_{d_1,i} - h_{d_2,j})$ indicates the number of reverse-order pairs comparing our metric to groundtruth among all levels of d_1 and d_2 type distortions, and $\Delta_{prev}(d_1, d_2) = \sum_{i,j} \text{sgn}(m_{prev,d_1,i} - m_{prev,d_2,j}) \text{sgn}(h_{d_1,i} - h_{d_2,j})$

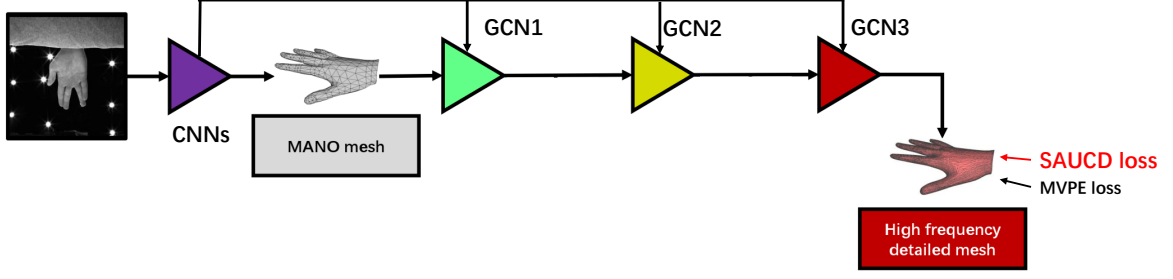


Figure 2.13: Network architecture used when adapting SAUCD to training loss.

Groundtruth mesh	Mesh w/ distortions				
User study \uparrow	1.23	3.74	2.36	4.57	2.63
Ours \downarrow	0.72	1.13	0.80	0.92	0.65
Ours extended \downarrow	1.06	1.96	1.12	1.34	1.02

Figure 2.14: Failure cases. We show a case in which our metric does not provide accurate evaluations aligned with the human evaluation.

means that of previous method. m_{ours} and m_{prev} are the scores obtained from our metric and the previous metric, respectively. h is the groundtruth score (human annotation). d_1 and d_2 indicate 2 kinds of distortions. i and j indicate the i th and j th level of distortions. N is the number of data. $sgn(\cdot)$ is sign function, which means $sgn(x) = 1$ when $x > 0$, $sgn(x) = -1$ when $x < 0$, and $sgn(x) = 0$ when $x = 0$. s 's range is $[-1, 1]$. When our metric is correct and the compared metric is incorrect for every pair, $s = 1$; When our metric is incorrect and the compared metric is correct for every pair, $s = -1$. We performed the experiment with 5 previous metrics as shown in Tab. 2.1. We observe that our metric has generally better results than previous metrics in most distortion pairs. When one of the distortion types in the pair is “Smoothing”, “Impulse”, “White noise”, or “Outlying noise”, our metric tends to have better human perception alignment. When the pair includes “Low-resolution mesh”, our metric does not align with humans very well. A possible reason is that some low-resolution meshes have much fewer vertices than groundtruth meshes, and

fewer vertices would cause larger noise when estimating the discrete spectrum.

Visualized evaluation results. We show more examples in our dataset and evaluation results using different metrics in Fig. 2.15. Compared to previous methods, our provided metrics generally align better with the human evaluation of mesh shape similarity.



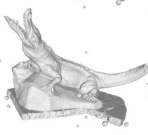
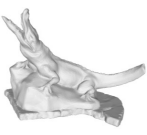

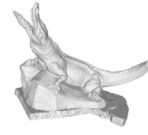

Failure cases. We also show a case that our metric does not provide accurate evaluations aligned with the human evaluation in Fig. 2.14.








6 Conclusions

In order to propose a 3D shape evaluation that better aligns with human perception, we design an analytic metric named Spectrum AUC Difference (SAUCD). Our proposed SAUCD leverages mesh spectrum analysis to evaluate 3D shape that aligns with human evaluation, and its extended version Human-adjusted SAUCD further explores the sensitivity of human perception of each frequency band. To evaluate our new metrics, we build a user study dataset to compare our metrics with existing metrics. The results validate that both our new metrics are well aligned with human perceptions and outperform previous methods.

Distortion types	Description	Generating details
Impulse	Adding impulsive noise on mesh surface	We add Gaussian noise on r percent of the ground truth mesh vertices. The mean of the Gaussian noise is set to 0 and standard derivation is set to σ percent of the mesh scale. For 4 levels of this distortion, (r, σ) are set to (1, 0.5), (5, 2), (8, 3), and (1, 5), respectively.
Poisson reconstruction noise	Synthesizing the noise occurs in Poisson reconstruction [71]	We first use Poisson disk sampling [72] to sample sN points from the groundtruth mesh surface, where N is the number of vertices in groundtruth mesh. Then, we use Poisson reconstruction provided in MeshLab [73] to reconstruct the mesh surface from the sampled points. The reconstruction depth is set to 6. For 4 levels of this distortion, s is set to 0.9, 0.5, 0.2, and 0.05, respectively.
Smoothing	Smoothing mesh surface	We apply i times of $\lambda - \mu$ Taubin smoothing [74] to smooth the groundtruth mesh surface, where $\lambda = 0.5$ and $\mu = -0.53$. For 4 levels of this distortion, i is set to 5, 20, 50, and 200, respectively.
Unproportion scaling	Stretching (or shrinking) the mesh along x , y , and z axis with different rates	We stretch the mesh to s_x percent to its original length along x axis, and shrink the mesh to s_z percent to its original length along z axis. For 4 levels of this distortion, (s_x, s_z) are set to (98, 102), (95, 105), (90, 110), (80, 120), respectively.
Low-resolution mesh	Simplifying mesh surface to lower resolution	We simply the groundtruth mesh surface using edge collapse algorithm [75]. For 4 levels of this distortion, the target face number is set to 5000, 2000, 1000, and 500, respectively.
White noise	Adding Gaussian white noise on mesh surface	We add Gaussian noise on <i>all</i> the groundtruth mesh vertices. The mean of the Gaussian noise is set to 0 and standard derivation is set to σ percent of the mesh scale. For 4 levels of this distortion, σ is set to 0.1, 0.2, 0.3, and 0.5, respectively.
Outlying noise	Adding outlying small floating spheres around the mesh	We add floating spheres around the groundtruth mesh to synthesize outlying noise that occurs in 3D reconstruction. The number of the spheres is set to n and the radius rA , where A is the maximum length of the mesh along x , y , and z dimensions. The locations of the spheres are sampled randomly from a cube that surrounds the groundtruth mesh. The edge size of the cube is set to $(1 + 6r)A$. For 4 levels of this distortion, (n, r) are set to (20, 0.002), (30, 0.004), (40, 0.006), (80, 0.008), respectively.

Table 2.6: Distortions in our provided *Shape Grading* dataset.

Groundtruth mesh	Mesh w/ distortions					
						
User study↑	4.70	3.93	3.90	2.38	4.81	2.37
Ours↓	0.30	0.42	0.41	1.35	0.27	0.89
Ours extended↓	0.23	0.31	0.35	1.20	0.21	0.73
Chamfer Distance↓	0.07	29.52	3.10	5.02	12.62	4.83
IoU↑	1.00	0.24	0.97	0.95	0.68	0.94
F-score↑	1.00	0.93	1.00	1.00	0.95	1.00
SSFID↓	0.00	1.38	0.01	0.02	0.05	0.08
UHD↓	12.60	0.00	30.13	37.96	36.93	14.12

Groundtruth mesh	Mesh w/ distortions					
						
User study↑	4.40	2.63	4.03	0.51	3.68	4.66
Ours↓	0.46	1.08	0.51	1.22	0.89	0.48
Ours extended↓	0.36	0.92	0.43	1.14	0.90	0.38
Chamfer Distance↓	0.006	1.32	1.86	2.45	0.44	0.63
IoU↑	1.00	0.87	0.24	0.09	0.89	0.92
F-score↑	1.00	0.95	0.94	0.85	1.00	1.00
SSFID↓	0.0002	0.04	0.44	8.57	0.03	0.02
UHD↓	1.03	6.72	0.51	1.22	0.89	0.48








Groundtruth mesh	Mesh w/ distortions					
						
User study↑	4.64	2.74	4.10	1.87	4.67	3.01
Ours↓	0.52	7.02	0.53	1.25	0.55	0.94
Ours extended↓	0.70	1.43	0.76	1.99	0.80	1.19
Chamfer Distance↓	0.01	1.26	2.12	1.13	0.69	0.30
IoU↑	1.00	0.96	0.29	0.81	0.93	0.97
F-score↑	1.00	0.97	0.93	1.00	1.00	1.00
SSFID↓	0.0001	0.01	0.59	0.11	0.03	0.004
UHD↓	1.45	1.02	0.53	1.25	0.55	0.94

Figure 2.15: Examples in our dataset and their evaluation results using different metrics. ↓ means lower is better. ↑ means higher is better. For each object, the mesh on the top-left is the groundtruth mesh, and the rest meshes are distorted meshes. The table below the meshes contains the scores they get from different metrics or from our user study. As shown in the figure, our metric aligns better with user study scores and human perception.

Chapter 3

Fully-supervised High-fidelity Hand Reconstruction

1 Introduction

High-fidelity and personalized 3D hand modeling have seen great demand in 3D games, virtual reality, and the emerging Metaverse, as it brings better user experiences, *e.g.*, users can see their own realistic hands in the virtual space instead of the standard avatar hands. Therefore, it is of great importance to reconstruct high-fidelity hand meshes that can adapt to different users and application scenarios.

Despite the previous successes in 3D hand reconstruction and modeling[76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90], few existing solutions focus on enriching the details of the reconstructed shape, and most current methods fail to generate consumer-friendly high-fidelity hands. When we treat the hand mesh as graph signals, like those of most natural signals, the low-frequency components have larger amplitudes than those of the high-frequency parts, which can be observe in a hand mesh spectrum curve (Fig. 3.1). Consequently, if we generate the mesh purely in the spatial domain, the signals of different frequencies could be biased, thus the high-frequency information can be easily overwhelmed

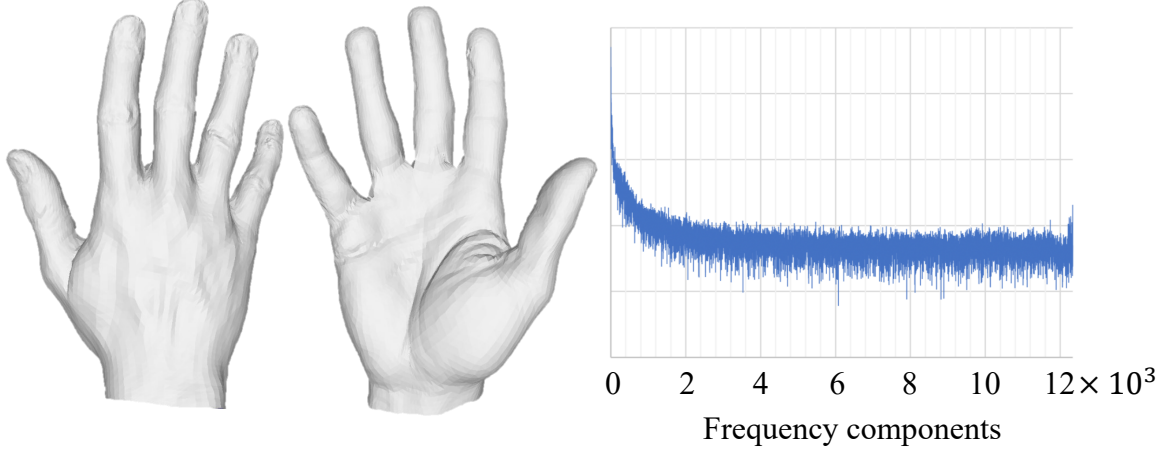


Figure 3.1: An exemplar hand mesh with sufficient details and its graph frequency decomposition. The x-axis shows frequency components from low to high. The y-axis shows the amplitude of each component on a logarithm scale. At the frequency domain, the signal amplitude generally decreases as the frequency increases.

by its low-frequency counterpart. Moreover, the wide usage of compact parametric models, such as MANO [68], has limited the expressiveness of personalized details. Even though MANO can robustly estimate the hand pose and coarse shape, it sacrifices hand details for compactness and robustness in the parameterization process, so the detail expression ability of MANO is suppressed.

Additionally, in the previous methods, transferring the detailed information from images to the mesh was also a significant challenge. As shown in Fig. 3.2, previous works such as [70] use a simple pooling to attach global features to mesh vertices (first row in Fig. 3.2), but global pooling would eliminate the shape detail information, which is needed to reconstruct high-fidelity hands. Other methods such as [81] use a projection strategy (second row in Fig. 3.2). The features on the mesh vertices are extracted from the projected location of the vertices on the feature map, but this method is sensitive to projection accuracy. A small projection error would result in very different local details in the image, thus influencing the reconstruction of hand details. Furthermore, datasets used in previous works lack mesh annotations that are rich in detail and correct topological structures, which are essential for training high-fidelity hands. Existing multiview hand datasets such as [91] can generate

scanned 3D hand meshes using multiview stereo methods, but a scanned hand mesh does not guarantee a complete hand topological structure. As shown in Fig. 3.3, when different parts of the hand are close, such as when fingers are touching or the hand is clenched into a fist, scanning only captures the outer surface shape of the hand. This results in a loss of shape and topological information for the inner areas.

To better model detailed 3D shape information, we transform the hand mesh into the graph frequency domain, and design a frequency-based loss function to generate high-fidelity hand meshes in a scalable manner. Supervision in the frequency domain explicitly constrains the signal of a given frequency band from being influenced by other frequency bands. Therefore, the high-frequency signals of hand shape will not be suppressed by low-frequency signals despite the amplitude disadvantage. To improve the expressiveness of hand models, we design a new hand model of 12,337 vertices that extends previous parametric models such as MANO with nonparametric representation for residual adjustments. While the nonparametric residual expresses personalized details, the parametric base ensures the overall structure of the hand mesh, *e.g.*, reliable estimation of hand pose and 3D shape. Instead of fixing the hand mesh resolution, we design our network architecture in a coarse-to-fine manner with three-resolution-level U-net for scalability. Different levels of image features contribute to different levels of detail. Specifically, we use low-level features in high-frequency detail generation and high-level features in low-frequency detail generation. At each resolution level, we use a Graph Convolution Network (GCN) to generate hand meshes, and our network outputs a hand mesh with the corresponding resolution. During inference, the network outputs an increasingly higher resolution mesh with more personalized details step-by-step, while the inference process can stop at any one of the three resolution levels.

To retain high-frequency information from image features and apply it to the Graph Convolutional Network (GCN) to reconstruct high-fidelity hands, we designed an Image-Graph Ring Frequency Mapping (IGRFM) module to transform image features into graph features via the frequency domain. IGRFM converts image information into frequency

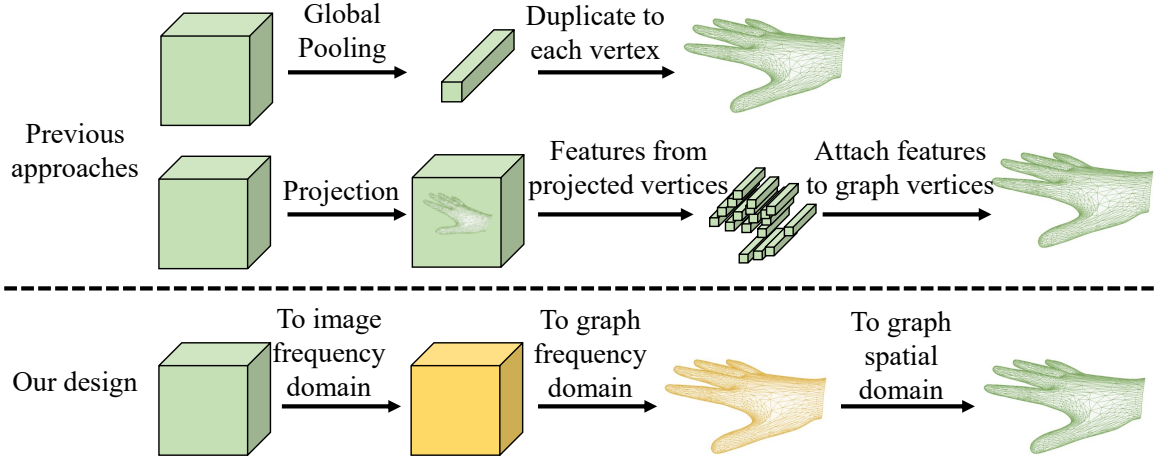


Figure 3.2: To map the image features to graph features, previous methods use a simple pooling strategy (first row) or a projection-interpolation strategy (second row), but the detailed high-frequency features would be easily damaged by pooling or small projection errors. In this paper, we propose an Image-Graph Ring Frequency Mapping (IGRFM) (third row) to map the image features to the graph features via the frequency domain.

domain signals, segregates and reassembles information from different frequency bands, and uses this information as the frequency feature of the GCN. It then transforms the frequency feature through a Graph Inverse Fourier Transform (Graph IFT) into per-vertex mesh spatial features. This design allows for a direct correspondence between different frequency band features in the image and the graph, enabling the reconstruction of high-fidelity hands to explicitly utilize information from the image across various frequency bands. To generate mesh annotations for training that are rich in detail and with correct topological structures, we designed a method for registering scanned mesh with a parametric model. Regardless of the occlusions or changes in the topology of the scanned mesh, our registration method consistently produces ground-truth meshes with detailed shapes and accurate topology.

In summary, our contributions include the following.

1. We design a high-fidelity 3D hand model for reconstructing 3D hand shapes from single images. The hand representation provides detailed expression, and our frequency decomposition loss helps capture the personalized shape information.
2. To enable computational efficiency, we propose a frequency split network architecture

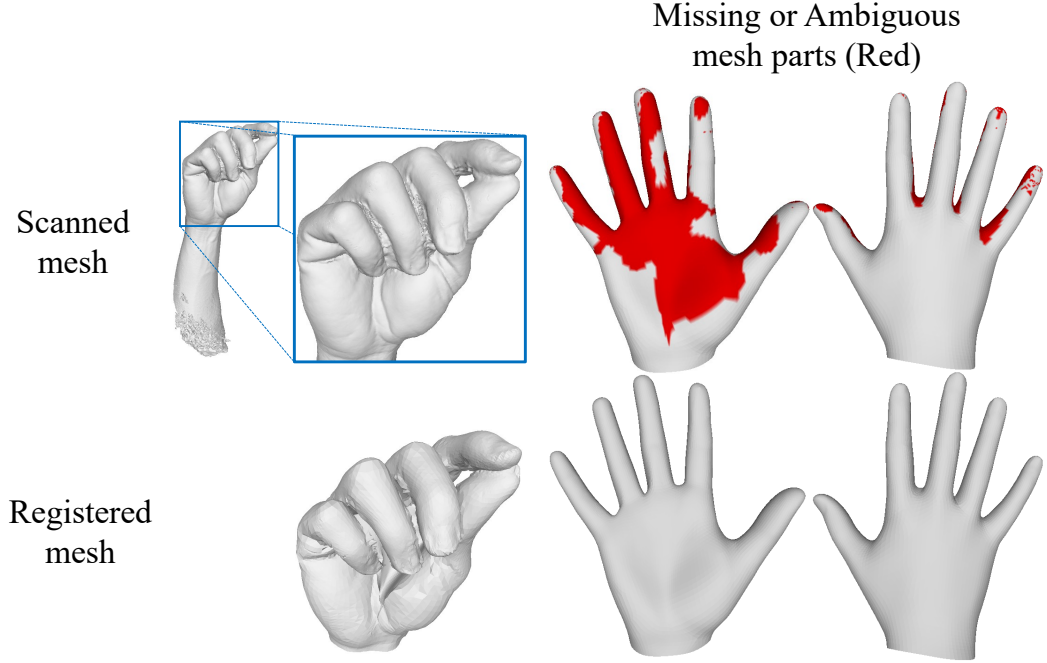


Figure 3.3: An example of the topology error of a scanned hand mesh. The **red** part on the top-right is the missing or ambiguous topology. To better train our network, we use a bidirectional registration strategy to generate valid ground-truth meshes.

to generate high-fidelity hand meshes in a scalable manner with multiple levels of detail. During inference, our scalable framework supports budget-aware mesh reconstruction when the computational resources are limited.

3. We propose a new metric to evaluate 3D mesh details. It better captures the average signal-to-noise ratio of the mesh signal on all frequency bands to evaluate high-fidelity hand meshes. The effectiveness of this metric has been validated by extensive experiments.

The new contributions of our extension include:

1. We design an Image-Graph Ring Frequency Mapping (IGRFM) module to convert image features to graph features through frequency domain analysis. This enables the explicit use of information across various frequency bands from the image for reconstructing high-fidelity hands.
2. We develop a method to register scanned meshes with a parametric model, ensuring

detailed and topologically accurate ground-truth meshes, even with occlusions or topological changes in the scans. This registration would give our network better supervision.

3. Our experiments show the effectiveness of our extended methods. Our visualized and quantitative results are further improved compared to the previous version.

We evaluate our method on the InterHand2.6M dataset [91]. In addition to the proposed evaluation metrics, we also evaluate mean per joint position error (MPJPE) and mesh Chamfer distance (CD). Compared to MANO and other previous methods, our proposed method achieves better results using all three metrics.

2 Related Work

Parametric hand/body shape reconstruction. Parametric models are popular approaches in hand mesh reconstruction. Romero *et al.* [68] proposed MANO, which uses a set of shape and pose parameters to control the movement and deformation of human hands. Many recent works [76, 78, 81, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106] combined deep learning with MANO or its body counterpart SMPL[107]. They use features extracted from the RGB image as input, a CNN to get the shape and pose parameters, and eventually use these parameters to generate the parametric mesh. These methods make use of the strong prior knowledge provided by the hand or body parametric model, so that it is convenient to train the networks and the results are robust. However, the parametric method limits the mesh resolution and details.

Non-parametric hand shape reconstruction. Non-parametric hand shape reconstruction typically estimates the vertex positions of a template with fixed topology. For example, Ge *et al.* [89] proposed a method using a Graph Convolution Network (GCN). It uses a predefined upsampling operation to build a multi-level spectrum GCN. Kulon *et al.* [108] used spatial GCN and spiral convolution operator for mesh generation. Moon *et al.* [109]

proposed a pixel-based approach. However, none of these works paid close attention to detailed shapes. Moon *et al.* [110] provided an approach that outputs fine details, but since they need the 3D scanned meshes of the test cases for training, their model cannot do cross-identity reconstruction. In our paper, we design a new hand model that combines the strength of both parametric and non-parametric approaches. We use this hand model as a basis to reconstruct high-fidelity hands.

Mesh frequency analysis. Previous works mainly focused on the spectrum analysis of the entire mesh graph. Chung [54] defines the graph Fourier transformation and graph Laplacian operator, which builds the foundation of graph spectrum analysis. [111] extends commonly used signal processing operators to graph space. [112] proposes a spectrum graph convolution network based on graph spectrum characteristics. The spectral decomposition of the graph function is used to define graph-based convolution. Recent works such as [113, 114, 115, 116, 117, 118, 119, 120] widely use spectrum GCN in different fields. However, these works mainly focus on the analysis of the overall graph spectrum. In this paper, we use spectrum analysis as a tool to design our provided loss function and metric.

Image-graph feature mapping. Image-graph feature mapping has always been an important problem when using GCN in version tasks. Previous works using GCN to handle mesh reconstruction typically use a simple feature mapping strategy with less consideration of detailed shape information. [121, 122, 70] use pooling on image features, and reassemble the resulting feature to graph vertices, but pooling would cause a loss of high-frequency information, and is thus not suitable for detailed shape reconstruction. [81, 17] use location-related approaches by sampling the feature from the projected location on feature maps. Despite the possibility of preserving details, these approaches are very sensitive to projected location. A small error in the projected location would cause a major local feature change, and is thus not friendly to high-fidelity details. In our proposed method, we use a global frequency mapping strategy which well preserves and maps the high-frequency information of the image features to graph features.

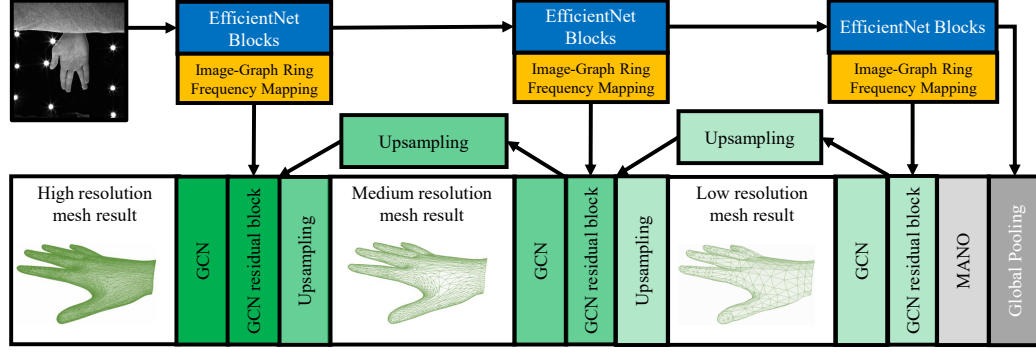


Figure 3.4: We design our scalable hand modeling network in a U-net manner. First, we generate a MANO mesh from image features (light gray block). Then, based on the MANO mesh, we use a multilevel GCN to recover 3 levels of personalized meshes (green blocks from shallow to dark). In order to obtain high-frequency hand details, we use Image-Graph Ring Frequency Mapping (IGRFM) skip-connected image features (yellow blocks) from different layers of the backbone network as parts of the GCN input. At inference, our network can stop at any resolution level, but still provides reasonable high-fidelity results at that resolution.

3 Proposed Method

We propose a scalable network that reconstructs the detailed hand shape, and uses a frequency decomposition loss to acquire details. Fig. 3.4 shows our network architecture. We design our network in the manner of a U-net. First, we generate a MANO mesh from image features from EfficientNet [69]. Based on the MANO mesh, we use a graph convolution network (green blocks in Fig. 3.4) to recover a high-fidelity hand mesh. In order to obtain high-frequency information, we use image features from different layers of the backbone network as a part of the Graph Convolution Network (GCN) inputs. Specifically, at the low-resolution level, we take high-level image features as part of the input, and use a low-resolution graph topology to generate a low-resolution mesh. At medium and high-frequency levels, we use lower-level image features through the skip connection to produce a high-resolution mesh. At each resolution level, we use Image-Graph Ring Frequency Mapping (IGRFM) module to map image features to graph features. Besides, the GCN will output the intermediate hand mesh at every resolution level, so it would naturally have the ability for scalable

inference. During the training process, we supervise both intermediate meshes and the final high-resolution mesh. We discuss the details in the following.

3.1 High-fidelity 3D Hand Model

We design our hand representation based on MANO [68]. MANO factorizes human hands into a 10-dimensional shape representation β and a 35-dimensional pose representation θ . The MANO model can be represented as

$$\begin{cases} M(\theta, \beta) = W(T_P(\theta, \beta), \theta, w) \\ T_P(\theta, \beta) = \bar{T} + B_S(\beta) + B_P(\theta) \end{cases} \quad (3.1)$$

where W is the linear blend skinning function. Model parameter w is the blend weight. Besides, \bar{T} is the original template mesh with n vertices and m edges, B_S and B_P are another two parameters of MANO named shape blend shape and pose blend shape, which are related to pose and shape parameters, respectively. MANO can transfer complex hand surface estimation into a simple regression of a few pose and shape parameters. However, MANO has limited capability in modeling shape detail. It is not only limited by the number of pose and shape dimensions (45) but also by the number of vertices (778). In our work, we designed a new parametric-based model with 12,337 vertices generated from MANO via subdivision. The large number of vertices greatly enhances the model’s ability to represent details.

Subdivided MANO. To address this problem. We design an extended parametric model that can better represent details. First, we add detail residuals to MANO as

$$\begin{aligned} M'(\theta, \beta, d) &= W(T'_P(\theta, \beta, d), \theta, w'), \\ T'_P(\theta, \beta, d) &= \bar{T}' + B'_S(\beta) + B'_P(\theta) + d, \end{aligned} \quad (3.2)$$

where, w' , \bar{T}' , $B'_S(\beta)$, and $B'_P(\theta)$ are the parameters our model, and d is the learnable

per-vertex location perturbation. The dimension of d is the same as the number of vertices.

Besides vertex residuals, we further increase the representation capability of our hand model by increasing the resolution of the mesh. Motivated by the traditional Loop subdivision[123], we propose to design our parametric hand model by subdividing the MANO template. Loop subdivision can be represented as

$$\overline{T}' = \mathcal{L}_s \overline{T}, \quad (3.3)$$

where \overline{T}' is the subdivided template mesh with $n + m$ vertices, where n and m is the number of vertices and edges of original template mesh \overline{T} , and $\mathcal{L}_s \in \mathbb{R}^{(n+m) \times m}$ is the linear transformation that defines the subdivision process. The position of each vertex on the new mesh is only determined by the neighboring vertices on the original mesh, so \mathcal{L}_s is sparse. We use similar strategies to calculate B_S and B_P . The MANO parameters map the input shape and pose into vertex position adjustments. These mappings are linear matrices of dimension $x \times n$. Therefore, we can calculate the parameters as

$$\begin{aligned} w' &= (\mathcal{L}_s w^\top)^\top, \\ B'_S &= (\mathcal{L}_s B_S^\top)^\top, \\ B'_P &= (\mathcal{L}_s B_P^\top)^\top. \end{aligned} \quad (3.4)$$

We repeat the procedure twice to get sufficient resolution.

Fig. 3.5 shows example meshes from the new model in different poses (d is set to 0). We can see that our representation inherits the advantages of the parametric hand model. It has a plausible structure with no visual artifacts when the hand poses change.

3.2 Hierarchical Graph Convolution Network

Our GCN utilizes a multiresolution graph architecture that follows the subdivision process in Section Sec. 3.1. Different from the single graph GCNs in previous works [124, 125],

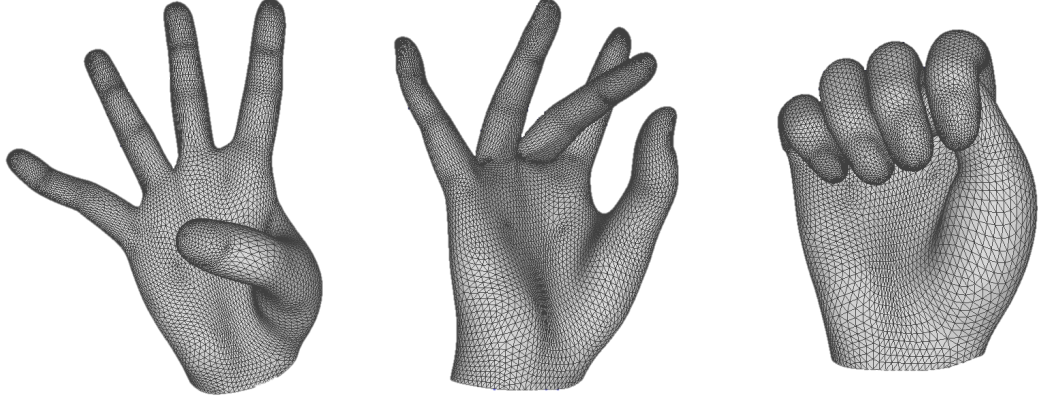


Figure 3.5: We design a new high-fidelity hand mesh with 12,337 vertices. Our new model inherits the advantages of the parametric hand model and provides reliable 3D shape estimation with fewer flaws when hand poses change.

our GCN uses different graphs in different layers. At each level, each vertex of the graph corresponds to a vertex on the mesh, and the graph topology is defined by the mesh edges. Between two adjacent resolution levels, the network uses the \mathcal{L}_s in Eq. (3.3) for upsampling operation.

This architecture is designed for scalable inference. When the computing resources are limited, only the low-resolution mesh needs to be calculated; when the computing resources are sufficient, then we can calculate all the way to the high-resolution mesh. Moreover, this architecture allows us to explicitly supervise the intermediate results, so the details would be added level-by-level.

3.3 Graph Frequency Decomposition

In order to supervise the output mesh in the frequency domain and design the frequency-based metric, we need to do frequency decomposition on mesh shapes. Here, we regard the mesh as an undirected graph, and 3D locations of mesh vertices as signals on the graph. Then, the frequency decomposition of the mesh is the spectrum analysis of this graph signal. Following [54], given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with a vertices set of

$\mathcal{V} = \{1, 2, \dots, N\}$ and a set of edges $\mathcal{E} = \{(i, j)\}_{i, j \in \mathcal{V}}$, the Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \quad (3.5)$$

where \mathbf{A} is the $N \times N$ adjacency matrix with entries defined as edge weights a_{ij} and \mathbf{D} is the diagonal degree matrix. The i th diagonal entry $d_i = \sum_j a_{ij}$. In this paper, the edge weights are defined as

$$a_{ij} := \begin{cases} 1, & (i, j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

which means all edges have the same weights. We decompose \mathbf{L} using spectrum decomposition:

$$\mathbf{L} = U^\top \mathbf{\Lambda} U. \quad (3.7)$$

Here, $\mathbf{\Lambda}$ is a diagonal matrix, in which the diagonal entries are the eigenvalues of \mathbf{L} , and U is the eigenvector set of \mathbf{L} . Since the Laplacian matrix \mathbf{L} describes the fluctuation of the graph signal, its eigenvalues show how "frequent" the fluctuations are in each eigenvector direction. Thus, the eigenvectors of larger eigenvalues are defined as higher frequency bases, and the eigenvectors of smaller eigenvalues are defined as lower frequency bases. Since the column vectors of U form an orthonormal basis of the graph space, following [126], we define the transform

$$F(x) = U^\top x \quad (3.8)$$

to be the Fourier Transform of graph signal, and

$$F'(x) = Ux \quad (3.9)$$

to be the Graph Inverse Fourier Transform (Graph IFT). This means, given any graph function $x \in \mathbb{R}^{N \times d}$, we can decompose x in N different frequency components:

$$x = \sum_{i=1}^N U_i (U_i^\top x), \quad (3.10)$$

where $U_i \in \mathbb{R}^{N \times 1}$ is the i th column vector of U , d is the dimension of the graph signal on each vertex, and $U_i^\top x$ is the frequency component of x on the i th frequency base.

Having Eq. (3.10), we can decompose a hand mesh into frequency components. Fig. 3.1 shows an example of a ground-truth mesh and its frequency decomposition result. The x-axis is the frequencies from low to high. The y-axis is the amplitude of each component in the logarithm. It is easy to observe that the signal amplitude generally decreases as the frequency increases. Fig. 3.6 shows the cumulative frequency components starting from frequency 0. We can see how the mesh shape changes when we gradually add higher frequency signals to the hand mesh. In general, the hand details increase as higher frequency signals are gradually included.

3.4 Image-Graph Ring Frequency Mapping

Image-Graph Ring Frequency Mapping (IGRFM) is a module designed to map image feature maps to graph vertices. As our task is to recover high-fidelity hand mesh via frequency decomposition supervision, this feature transform module is designed to retain information from all frequency bands in the image feature map and transform it into a graph function. As shown in Fig. 3.7, our IGRFM module includes the following steps. (a)→(b): Transform image spatial feature to image frequency feature via image Fast Fourier Transform (FFT). (b)→(c): Segment the image frequency feature using a ring manner. (c)→(d): Aggregate the frequency feature in the rings via ring pooling and use them as the graph frequency features. (d)→(e): Transform the graph frequency feature back into the graph spatial feature on each graph vertex. We will explain the detailed design of each step in the following.

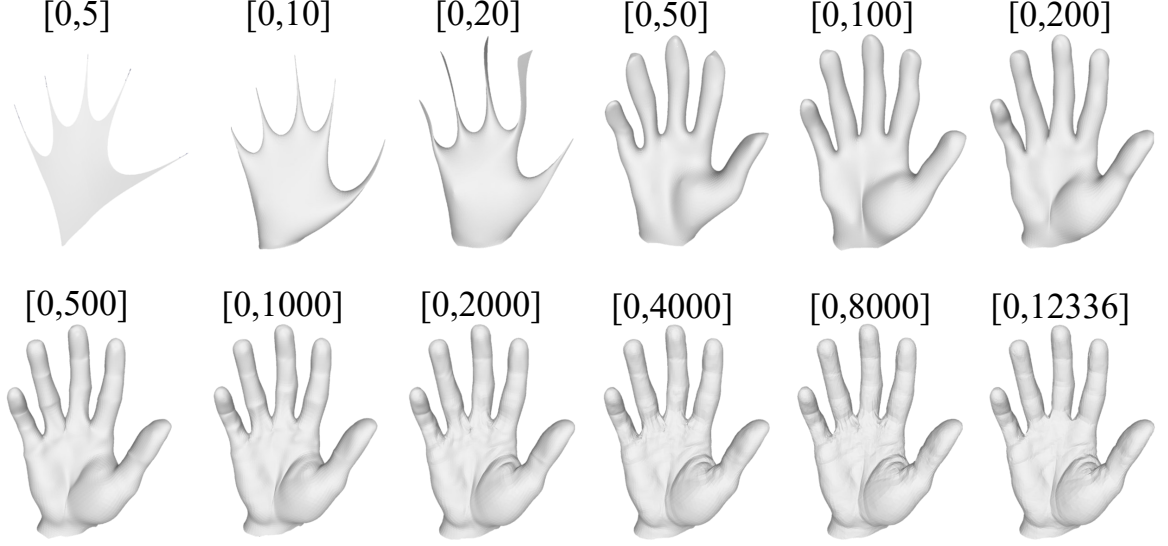


Figure 3.6: Frequency decomposition of a 3D hand mesh. Cumulative frequency components start from frequency 0. The range shows the frequency band. For example, $[0,20]$ means the signal of the first 21 frequencies (lowest 21) added together. We can see how the mesh shape changes when we gradually add higher frequency signals to the hand mesh. In general, the hand details increase as higher frequency signals are included.

Image FFT. To get high-frequency features from the feature map, we first do a Fast Fourier Transform (FFT) on the feature map (Fig. 3.7a). For input feature map I_l of resolution level l ($l = 0, 1$, or 2), we have the feature map in frequency space as

$$\mathcal{F}_{l,c}(\omega_i, \omega_j) = FFT_{\omega_j}(FFT_{\omega_i}(I_{l,c}(i, j))), \quad (3.11)$$

where $I_{l,c}(i, j)$ is the feature map I_l in channel c , (i, j) is the pixel location in the spatial domain, FFT_{ω_j} is FFT along y axis. and FFT_{ω_i} is FFT along x axis. Besides, $\mathcal{F}_{l,c}(\omega_i, \omega_j)$ is the frequency feature map of I_l in channel c , and (ω_i, ω_j) is the pixel location in frequency domain. Fig. 3.7b is the resulting frequency feature map. The inner part represents the lower frequency information and the outer part represents high frequency information.

Ring segmentation. The ring segmentation is designed to segment the image frequency feature along the frequency band so that the features of each frequency can be separated. We segment the image frequency map in a uniform-spaced concentric ring manner. We define a set of evenly spaced concentric ellipses on the image frequency feature map, with

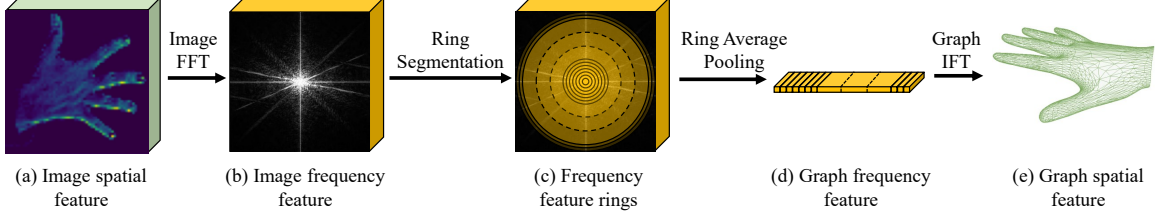


Figure 3.7: We use Image-Graph Ring Frequency Mapping (IGRFM) to map image features to graph features through frequency. (a) Input image spatial feature map. (b) Image frequency feature after Fast Fourier Transform (FFT). (c) Frequency feature rings. The inner rings are low-frequency features and the outer rings are high-frequency features. (d) Graph frequency features after ring average pooling. (e) Graph spatial features transformed from graph frequency features using Graph Inverse Fourier Transform (Graph IFT).

the outermost ellipse being tangent to the edges of the frequency feature map. The centers of those ellipses are defined as the center of the image frequency feature map, and the major and minor axes of the ellipses are paralleled to the edges of the feature map. Since in each channel, the width and height of the image frequency feature map are the same, we let W_l and H_l be the width and height of image frequency feature $\mathcal{F}_{l,c}(\omega_i, \omega_j)$ in level l channel c , and we let $W_l \geq H_l$ for simplicity. Then, the major axis length of the k th innermost ellipse M_k is

$$a_{l,k} = \frac{k}{N+1} W_l, \quad (3.12)$$

where N is the number of vertices of the target graph. Similarly, the minor axis length of the k th innermost ellipse is

$$b_{l,k} = \frac{k}{N+1} H_l. \quad (3.13)$$

The frequency ring $R_{l,k}$ (shown in Fig. 3.7c) is defined as the area between $M_{l,k}$ and $M_{l,k+1}$.

In our network, our image frequency feature map is square, meaning $W_l = H_l$, so that

$$r_{l,k} = a_{l,k} = b_{l,k} = \frac{k}{N+1} A_l, \quad (3.14)$$

where A_l is the edge length of the square image frequency feature in resolution level l , and $R_{l,k}$ is the radius of frequency ring $R_{l,k}$.

Ring average pooling. Having the frequency rings, we design a ring average pooling operation to aggregate the features in each frequency ring. The aggregation results will be used as the graph frequency feature. Inside ring $R_{l,k}$, the ring average pooling is defined as

$$\mathcal{G}_l(k) = \frac{\sum_{\theta=1}^n \mathcal{F}_l\left(\frac{a_{l,k}+a_{l,k+1}}{2} \cos \frac{2\pi}{n}\theta, \frac{b_{l,k}+b_{l,k+1}}{2} \sin \frac{2\pi}{n}\theta\right)}{n}, \quad (3.15)$$

where $a_{l,k}$ and $b_{l,k}$ is defined the same as in Eq. (3.14), θ represents the angle between the line connecting the sample point and the origin, and the positive direction of the X-axis, while n is the number of sample points. Eq. (3.15) means to uniformly sample n points in the middle of ring $R_{l,k}$ and define the average feature of these sample points as the ring's pooling result. In our design, the pooling result will be used as the graph frequency $\mathcal{G}_l(k)$.

In our network where $W_l = H_l$, Eq. (3.15) degenerates to

$$\mathcal{G}_l(k) = \frac{1}{n} \sum_{\theta=1}^n \mathcal{F}_l(\bar{r}_{l,k} \cos \frac{2\pi}{n}\theta, \bar{r}_{l,k} \sin \frac{2\pi}{n}\theta), \quad (3.16)$$

where $\bar{r}_{l,k} = \frac{r_{l,k}+r_{l,k+1}}{2}$.

Graph Inverse Fourier Transform (Graph IFT). We do an Graph IFT on the graph frequency feature $\mathcal{G}_l(k)$ to obtain the per-vertex graph spatial features. Similar to Eq. (3.9), the graph spatial features of resolution l can be calculated as:

$$g_l(v) = U\mathcal{G}_l(k), \quad (3.17)$$

where U is the Graph IFT matrix, which is defined the same as in Eq. (3.9), and v is the graph vertex.

Overall, the transformation from the image spatial feature map I_l to graph spatial feature g_l is defined via Eq. (3.11)-Eq. (3.17).

3.5 Frequency Decomposition Loss

Frequency decomposition loss. Conventional joint and vertex loss, such as the widely used pre-joint error loss [88, 127, 128, 129, 78, 130, 131, 132, 133] and mesh pre-vertex error loss [134, 135, 136, 137] commonly used in human body reconstruction, and Chamfer Distance Loss [138, 139, 140, 141] commonly used in object reconstruction and 3D point cloud estimation, all measure the error in the spatial domain. In that case, the signals of different frequency components are aliased together. As shown in Fig. 3.1, the amplitudes of low-frequency signals of hand shape are much larger than high-frequency signals, so when alias happens, the high-frequency signals will be overwhelmed, which means direct supervision in the spatial domain would mainly focus on low-frequency signals. Thus, spatial loss mostly does not drive the network to generate high-frequency details. Our experiments in Sec. 5.3 also demonstrate this.

To generate detailed information without being overwhelmed by low-frequency signals, we design a loss function in the frequency domain. Specifically, we use graph frequency decomposition (Sec. 3.3) to define our frequency decomposition loss as

$$\mathcal{L}_F = \frac{1}{F} \sum_{f=1}^F \log \left(\frac{\|U_f^\top \hat{V} - U_f^\top V\|^2}{\|U_f^\top \hat{V}\| \|U_f^\top V\| + \epsilon} + 1 \right), \quad (3.18)$$

where $F = N$ is the number of total frequency components, U_f is the f th frequency base, $\|\cdot\|$ is l_2 norm, $\epsilon = 1 \times 10^{-8}$ is a small number to avoid division-by-zero, $\hat{V} \in \mathbb{R}^{N \times 3}$ and $V \in \mathbb{R}^{N \times 3}$ are the predicted and ground-truth vertex locations, respectively. During training, for every frequency component, our loss reduces the influence of the amplitude of each frequency component, so that information from different frequency components would have equivalent attention. In Tab. 3.3, we demonstrate the effectiveness of the frequency decomposition loss.

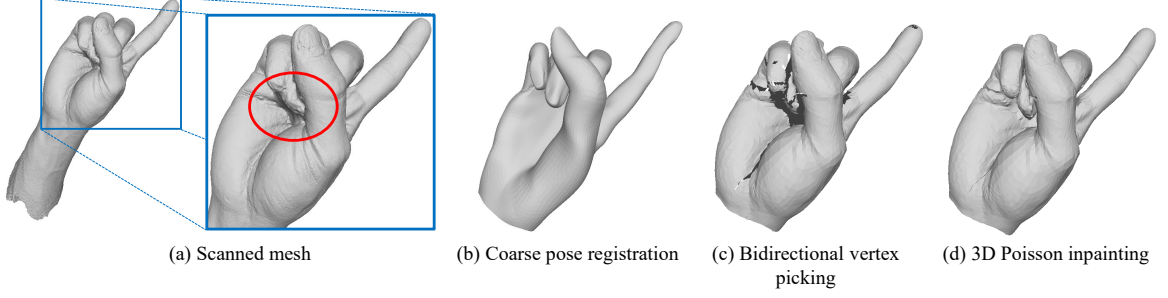


Figure 3.8: Example of Topology-correct hand mesh registration. (a) Scanned mesh with topology flaws (red circle). (b) We use an optimization-based coarse pose registration to get the coarse pose. (c) We then use bidirectional vertex picking to get the topology-correct part of mesh vertices. (d) We finally use 3D Poisson editing to inpaint the topology-incorrect part of the mesh vertices.

Total loss function. We define the total loss function as:

$$\mathcal{L} = \lambda_J \mathcal{L}_J + \sum_{l=1}^3 \left[\lambda_v^{(l)} \mathcal{L}_v^{(l)} + \lambda_F^{(l)} \mathcal{L}_F^{(l)} \right], \quad (3.19)$$

where l is the resolution level, while $l = 1$ is the lowest-resolution level and $l = 3$ is the highest-resolution level. Besides, $\mathcal{L}_J^{(l)}$ is 3D joint location error, $\mathcal{L}_v^{(l)}$ is per-vertex error, $\mathcal{L}_F^{(l)}$ is the frequency decomposition loss, $\lambda_J^{(l)}$, $\lambda_v^{(l)}$, and $\lambda_F^{(l)}$ are hyper-parameters. For simplicity, we refer to $\mathcal{L}_J^{(l)}$, $\mathcal{L}_v^{(l)}$, and $\mathcal{L}_F^{(l)}$ as \mathcal{L}_J , \mathcal{L}_v , and \mathcal{L}_F for the rest of the paper.

Following previous work [142, 136], we define 3D joint location error and per-vertex loss as:

$$\begin{aligned} \mathcal{L}_J &= \frac{1}{N_J} \sum_{j=1}^{N_J} \|\hat{J}_j - J_j\|, \\ \mathcal{L}_v &= \frac{1}{N} \sum_{i=1}^N \|\hat{v}_i - v_i\|, \end{aligned} \quad (3.20)$$

where \hat{J}_j and J_j are the output joint location and ground-truth joint location, and N_J is the number of joints. Besides, \hat{v}_i and v_i are the estimated and ground-truth locations of the i th vertex, and N is the number of vertices.

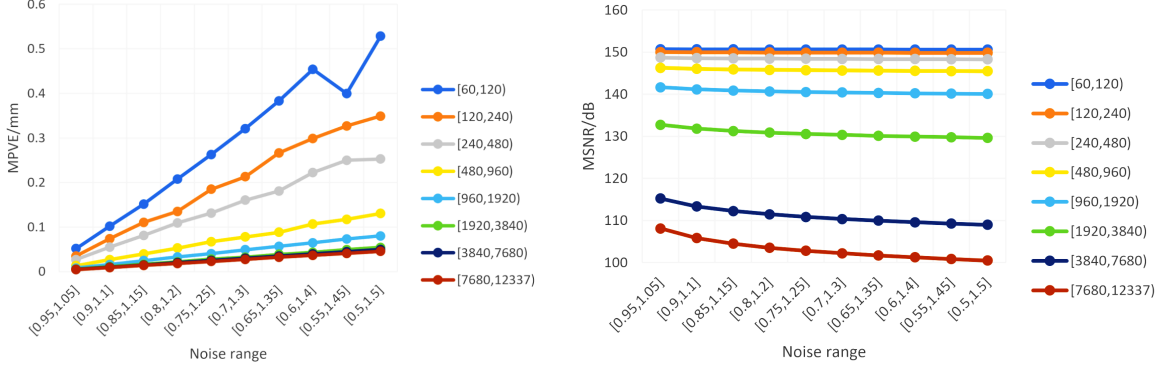


Figure 3.9: Evaluations using Euclidean distance and MSNR under different noise amplitudes in every frequency band. Each line of a different color indicates a frequency band. The maximum and minimum frequencies are shown in the legend. On each line, every dot means adding a random amplitude noise to the mesh. The noise amplitude of each dot is evenly distributed over the ranges shown on the x-axis. The result validates that Euclidean distance is more sensitive to error in low-frequency bands, and MSNR is more sensitive to noise in high-frequency bands. Thus, compared to Euclidean distance, MSNR can better measure the errors in high-frequency details.

4 Datasets and Annotation Generation

4.1 High-fidelity Hand Dataset

Our task requires detailed hand meshes for supervisory purposes. Given the challenges and high costs associated with acquiring 3D scan data, obtaining such supervision on a large scale is problematic. To address this, we have devised an alternative approach: creating meshes from multiview RGB images through multiview stereo techniques. Due to the relatively easy access to these resources, we have decided to adopt this method and utilize the meshes generated in this manner as the ground-truth for our experimental work. We do all our experiments on the InterHand2.6M dataset [91], which is a dataset consisting of multiview images, rich poses, and human hand pose annotations. The dataset typically provides 40-100 views for every frame of a hand video. Such a large amount of multiview information would help with more accurate mesh annotation. To provide detailed hand mesh supervision for training, a multiview stereo method is used to generate the scanned mesh. In this paper, we use the mesh results provided in [110], which are generated using

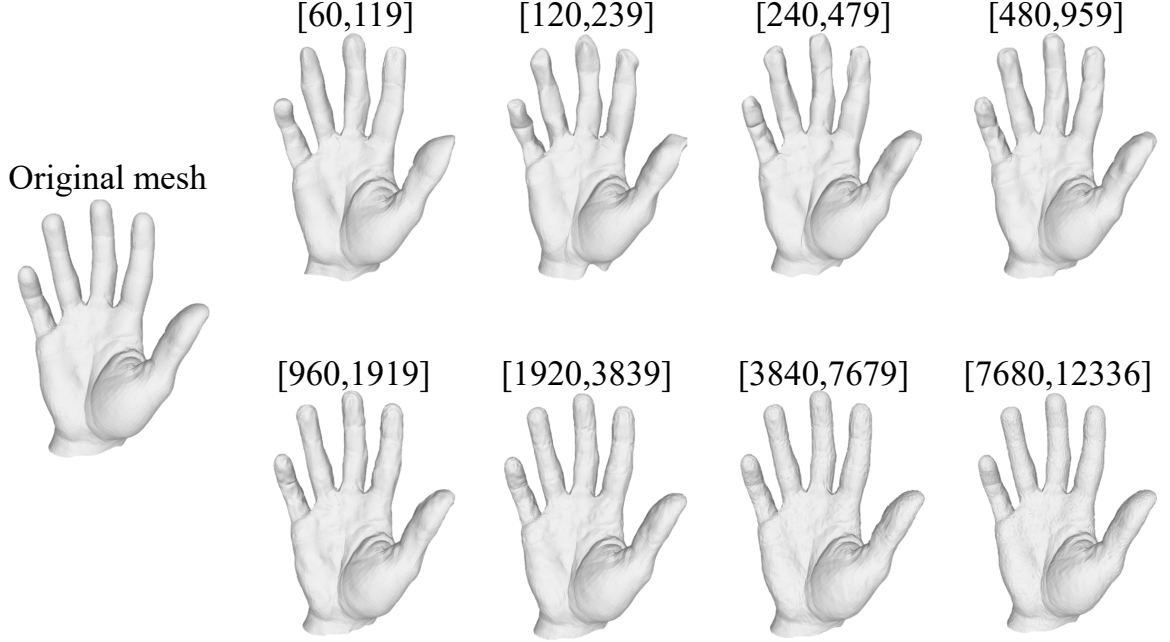


Figure 3.10: We show examples of Noisy Meshes. The meshes from left to right are meshes with a noise maximum amplitude of 0.6 and the frequency band changed from $[60,119]$ to $[7680,12336]$. For easier visualization, we magnify the vertices location changes by a factor of 5.

the multiview methods of [143], and only use a subset of InterHand2.6M, due to the large volume of data in the original dataset. We show one example of the scanned mesh in Fig. 3.8a. The hand details are clearly shown on the mesh surface. Those details can well support our high-fidelity hand training, and the generation of those detailed high-fidelity hand ground-truth is also budget-friendly.

4.2 Topology-correct Annotation Generation

Although the mesh generated using multiview stereo methods has good details, those methods do not guarantee a complete hand topological structure. To generate mesh supervision with accurate details and a correct topological structure, we designed a mesh registration method that uses the subdivided MANO topology to register a scanned mesh. As shown in Fig. 3.8, the registration is divided into the following steps: (1) Optimization-based coarse registration. We first register the overall translation, rotation, and subdivided MANO pose

and shape parameters of the hand through an optimization method. (2) Bidirectional vertex picking. Through bidirectional matching, we find a suitable vertex on the scanned mesh for each coarse-registered vertex, and use that location as the final registration. (3) 3D Poisson inpainting. For coarse-registered vertices that do not find a suitable match, we inpaint them using Poisson mesh editing. We use the registration technique across 3 levels of hand mesh templates, so that we can provide mesh supervision for all 3 levels of our network. The details of the registration design will be discussed in the following.

Optimization-based coarse pose registration. Before registering the detailed shapes and topology, it is crucial to align the hand pose of the subdivided MANO model with the scanned mesh. This alignment ensures that the resulting hand pose matches the scanned hand pose, and the further step would only need to consider detailed shapes. Here, we implemented a two-stage optimization process, utilizing the joint location ground-truth as annotations. Initially, we set the global translation of the hand, denoted as T , to the wrist joint’s ground-truth location. This initial step simplifies the optimization process, so it can converge more effectively. We initialize both MANO pose parameters θ and shape parameters β as $\vec{0}$. Then, we jointly optimize θ and T as

$$(\theta_1, T_1) = \arg \min_{\theta, T} \|\mathcal{J}(\mathcal{M}(\theta, \beta) + T) - J\|_f, \quad (3.21)$$

where \mathcal{M} is MANO hand model, \mathcal{J} is MANO hand regressor, J is the ground-truth joint locations, and $\|\cdot\|_f$ represent Frobenius norm. We use the joint location as a guideline to jointly optimize the global translation and the MANO pose parameters.

Besides global translation and pose parameters, MANO shape parameters also influence the length of the palm and fingers, and consequently influence the joint locations. Here, we initialize T , θ , and β as T_1 , θ_1 , and $\vec{0}$, respectively, so that the optimization would be easier to converge and less likely to fall into local minimums. Then, we jointly optimize β , θ , and

T as

$$(\beta_2, \theta_2, T_2) = \arg \min_{\beta, \theta, T} \|\mathcal{J}(\mathcal{M}(\theta, \beta) + T) - J\|_f. \quad (3.22)$$

Here, β_2 , θ_2 , and T_2 define the subdivided MANO coarse registration of the scanned mesh.

In Fig. 3.8b we show an example of the coarse registration result.

Bidirectional vertex picking. Having the correct hand pose, for each vertex p_i on the parametric hand M_p , we try to find a suitable vertex s_j on the scanned mesh M_s , and use its location v_{s_j} as the location of p_i , *a.k.a.* $v_{p_i} \leftarrow v_{s_j}$.

We find the suitable vertex s_j for p_i using a bidirectional manner. First, we divide the vertices in scanned mesh M_s into $|\{p_i\}|$ sets. The i th set is defined as

$$C_i = \{s_j | i = \arg \min_k \|v_{s_j} - v_{p_k}\|_2\}, \quad (3.23)$$

which is the set of s_j whose closest vertex in M_p is p_i . Considering the scanned mesh has a much larger vertex number than the parametric mesh (500k vertices vs. no more than 12.4k vertices), each vertex in M_s does not need to be covered by more than 1 vertex in M_p , but each vertex on M_p may cover one or multiple vertices on M_s . Besides, the scanned mesh may have topology lost, which means there may be vertices on M_p that cover 0 vertex on M_s . Under that assumption, if we use distance as the metric to determine the covering relations, Eq. (3.23) is to find the set that p_i covers in M_s . Here $|C_i| \geq 0$. When $|C_i| = 0$, it means there are no vertices on M_s that should be covered by p_i .

Having the covering set C_i , we can pick one vertex from the set for each p_i , and register p_i 's location to that vertex. Since C_i is the set in which every vertex is covered by p_i , picking any vertex in C_i for p_i to register would be reasonable. In practice, we consider there may be outlying vertices in the scanned mesh M_s , which means these vertices are far from any vertex on M_p , and no vertex on M_p should cover them. Thus, to minimize the influence of outliers and enhance robustness, we register p_i to be the vertex in C_i that is closest to the

original location of p_i as

$$v'_{p_i} = \min_{s_j \in C_i} \|v_{s_j} - v_{p_i}\|_2, \text{ when } C_i \neq \emptyset \quad (3.24)$$

where v'_{p_i} is the registered vertex location of p_i . In Fig. 3.8c, we show an example of the registered vertices $\{p_i\}$ on M_p when $C_i \neq \emptyset$.

3D Poisson inpainting. So far we registered the vertices $\{p_i\}$ on M_p when $C_i \neq \emptyset$. However, when $C_i = \emptyset$, the scanned mesh does not provide enough detailed information for those vertices. To generate a complete hand mesh, we inpaint those vertices using the local shape information from the coarse registered mesh. Here, we use a Poisson 3D mesh editing approach [144] to achieve this. Specifically, we let $\Omega = \{p_i | C_i = \emptyset\}$ to be the vertex set on M_p that covers 0 vertices on M_s , and $\partial\Omega = \bigcup_k \{p_k | p_k \in \mathcal{N}(p_i), C_i = \emptyset, p_k \notin \Omega\}$ to be the nearest neighbor vertices set of Ω on M_p , where $\mathcal{N}(p_i)$ is the nearest neighbor vertices set of p_i . We define the Poisson Equation as

$$\begin{cases} L\hat{\mathbf{v}}_p = L\mathbf{v}_p, & p_i \in \Omega, \\ \hat{\mathbf{v}}_p = \mathbf{v}'_p, & p_i \in \partial\Omega, \end{cases} \quad (3.25)$$

where $\hat{\mathbf{v}}_p$ is the registered vertex location in matrix form we try to solve, and \mathbf{v}_p and \mathbf{v}'_p are the coarse-registered vertices and the bidirectional-registered vertices in matrix form, respectively. Besides, L is the Laplacian matrix of the mesh graph same as in Eq. (3.5). Eq. (3.25) means the local details in Ω should follow the coarse registered mesh while the edge of Ω should be the same with bidirectional registered mesh. We rewrite Eq. (3.25) in matrix form as

$$\begin{bmatrix} L \\ I \end{bmatrix} \hat{\mathbf{v}}_p = \begin{bmatrix} L\mathbf{v}_p \\ \mathbf{v}'_p \end{bmatrix}, \quad (3.26)$$

Table 3.1: Joint and mesh errors (Chamfer distance) of topology-correct mesh annotations of 3 resolution levels on InterHand2.6M. For joint error and Chamfer distance, lower is better.

Annotation Level	Joint error/mm↓	Chamfer distance/mm↓
1	5.5	0.52
2	5.5	0.57
3	5.5	0.61

where I is an identity matrix. Then, we can solve $\hat{\mathbf{v}}_p$ using

$$\hat{\mathbf{v}}_p = \left(\begin{bmatrix} L \\ I \end{bmatrix}^\top \begin{bmatrix} L \\ I \end{bmatrix} \right)^{-1} \begin{bmatrix} L \\ I \end{bmatrix}^\top \begin{bmatrix} L\mathbf{v}_p \\ \mathbf{v}'_p \end{bmatrix}. \quad (3.27)$$

Thus, we have

$$v'_{p_i} = \hat{\mathbf{v}}_{p,i}, \text{ when } C_i = \emptyset, \quad (3.28)$$

An example of the completely registered mesh is shown in Fig. 3.8d. In this way, we generate a topology correct hand mesh ground-truth with shape details. In Tab. 3.1, we analyze the joint and vertex errors of our registration in 3 resolution levels.

5 Experiments

5.1 Implementation Details

We follow the network architecture in [142] to generate intermediate MANO results. We use EfficientNet [69] as a backbone. The low-level, mid-level, and high-level features are extracted after the 1st, 3rd, and 7th blocks of EfficientNet, respectively. For each image feature, we use 1×1 convolutions to deduce dimensions. The channel numbers of 1×1 convolution are 32, 32, and 64 for low-, mid-, and high-level networks, respectively. After that, we project the initial human hand vertices to the feature maps, and sample a feature vector for every vertex using bilinear interpolation. The Graph Convolution Network (GCN) graph has 778, 3093, and 12,337 vertices at each resolution level. At each level, the input

Table 3.2: Module effectiveness Results on the InterHand2.6M [91] dataset. **Bold** number means the best. For MPJPE and Chamfer distance (CD), lower is better. For MSNR, higher is better. The proposed method improves the accuracy of hand surface details compared to previous methods and our conference version (Conf-level 1-3). While our method generates better shape details in a scalable manner, the general accuracy (MPJPE and CD) of overall shape also increases.

Method	MPJPE/mm↓	CD/mm↓	MSNR↑
MANO	13.41	6.20	-2.64
DIGIT [145]	13.36	6.32	-2.72
Conf-level 1 [17]	13.25	5.53	-2.70
Conf-level 2 [17]	13.25	5.49	-2.62
Conf-level 3 [17]	13.25	5.49	-0.68
Ours-level 1	13.09	5.06	-2.57
Ours-level 2	13.09	5.00	-2.19
Ours-level 3	13.09	5.00	-0.28

features go through a 10-layer GCN Residual Block, which outputs a feature vector and a 3D location at each vertex.

In the training process, we first train [142] network, and then use the pre-trained result to train our scalable network. For training [142], we use their default hyper-parameters, set the learning rate to 1×10^{-4} , and set batch size to 48. When training GCN, we set λ_J to be 5, set $\lambda_v^{(1)}$, $\lambda_v^{(2)}$, and $\lambda_v^{(3)}$ to be 1, and set $\lambda_F^{(1)}$, $\lambda_F^{(2)}$, and $\lambda_F^{(3)}$ to be 5. The batch size is set to 24. The learning rate is set to 1×10^{-4} . After some code revision, the training process takes about 6 hours on 1 NVIDIA GTX3090 GPU for 100 epochs. In reference, we use a smooth kernel to post-process the mesh to reduce the higher-frequency noise in resolution level 3.

5.2 Quantitative Evaluation

We use mean per joint position error (MPJPE) and Chamfer distance (CD) to evaluate the hand pose and coarse shape. Besides, to better evaluate personalized details, we also evaluate our mesh results using the proposed Mean-frequency Signal-to-Noise Ratio (MSNR) metric.

Mean-frequency Signal-to-Noise Ratio (MSNR). Previous metrics for 3D hand mesh mostly calculate the Euclidean distance between the results and the ground-truth. Although

Table 3.3: Ablation study on the feature skip connection design, new registration strategy, and the effect of loss functions. **Bold** number means the best. The 2nd-4th lines show the effectiveness of our Image-Graph Ring Frequency Mapping (IGRFM). The 5th line shows the result of using the previous registration from the conference version. The 6th and 7th lines show the effectiveness of our loss functions.

Method	MPJPE/mm↓	CD/mm↓	MSNR↑
proposed	13.09	5.00	-0.28
w/o skip-connected feature	14.20	5.85	-0.52
w/ average pooling feature	13.39	5.23	-0.40
w/ projection feature	13.12	5.04	-0.34
w/o coarse registered ground-truth	13.25	5.37	-0.62
w/o frequency decomposition loss	14.47	5.47	-0.94
w/o per-vertex error loss	14.12	43.0	-0.51

Table 3.4: Quantitative results of removing high-frequency features in IGRFM. We remove the high-frequency features and retain 10% and 1% of the lowest-frequency features in IGRFM frequency rings, and show 3 levels of the quantitative results comparison.

Frequency Bands	Level 1			Level 2			Level 3		
	MPJPE/mm↓	CD/mm↓	MSNR↑	MPJPE/mm↓	CD/mm↓	MSNR↑	MPJPE/mm↓	CD/mm↓	MSNR↑
Whole band	13.09	5.06	-2.57	13.09	5.00	-2.19	13.09	5.00	-0.28
10% low frequency	13.10	5.09	-2.58	13.10	5.01	-2.21	13.10	5.00	-0.31
1% low frequency	13.25	5.17	-2.57	13.25	5.07	-2.21	13.25	5.04	-0.32

in most cases, Euclidean distance can roughly indicate the accuracy of the reconstruction results, it is not consistent with human cognitive standards: it is more sensitive to low-frequency errors, but does not perform well in personalized detail distinction or detailed shape similarity description.

Thus, we propose a metric that calculates the signal-to-noise ratio in every frequency basis of the graph. We define our Mean-frequency Signal-to-Noise Ratio (MSNR) metric as

$$\text{MSNR} = \frac{1}{F} \sum_{f=1}^F \log\left(\frac{\|U_f^\top \hat{V}\|}{\|U_f^\top \hat{V} - U_f^\top V\| + \epsilon}\right), \quad (3.29)$$

where $F = N$ is the total number of frequency components and S_f is the signal-to-noise ratio of the f th frequency component. Besides, U_f , \hat{V} , and V are defined the same as in Eq. (3.18), and $\epsilon = 1 \times 10^{-8}$ is a small number to avoid division-by-zero. Thus, the maximum of S_f is 8. By this design, the SNR of different frequency components would not

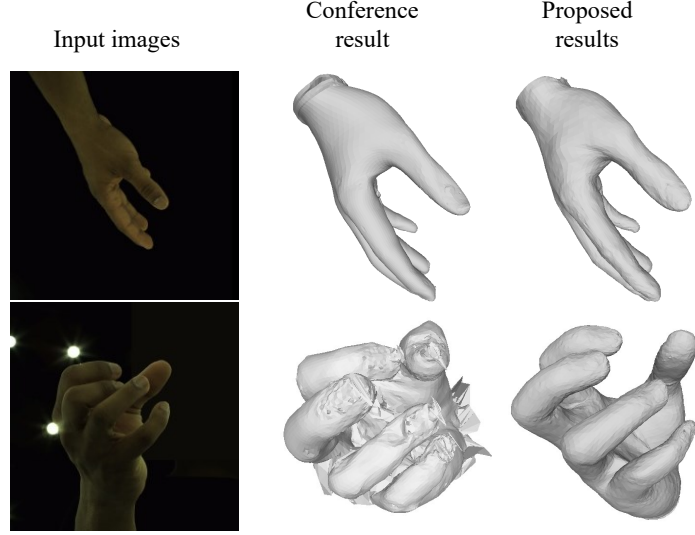


Figure 3.11: Visualized comparison with the conference version. We compare our results with those of our conference version. Our results have better high-fidelity details (first row). Moreover, our proposed method can solve some failure cases of the previous conference version (second row).

influence each other, so we can better evaluate the high-frequency information compared to the conventional Euclidean Distance.

We designed an experiment on InterHand2.6M to validate the effectiveness of our metric in evaluating high-frequency details. We add errors of 8 different frequency bands to the hand mesh. For each frequency band, the error amplitude drawn from 10 different uniform distributions. As shown in Fig. 3.9, we measure the MPVE and MSNR for every noise distribution in every frequency band, to see how the measured results of the two metrics change with the noise amplitude in each frequency band. The result shows that in the low-frequency part, MPVE increases fast when the noise amplitude increases (the upper lines), but in high-frequency bands, the measured result changes very slowly when the noise amplitude increases. MSNR behaves completely differently from MPVE. It is more sensitive to noise in the high-frequency band than in the low-frequency band. Thus, compared to Euclidean distance, MSNR better measures the error in high-frequency details. Fig. 3.10 shows a few examples of noisy meshes.

Evaluation on InterHand2.6M dataset. We report the mean per joint position error

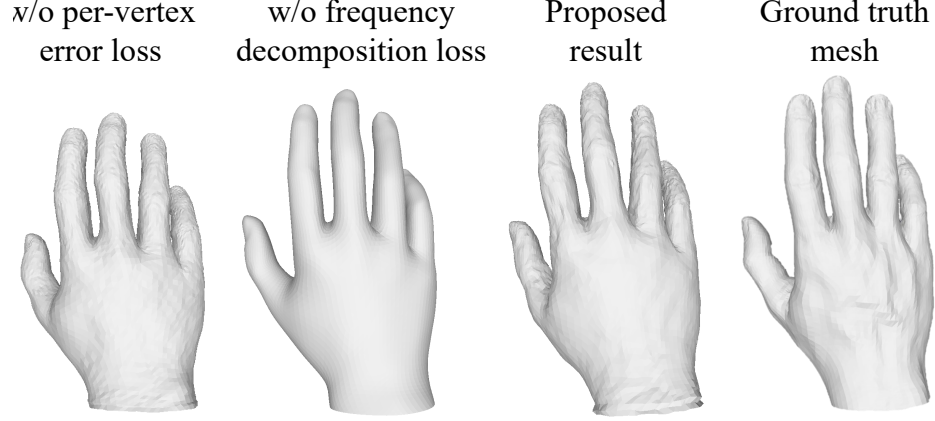


Figure 3.12: Visualization results of “w/o frequency decomposition loss” and “w/o per-vertex error loss” in Sec. 5.3. As shown, if we do not use frequency decomposition loss, the mesh result we get tends to be smoother with fewer personalized details. If we do not use the per-vertex error loss, the mesh’s low-frequency information is not well learned. The mesh we generate exhibits overall shape deformation.

(MPJPE), Chamfer distance (CD), and Mean-frequency Signal-to-Noise Ratio (MSNR) to evaluate the overall accuracy of the reconstructed hand meshes. Tab. 3.2 shows the comparison of 3 resolution levels of our proposed method with previous methods and our conference version. We observe that our proposed method outperforms previous mesh reconstruction methods and our conference version. Moreover, while our method generates better shape details in a scalable manner, the accuracy of our joint locations and the overall shape of the output meshes also slightly increased (as indicated by MPJPE and CD). Here, the MSNR is calculated after subdividing the meshes to the level 3 resolution.

In Fig. 3.11 we show some results compared with our conference version. We can see our details are better than those of the conference version, Additionally, we can also solve some failure cases of the conference version. The last column contains our current results.

5.3 Ablation Studies

Module effectiveness. We conduct several experiments to demonstrate the effectiveness of different modules, our new data registration, and different loss functions. The results are shown in Tab. 3.3. The 2nd-4th lines show the effectiveness of our Image-Graph

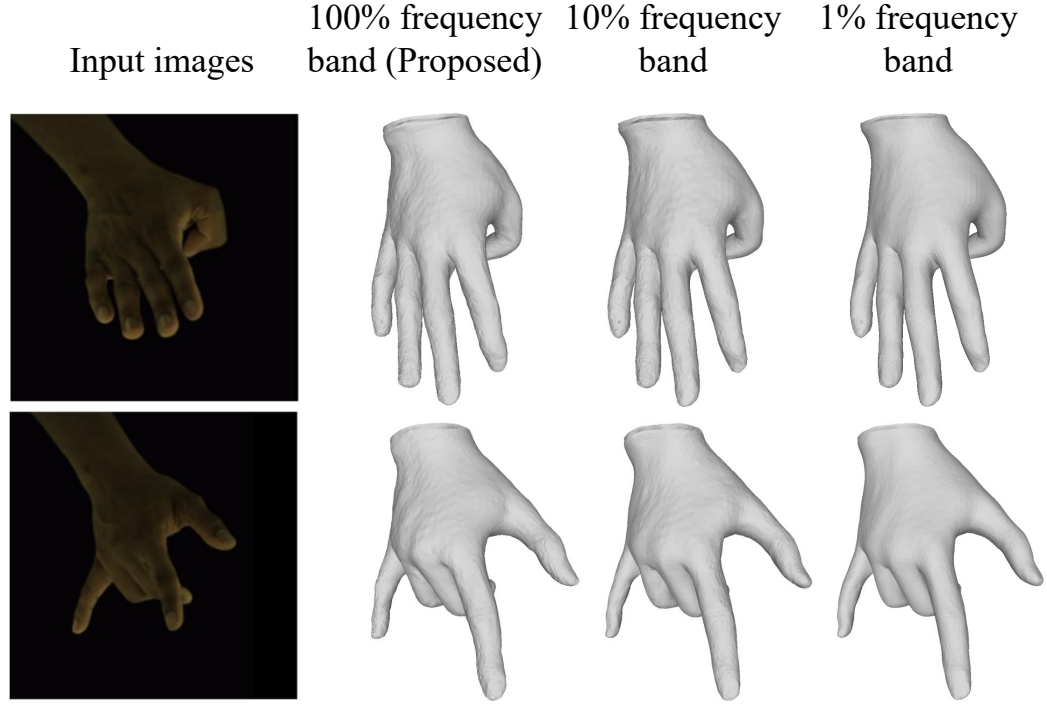


Figure 3.13: Visualized results of removing high-frequency features in IGRFM. (Best viewed at magnification.) We remove the high-frequency feature rings and retain 10% and 1% of the lowest-frequency features in IGRFM frequency rings, and show the highest-resolution visualized results comparison. As shown in the figure, removing the high-frequency feature rings will cause a loss of the shape details.

Ring Frequency Mapping (IGRFM). Our feature mapping design outperforms the 3 other feature mappings including the “no skip-connected feature”, “average pooling feature”, and “projection feature”. From thees results, we observe that our projection-to-feature-map skip connection design leads to performance improvements in all three metrics. The 5th line shows the result of using the previous registration used in the conference version [17]. Our result outperforms the previous registration strategy. For the loss functions (in the 6th and 7th lines), we observe that MSNR degrades when the frequency decomposition loss is removed, indicating inferior mesh details. Removing the per-vertex error loss dramatically increases the Chamfer distance, indicating that the overall shape is not well constrained. The visualization results of the last 2 experiments are shown in Fig. 3.12. Using our new registered mesh annotation for training can avoid some mesh flaws in the topology error areas. If we do not use the frequency decomposition loss, the mesh result we get tends to be

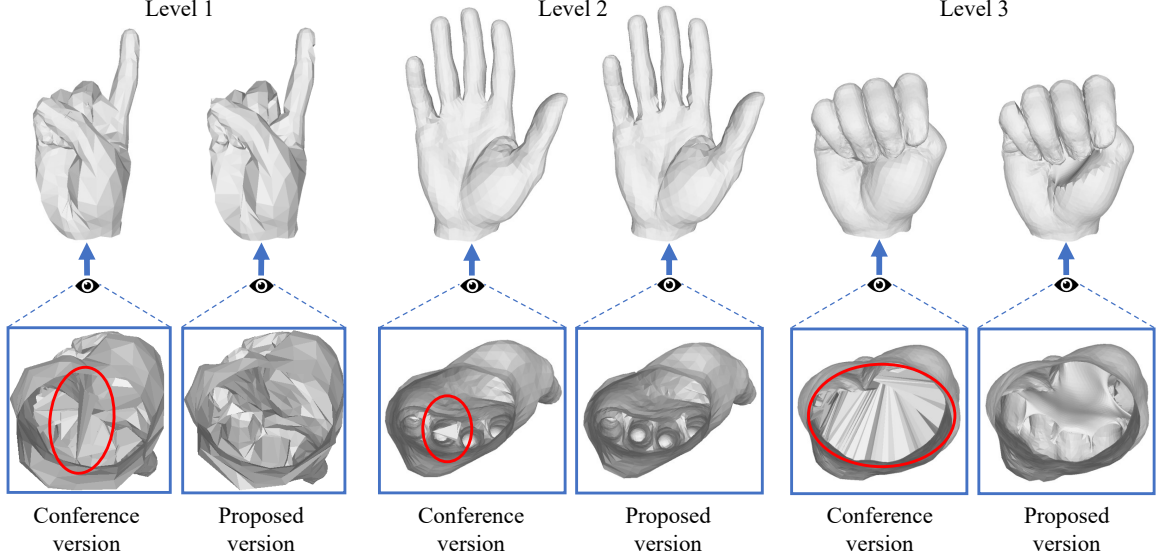


Figure 3.14: Comparison of registered mesh annotations. For each case, the left meshes are our conference version results, and the meshes on the right are our proposed registration method results. We can observe some topology flaws in the conference version results (abnormal triangle faces in the red circles), while our proposed registration does not have such flaws.

smoother with fewer personalized details. If we do not use per-vertex error loss, the mesh’s low-frequency information is not well-learned. The mesh we generate will have an overall shape deformation.

Removing high-frequency features in IGRFM. We further elaborate on the effectiveness of high-frequency feature rings in the IGRFM module. We remove the high-frequency features and retain 10% and 1% of the lowest-frequency feature rings before ring average pooling, and show the quantitative and qualitative results of the resulting high-fidelity hands in Tab. 3.4 and Fig. 3.13. We can see that with the removal of high-frequency image features, the quantitative performance drops, and the details of the high-fidelity hands disappear.

IGRFM ring segmentation strategies. We also tried different ring segmentation strategies for IFRFM. In our proposed ring segmentation, we keep the radius difference between adjacent rings to be the same. In Tab. 3.5, we change the segmentation strategy by keeping the same area difference between adjacent rings (“Area”), keeping the radius difference to be the same as the graph frequency Λ in Eq. (3.7) (“Graph frequency”),

Table 3.5: Comparison of different IGRFM ring segmentation strategies. From top to bottom: Radius(proposed method): Keeping the radius difference between adjacent rings to be the same. Area: Keeping the same area difference between adjacent rings. Graph frequency: Keeping the radius difference to be the same as the graph frequency Λ in Eq. (3.7). Square root of graph frequency: Keeping the radius difference to be the same as the square root of graph frequency. Random segmentation: Randomly segment the image frequency band. Our results show that our proposed radius segmentation has the best MSNR result.

Method	MPJPE/mm↓	CD/mm↓	MSNR↑
Radius (proposed)	13.09	5.00	-0.28
Area	13.40	5.30	-0.34
Graph frequency	13.19	5.04	-0.33
Square root of graph frequency	13.23	5.00	-0.36
Random segmentation	13.11	5.00	-0.31

Table 3.6: The mesh sizes and the resources needed for generating different resolution levels of meshes for both our proposed method and conference version. We observe that despite our performance exceeding that of our conference version, the computational costs barely increase.

Methods	#parameter	GFLOPS	#vertices	#faces
Baseline	14.5M	1.81	778	1538
Conf-level 1 [17]	14.5M	1.87	778	1538
Conf-level 2 [17]	14.5M	2.54	3093	6152
Conf-level 3 [17]	14.7M	4.81	12337	24608
Ours-level 1	14.5M	1.87	778	1538
Ours-level 2	14.5M	2.54	3093	6152
Ours-level 3	14.7M	4.83	12337	24608

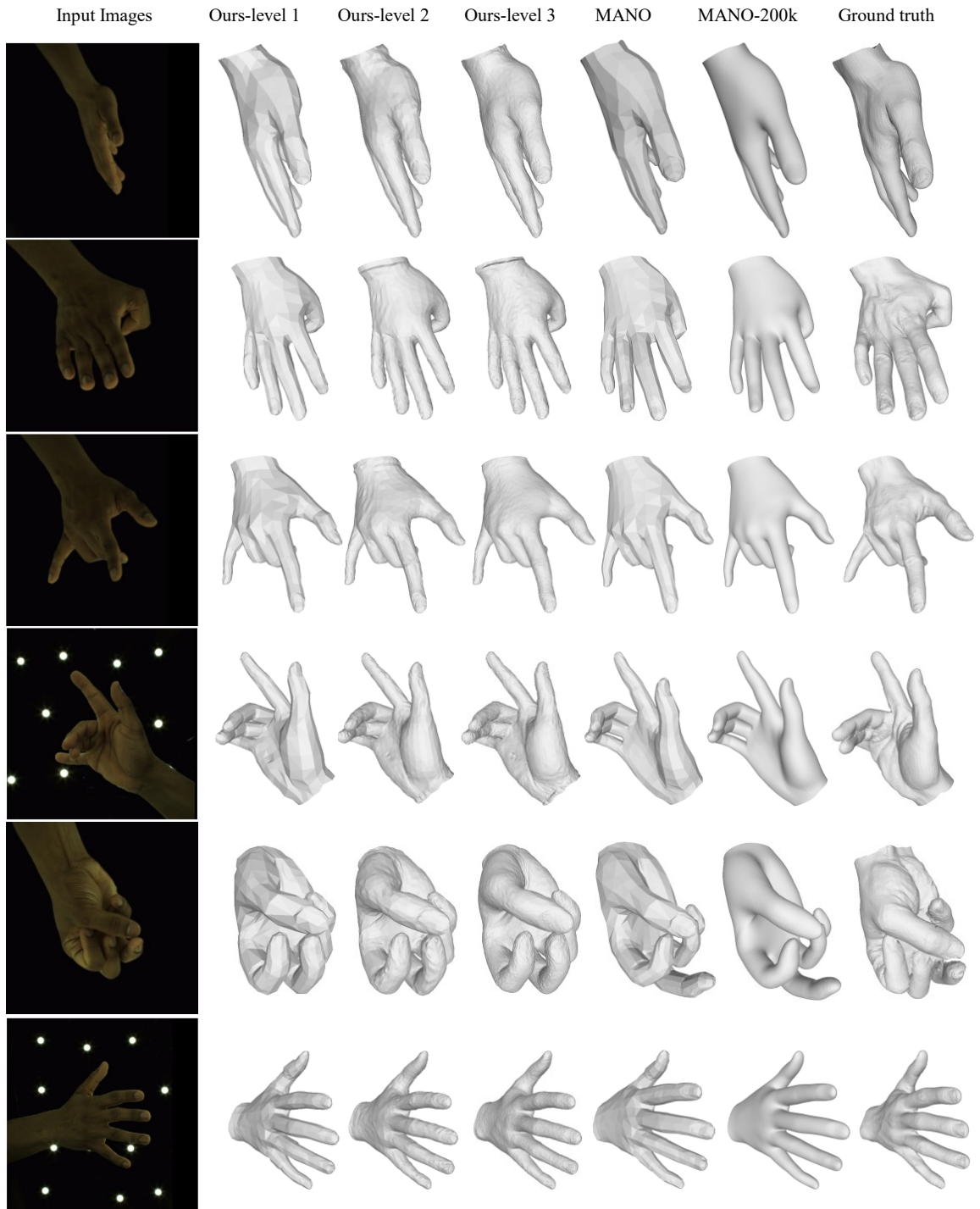


Figure 3.15: Qualitative reconstruction results. The columns, from left to right, are input images, our level 1-3 output meshes, MANO mesh, MANO mesh subdivided to 200k vertices (*i.e.* the same number of vertices as our mesh), and the ground-truth, respectively. We can see that even if we upsample MANO into the same number of vertices as our mesh, it still does not provide personalized details comparable to our results.

keeping the radius difference to be the same as the square root of graph frequency (“Square root of graph frequency”), or randomly segment the image frequency band (“Random segmentation”). Our results show that our proposed radius segmentation has the best MSNR result.

Comparison of registered mesh annotations. We show in Fig. 3.14 a visualized comparison of the registered mesh annotations using our proposed registration method with those in the conference version. We can observe that our registered annotation does not have the topology flaws that occur in the conference version.

Scalable design. We also demonstrate the scalable design of the proposed network by analyzing the resources needed at each resolution level, and comparing that of our proposed method with our conference version in Tab. 3.6. In general, higher resolution levels require more computational resources in the network, and more resources to store and render the mesh. Still, our approach supports scalable reconstruction and can be applied to scenarios with limited computational resources. Moreover, despite our proposed method outperforming our conference version, the computational costs barely increase. Here, “baseline” means only generating the MANO mesh in our network.

Visualization results. The qualitative reconstruction results are shown in Fig. 3.15. We observe that even when MANO is upsampled to 200,000 vertices, it still does not capture personalized details, while our results provide better shape details.

Chapter 4

Self-supervised High-fidelity Hand Reconstruction

1 Introduction

High-fidelity 3D hand reconstruction has emerged as a critical component in modern AR/VR applications. In consumer-grade immersive experiences, users consistently prefer realistic hand representations over simplified parametric meshes such as MANO [146]. However, accurately reconstructing hands in 3D remains challenging due to their complex geometry, intricate articulations, and the significant difficulty in acquiring high-quality 3D training data.

Recent advances in high-fidelity hand reconstruction, exemplified by methods such as [147], have made substantial progress. However, as illustrated in Fig. 4.1, these approaches rely heavily on 3D scanned ground-truth data for training supervision. Collecting such high-fidelity 3D hand scans presents formidable challenges: it requires specialized hardware setups, involves considerable expense, and most critically, suffers from limited scalability. As noted in [68], comprehensive 3D hand capture typically requires elaborate equipment like the lightStage system used in [91] — setups involving dozens of synchronized cameras

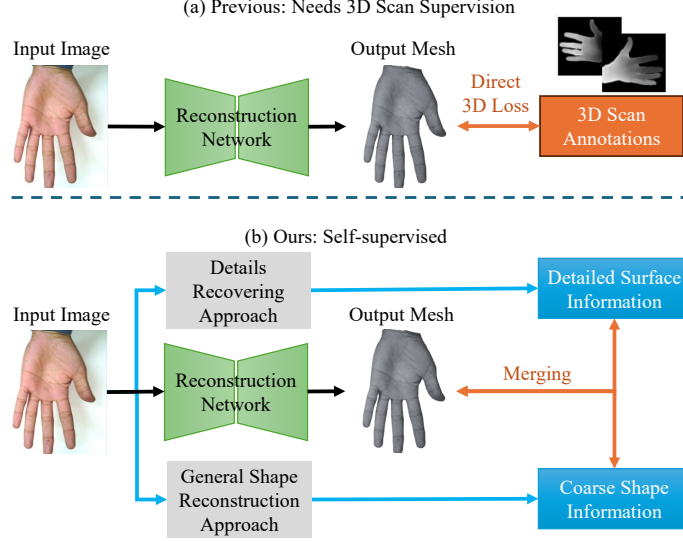


Figure 4.1: (a) Existing high-fidelity 3D hand reconstruction methods typically rely on specialized 3D scan ground-truth data, which require expensive hardware, time-consuming procedures, and controlled environments. (b) Our self-supervised approach reconstructs high-fidelity 3D hands directly from image inputs, leveraging general shape and detail priors without requiring 3D annotations. This method reduces reliance on specialized 3D-scanned data and broadens applicability across diverse subjects.

and light sources. A single hand scan acquisition can take anywhere from several minutes to over half an hour, and subjects must be present at specific locations with controlled environments. These constraints severely limit subject diversity and consequently restrict the generalizability of trained models across different populations. These limitations are evident in datasets like DeepHandMesh [110], which includes fewer than 10 subjects, leading to models that struggle to accurately reconstruct high-fidelity details for hands outside the limited training distribution.

To overcome these diversity limitations imposed by data constraints, we propose a self-supervised approach that reconstructs high-fidelity 3D hands from ordinary RGB images without requiring explicit 3D annotations. By leveraging readily available 2D hand photographs, our method fundamentally reduces dependency on scarce 3D-scanned data while expanding applicability across a much broader range of subjects. The primary challenge lies in reliably deriving accurate 3D information from 2D images alone. Fortunately, human hands possess rich geometric priors, with both overall shape and surface details following

patterns that correlate strongly with appearance. Existing techniques such as GeoWizard [148] and Shape-from-Shading methods like [149] already demonstrate partial capability in encoding appearance-to-geometry relationships without requiring hand-specific 3D ground truth, providing valuable prior knowledge that circumvents limitations imposed by 3D data scarcity. By designing a self-supervised framework that effectively leverages these priors for both global structure and local surface details, we enable high-fidelity reconstruction directly from image inputs without 3D annotations, offering a solution with substantially reduced data requirements and significantly greater adaptability to diverse hands.

Based on these insights, we developed *FlipFlop*, a method that reconstructs textured high-fidelity 3D hands using just two RGB images capturing the front and back of the hand. At the core of our approach is a strategy for extracting and integrating general shape priors and detailed geometry information from different off-the-shelf models through a frequency-based regulation loss inspired by [17]. Unlike conventional per-vertex loss formulations, this frequency-based regulation decomposes hand shape by frequency bands, enabling more effective supervision of high-frequency details extracted from existing models. Simultaneously, it applies stronger constraints on low-frequency components to maintain alignment with general hand shape priors derived from other sources. We further introduce a novel color regulation loss that operates alongside shape detail modeling. Recognizing that local color variations across a single hand are typically subtle, with surface appearance primarily defined by geometric details rather than texture, this color loss enforces consistency in color distribution while encouraging the model to express appearance variations through meaningful surface geometry changes rather than superficial color adjustments. To accommodate different practical deployment scenarios, we designed two complementary workflows: a *direct inference* pipeline that requires no training data but involves optimization during inference, and a *fast inference* pipeline that delivers rapid results through prior training on collected hand image datasets. For objective evaluation, we created a comprehensive benchmark dataset with ground-truth 3D scans, providing a reliable foundation for assessing

reconstruction accuracy.

In summary, our contributions are as follows:

- We introduce *FlipFlop*, a novel self-supervised approach for reconstructing textured high-fidelity 3D hands from just two RGB images without requiring 3D annotations, significantly expanding the accessibility and applicability of detailed hand reconstruction.
- We develop a frequency-based shape regulation loss that enables the model to effectively integrate priors from multiple sources, balancing between global structure and fine details while leveraging the complementary strengths of different off-the-shelf methods. We further introduce a specialized color regulation loss that encourages the model to represent appearance variations through geometric surface details rather than merely color variations.
- We present two complementary workflows—direct inference and fast inference—offering flexible trade-offs between training requirements and inference speed to accommodate different deployment scenarios.
- We establish a new benchmark dataset specifically designed for evaluating 3D hand reconstruction quality from front and back RGB images, where FlipFlop demonstrates consistently superior performance compared to state-of-the-art methods, particularly in capturing fine surface details.

2 Related Works

Human hand reconstruction. 3D hand reconstruction has garnered significant attention in recent years, with methods achieving promising results in reconstructing hand pose and general shape. Traditional approaches [150, 82, 151, 152, 92, 142, 80, 83, 133, 108, 153, 154, 155, 156, 157, 158, 159, 160] often rely on parametric models like MANO

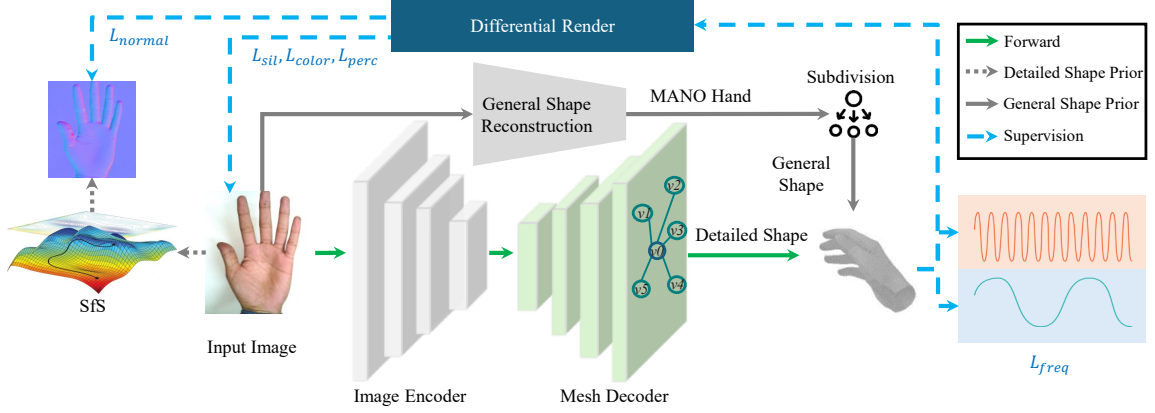


Figure 4.2: Overview of our self-supervised pipeline for high-fidelity 3D hand reconstruction. From a single RGB image, we obtain a coarse MANO [146]-based mesh, subdivide it for higher resolution, and refine it with per-vertex displacements predicted by a detail enhancement network. A differentiable renderer projects the refined mesh for image-space supervision using multiple loss terms (e.g., perceptual l_{perc} , silhouette l_{sil} , Laplacian color l_{color} and normal l_{normal} , frequency-based l_{freq}). This framework allows end-to-end training without requiring 3D ground-truth scans.

[146], which provide a simplified representation of hand shape and pose. However, these MANO-based methods are limited in their capacity to capture high-fidelity details, as they are constrained by the model’s low resolution and simplified mesh topology. Recent works [110, 147, 161, 17] have attempted to address these limitations by employing high-fidelity reconstruction techniques that leverage 3D scan data for training. Unfortunately, these methods face scalability issues due to the high cost and effort required to acquire diverse and detailed 3D hand scans, as datasets such as DeepHandMesh [110] are often limited to a small number of subjects. In contrast, our approach eliminates the dependency on 3D annotations by leveraging self-supervised learning and incorporating shape and detail priors from off-the-shelf methods, enabling high-fidelity reconstructions with improved generalizability.

Self-supervised 3D human reconstruction. Self-supervised approaches for 3D human reconstruction have emerged as an alternative to address the scarcity of 3D ground-truth data. These methods often exploit human body priors to estimate general shape and depth information from monocular images without requiring explicit 3D supervision. Notable

works [103, 162] have shown the efficacy of using silhouette constraints, keypoint alignment, and differentiable rendering to recover coarse human shapes. However, capturing fine details, especially for articulated parts like hands, remains challenging due to the lack of high-resolution priors and the inherent ambiguity in image-to-geometry mappings. Our method bridges this gap by leveraging detailed priors from off-the-shelf models and incorporating a frequency-based regulation loss, which effectively separates global shape constraints from high-frequency details, thus enabling detailed reconstructions without 3D ground-truth annotations.

Detail enhancement in 3D reconstruction. Enhancing fine-scale details in 3D reconstructions has been explored in various domains, including face [163, 164] and object modeling [165]. Techniques like Shape-from-Shading (SfS) [149] and neural rendering [166] have demonstrated the potential of leveraging image cues to refine geometry. While these methods focus on static objects or limited articulation, their principles inspire our approach to enhance hand reconstructions. By integrating multi-scale priors and applying losses that emphasize local surface variations, such as Laplacian color and normal losses, our method achieves detailed hand reconstructions that capture subtle geometric and appearance features. Furthermore, the use of a frequency-based regulation loss allows for adaptive detail enhancement, ensuring a balance between global structure and local fidelity.

In summary, our work combines advancements in self-supervised learning, detail refinement, and 3D hand reconstruction to address the limitations of existing methods. By eliminating the need for 3D ground-truth annotations and introducing novel loss functions, we provide a scalable and effective solution for high-fidelity hand modeling.

3 Fully-supervised High-fidelity Hand Reconstruction

We introduce *FlipFlop*, a self-supervised analysis via synthesis method for high-fidelity 3D hand reconstruction from only a pair of RGB images capturing the front and back views

of a hand. Building upon coarse hand meshes generated by HaMeR [151], our approach enhances these initial meshes with fine-scale details without the need for 3D ground truth annotations. As depicted in Figure 4.2, our pipeline comprises several key components: a coarse hand reconstruction module, a detail enhancement network, a differentiable renderer, and multiple loss functions designed to preserve geometric and appearance details.

3.1 Coarse Hand Reconstruction

Our reconstruction pipeline begins by obtaining an initial hand mesh estimation using HaMeR [151], which provides a MANO-based reconstruction with pose parameters $\theta \in \mathbb{R}^{48}$ (16 joints \times 3 rotation angles) and shape parameters $\beta \in \mathbb{R}^{10}$. The MANO function \mathcal{M} maps these parameters to vertex positions:

$$\mathbf{v}_{\text{mano}} = \mathcal{M}(\theta, \beta) \in \mathbb{R}^{778 \times 3}. \quad (4.1)$$

Initial reconstruction. While HaMeR provides a good initialization for global hand pose and shape, the base MANO mesh has significant limitations for high-fidelity reconstruction. Its low resolution of only 778 vertices is insufficient for capturing fine-scale surface details such as wrinkles, creases, and skin deformations. Additionally, challenging hand poses can lead to misalignments between the reconstructed mesh and the input images, necessitating further refinement.

Mesh subdivision. To support high-frequency geometric details, we apply the Loop subdivision to the MANO template mesh in [17] to generate a more detailed mesh. In each iteration, new vertices are introduced at edge midpoints, and vertex positions are updated using weighted averages of neighboring vertices, increasing the mesh resolution from 778 to over 12,000 vertices, with faces expanding from 1,538 to roughly 25,000. The resulting high-resolution mesh provides sufficient geometric freedom to capture millimeter-scale surface details while maintaining smooth surface continuity through the subdivision rules.

Additionally, the regular mesh topology created by this process is particularly beneficial for subsequent graph convolution operations.

Optimization strategy. We implement the subdivision process efficiently using sparse matrix operations, expressing the final subdivided mesh as:

$$\mathbf{v}_{\text{coarse}} = \mathbf{S}\mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta}), \quad (4.2)$$

where \mathbf{S} is the subdivision matrix encoding the Loop subdivision rules. During training, our method jointly optimizes the MANO parameters $(\boldsymbol{\theta}, \boldsymbol{\beta})$ for improved global alignment and the per-vertex displacements $\boldsymbol{\delta}_v$ for detail enhancement. This dual optimization ensures both accurate hand pose and shape, as well as high-fidelity surface details. The subdivided mesh $\mathbf{v}_{\text{coarse}}$ serves as an ideal base template, preserving the anatomical structure defined by MANO while providing the geometric flexibility needed for detailed reconstruction.

3.2 Detail Enhancement Network

Our detail enhancement network adopts an encoder-decoder architecture to translate 2D image observations into 3D geometric details.

Image encoder. The image encoder takes a single RGB hand image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ as input and extracts multi-scale features through a convolutional neural network. The encoder outputs a feature vector $\mathbf{f} \in \mathbb{R}^D$ that encodes both global hand structure and local appearance details.

Mesh decoder. The mesh decoder takes the image feature \mathbf{f} and predicts per-vertex displacement vectors $\boldsymbol{\delta}_v \in \mathbb{R}^{N \times 3}$ relative to the subdivided coarse mesh vertices, where N is the number of vertices after subdivision. The decoder uses graph convolutional layers to process mesh features while maintaining mesh topology. The enhanced mesh vertices are computed as:

$$\mathbf{v}_{\text{detailed}} = \mathbf{v}_{\text{coarse}} + \boldsymbol{\delta}_v, \quad (4.3)$$

where $\mathbf{v}_{\text{coarse}}$ are the vertices of the subdivided HaMeR mesh.

3.3 Differentiable Rendering

To enable end-to-end training without 3D supervision, we employ a differentiable renderer \mathcal{R} that projects the enhanced 3D mesh back to 2D. We represent the scene lighting using second-order spherical harmonics (SH) [167] with nine coefficients per color channel. Given the enhanced mesh vertices $\mathbf{v}_{\text{detailed}}$, surface normals \mathbf{N} , and SH lighting coefficients γ , the rendered image is computed as:

$$\mathbf{I}_r = \mathcal{R}(\mathbf{v}_{\text{detailed}}, \gamma) = \sum_{i=1}^9 \gamma_i \mathbf{H}_i(\mathbf{N}), \quad (4.4)$$

where \mathbf{H}_i are the spherical harmonics basis functions. Both surface normals \mathbf{N} and SH lighting coefficients γ can be estimated using off-the-shelf estimation approaches, e.g., [168].

3.4 Loss Functions

We train our network using five complementary loss terms that work together to produce high-quality reconstructions.

Perceptual loss. We employ a direct image-space supervision through rendered images:

$$\mathcal{L}_{\text{perc}} = \|\mathbf{I} - \mathcal{R}(\mathbf{v}_{\text{detailed}}, \gamma)\|_2, \quad (4.5)$$

where \mathcal{R} is our differentiable renderer and γ represents the spherical harmonics lighting parameters. This loss provides a global supervision signal for both geometry and appearance.

Laplacian color and normal losses. To preserve fine-scale surface details, we introduce Laplacian losses on both color and normal variations. For normal supervision, we leverage

normal maps estimated through shape-from-shading [149]:

$$\mathcal{L}_{\text{color}} = \|\Delta \mathbf{C} - \Delta \mathbf{C}_{\text{target}}\|_1, \quad (4.6)$$

$$\mathcal{L}_{\text{normal}} = \|\Delta \mathbf{N} - \Delta \mathbf{N}_{\text{target}}\|_1 \quad (4.7)$$

where the discrete Laplacian operator Δ for vertex i and feature \mathbf{X} (either color \mathbf{C} or normal \mathbf{N}) is defined as:

$$\Delta \mathbf{X}_i = \mathbf{X}_i - \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{X}_j \quad (4.8)$$

Here, $\mathcal{N}(i)$ represents the one-ring neighbors of vertex i . The target Laplacians are computed by projecting mesh vertices to the image plane and sampling corresponding features: colors directly from the input image for $\Delta \mathbf{C}_{\text{target}}$, and normal values from shape-from-shading estimates for $\Delta \mathbf{N}_{\text{target}}$. These Laplacian loss terms effectively capture local variations while being invariant to global transformation, naturally emphasizing high-frequency details crucial for realistic wrinkles and creases, while providing robustness to lighting and camera variations.

Silhouette loss. The initial MANO pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$ parameters obtained from HaMeR [151] provide a coarse reconstruction but may not perfectly align with the input image. To address this, we jointly optimize these MANO parameters along with detail enhancement during training. The silhouette loss ensures accurate alignment between the projected mesh and the input hand mask \mathbf{M} :

$$\mathcal{L}_{\text{sil}} = \text{BCE}(\mathcal{R}_{\text{mask}}(\mathbf{v}_{\text{detailed}}, \boldsymbol{\theta}, \boldsymbol{\beta}), \mathbf{M}), \quad (4.9)$$

where $\mathcal{R}_{\text{mask}}$ is a differentiable rendering function that generates a binary mask of the hand mesh. The binary cross-entropy (BCE) loss effectively penalizes misalignment between the

rendered and target masks:

$$\begin{aligned} \text{BCE}(\mathbf{P}, \mathbf{M}) = & \\ -\frac{1}{|\Omega|} \sum_{i \in \Omega} [\mathbf{M}_i \log(\mathbf{P}_i) + (1 - \mathbf{M}_i) \log(1 - \mathbf{P}_i)], & \end{aligned} \quad (4.10)$$

where Ω is the image domain, \mathbf{P} is the rendered mask probability map, and $\mathbf{M}_i \in \{0, 1\}$ is the binary target mask M at pixel i .

This loss serves multiple purposes: 1) refines initial MANO parameters for better global pose and shape alignment; 2) guides the detail enhancement to maintain silhouette consistency; 3) provides a strong supervision signal even without 3D ground truth; and 4) helps prevent geometric artifacts that could distort the hand silhouette. The gradients from this loss flow back to both the MANO parameters $(\boldsymbol{\theta}, \boldsymbol{\beta})$ and the vertex offsets $\boldsymbol{\delta}_v$, allowing joint optimization of global pose and local details. This is particularly important for capturing hand poses with self-occlusions or complex articulations where initial HaMeR estimates may be imprecise.

Frequency loss. To ensure the enhanced mesh maintains the global structure while adding plausible fine-scale details, we decompose the mesh geometry into different frequency bands using spectral analysis. Given a mesh with N vertices, we first construct the graph Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \quad (4.11)$$

where \mathbf{D} is the degree matrix, and \mathbf{W} is the adjacency matrix with cotangent weights.

The eigenvectors $\{\mathbf{e}_k\}_{k=1}^N$ of \mathbf{L} form an orthogonal basis for mesh deformation, ordered by their corresponding eigenvalues λ_k from low to high frequencies. We decompose the vertex positions into frequency components using this basis:

$$\mathbf{v} = \sum_{k=1}^N \alpha_k \mathbf{e}_k, \quad \text{where} \quad \alpha_k = \mathbf{v}^T \mathbf{e}_k. \quad (4.12)$$

The low- and high-frequency components are then obtained by:

$$\mathbf{v}_{\text{low}} = \sum_{k=1}^K \alpha_k \mathbf{e}_k, \quad \mathbf{v}_{\text{high}} = \sum_{k=K+1}^N \alpha_k \mathbf{e}_k, \quad (4.13)$$

where K is empirically set to capture the first 20% of the frequency spectrum. Fig. 4.4 shows a sketch of how frequency components correspond to the shape.

Our frequency loss applies different constraints to these components:

$$\mathcal{L}_{\text{freq}} = \lambda_{\text{low}} \|\mathbf{v}_{\text{low}} - \mathbf{v}_{\text{coarse_low}}\|_2 + \lambda_{\text{high}} \|\mathbf{v}_{\text{high}}\|_2, \quad (4.14)$$

where the first term ensures the low-frequency components maintain the overall shape from the coarse HaMeR reconstruction, and the second term acts as a noise regularizer to prevent excessively high-frequency details that might lead to artifacts. The weights λ_{low} and λ_{high} balance between shape preservation and detail enhancement, empirically set to 0.2 and 1, respectively. Fig. 4.4 shows how the hand shape changes when the frequency increases, and indicates the frequency bands used in this paper.

This spectral approach has several advantages: 1) provides natural separation of global shape and local details; 2) enables independent control over different geometric scales; 3) helps prevent overfitting to noise in the input images; and 4) maintains mesh smoothness without over-smoothing important features. The frequency decomposition is differentiable, allowing end-to-end training while effectively guiding the network to produce geometrically plausible detail enhancement.

Total loss. The total loss is a weighted combination of the aforementioned loss terms:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & w_{\text{perc}} \mathcal{L}_{\text{perc}} + w_{\text{color}} \mathcal{L}_{\text{color}} \\ & + w_{\text{normal}} \mathcal{L}_{\text{normal}} + w_{\text{sil}} \mathcal{L}_{\text{sil}} + w_{\text{freq}} \mathcal{L}_{\text{freq}}. \end{aligned} \quad (4.15)$$

3.5 Training and Inference

We provide two workflows for practical deployment:

Direct inferences In this workflow, we optimize the network weights and mesh details directly on the test images without requiring pre-training. While slower, this approach can potentially achieve better quality by tailoring the model to the specific input.

Fast inference. Alternatively, we pre-train the network on a large dataset of hand images. During inference, a single forward pass produces detailed reconstructions, enabling real-time applications.

In both cases, our method requires only RGB images for training and testing, making it highly practical for real-world deployment.

4 Experiments

4.1 Training Dataset

We use the 11K Hands [169] dataset as our training set. This dataset was originally created for palmprint recognition and contains over 11,000 images from 190 different subjects. Such a dataset provides our model with a larger number of subjects and a wider variety of hand images. Since our task involves using both the front and back of the hand as inputs for self-supervised high-fidelity mesh reconstruction, the palmprint recognition dataset—with its images of both sides of the hand and clear detailed features—serves as an ideal input source for our purposes. Fig. 4.3(a) shows several examples of these training inputs. As can be seen, the dataset provides very clear palmprint details.

4.2 *HandScan* Benchmark

Our dataset contains nearly 400 samples from 16 different subjects. Each sample is an image of either the palm or the back of a hand, along with its corresponding 3D scan.

Table 4.1: *HandScan* and 11k Hands dataset attributes. Compared with 11k Hands covers more subjects, *HandScan* provides high-resolution 3D scans for 16 subjects with 3D scanned shape ground-truth and MANO registration.

Dataset	HandScan	11k Hands
Number of Subjects	16	190
MANO registration	Yes	No
Scanned 3D shape groundtruth	Yes	No
Total number of inputs	392	11,000
Scanner resolution	0.174 mm	N/A
Number of Poses	6	4
Handedness	Left & Right	Left & Right

Table 4.2: SOTA comparison of our result with previous works. We evaluate the hand’s general shape using Chamfer Distance (CD) and the details fidelity using FSNR [17]. As shown in the table below, our methods outperform SOTA methods, especially in detail measurements. We also report the average inference time. For CD and Inference Time, a lower number means better performance. For FSNR, the higher the number, the better the performance. The inference time is measured on a single NVIDIA A40 GPU.

Methods	CD (mm)↓	FSNR↑	Inference Time (s)↓
HaMeR [151]	6.89	-3.2	0.11
Ours (Fast Inference)	6.90	-0.87	0.13
Ours (Direct)	6.90	-0.74	39.0

We use this benchmark to evaluate our reconstruction algorithm. Note that there is no training set in our benchmark; the model is trained on the Hands11K [169] dataset and then directly evaluated on *HandScan*. Tab. 4.1 shows the statistical information of our dataset, and Fig. 4.3(b) presents some examples of our input images and the corresponding 3D scans. During data collection, to minimize the influence of background differences on model transfer, we chose backgrounds similar to those in the training data. In this work, we focus on unsupervised high-fidelity reconstruction, hence we simplify the requirements for background generalization.

Data preprocessing. We use the fine-grain parametric hand mesh proposed in [17] to fit the *HandScan* dataset. This mesh is parameterized in the same way as MANO but contains 12,337 vertices to ensure high-fidelity representation. Specifically, we employ a multi-step

Table 4.3: We do an ablation study on the fast inference part of our method. The result demonstrates the importance of each loss term. Removing any single loss, including perceptual loss, Laplacian loss, silhouette loss, and frequency loss, it degrades our method’s reconstruction performance, underscoring its contribution to the final result.

Methods	Chamfer Distance (mm)↓	FSNR↑
w/o Perceptual Loss	6.93	-0.97
w/o Laplacian Losses	7.02	-1.88
w/o Silhouette Loss	31.3	-2.52
w/o Frequency Loss	6.90	-2.24
Ours (Fast Inference)	6.90	-0.87

fitting approach. First, we use Mediapipe [170], an off-the-shelf method, to extract 2D keypoints and then perform a coarse alignment based on these keypoints. This alignment is achieved through an optimization process, with the following energy function:

$$e_{2d} = \|\Pi J(M(\theta, \beta, R, t)) - p\|^2, \quad (4.16)$$

where Π is the projection matrix from 3D to 2D. J is the joint regressor same as in MANO [146]. M is the MANO model. θ , β , R , and t are the hand pose, hand shape, hand global rotation, and hand global translation, respectively. p is the 2D keypoint. At this step, we first fix θ and β to optimize the coarse global rotation and translation, and then free θ and β to optimize θ , β , R , and t together.

4.3 Implementation Details

We use Swin Transformer [171] (swin_b) for image encoder, and a GCN network from [172] as the mesh decoder. The output dimension of the GCN network is changed to 12,337 to fit the mesh vertices number. We use the shape-from-shading approach [149] to generate the normal map, and the pretrained method HaMeR [151] to generate the general hand mesh. We also use Mediapipe [170] for keypoint extraction in Sec. 4.2, and use SAM [173] to generate hand mask in Sec. 3.4. We set w_{perc} , w_{color} , w_{normal} , w_{sil} , and w_{freq} to 2, 10, 1, 1,

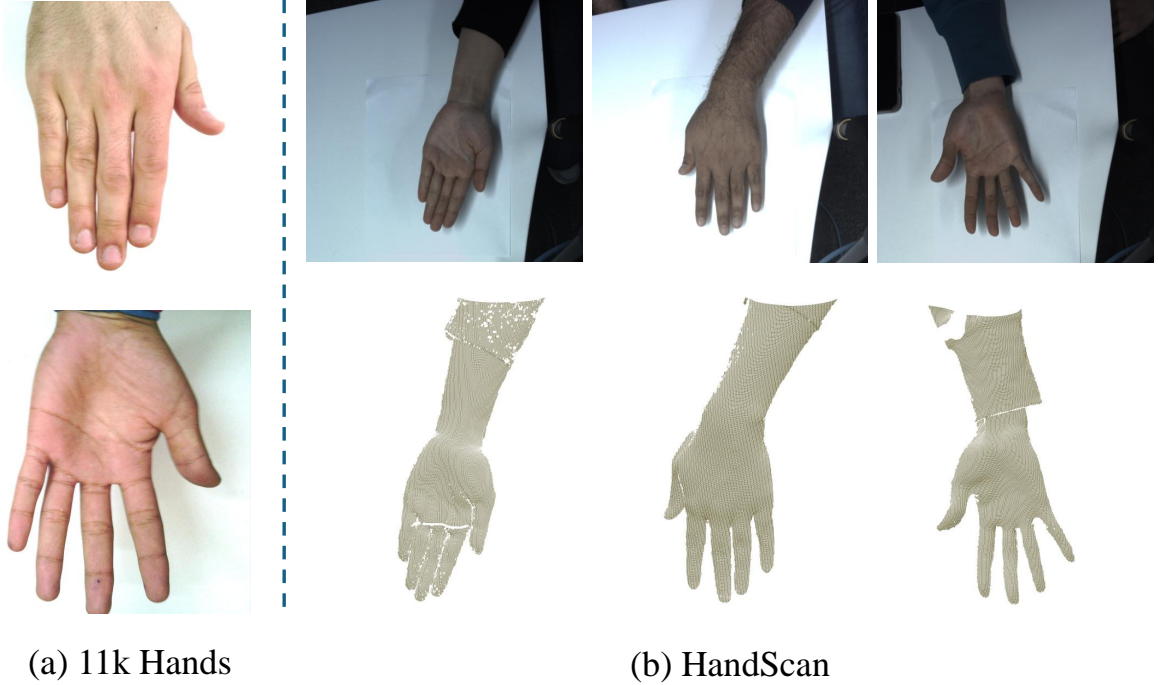


Figure 4.3: Example data of 11k Hands (a) and our benchmark *HandScan* (b). For *HandScan*, the top row is the input images, and the bottom row is the hand scan data. As shown in the figure, our dataset scanning has good hand shape details for evaluation.

and 0.2, respectively. The learning rate is set to 1×10^{-4} . We use PyTorch [67] to implement our code. The framework is trained on a single NVIDIA A40 GPU for 10 epochs. The total training time for the fast inference framework is around 7 hours.

4.4 Quantitive and Qualitative Results

SOTA comparison. As shown in Tab. 4.2, we measure the hand’s global shape using Chamfer Distance (CD) and its fine details via Frequency Signal-to-Noise Ratio (FSNR), from [17]. The results show our approach outperforms existing methods, especially in high-frequency details. We also report the average inference times, which are measured using a single NVIDIA A40 GPU. Note that lower values are better for both CD and inference time, whereas higher values are preferable for FSNR.

Ablation studies. We verify the effectiveness of our loss functions using an ablation study. The experiment is done using the fast inference part of our method. As shown in

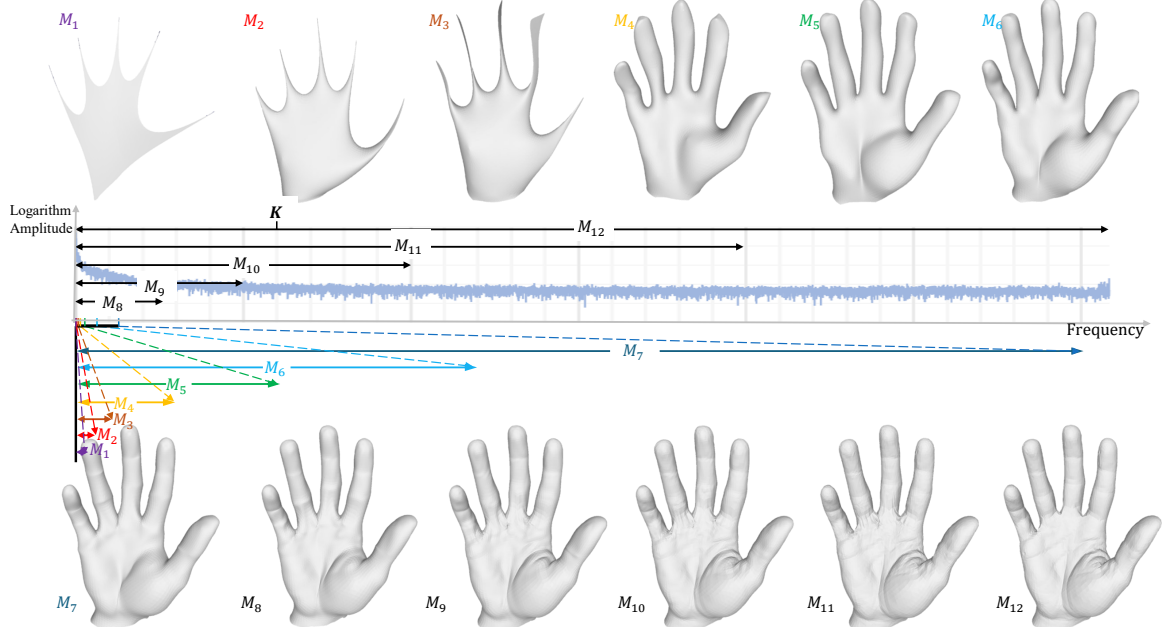


Figure 4.4: Example of decomposing a hand mesh into different frequency bands. We accumulate frequency components from low to high, resulting in 12 hand meshes (M_1 to M_{12}). The boundary between low and high frequencies is marked at \mathbf{K} , roughly between M_9 and M_{10} . The central figure illustrates the overall frequency decomposition of the hand shape.

Tab. 4.3, by removing any of those loss functions, including perceptual loss, Laplacian loss, silhouette loss, and frequency loss, the performance of our method drops.

Frequency of the hands. We show an example of the shape of the human hand of different frequency components in Fig. 4.4. The figure in the middle is the frequency decomposition result of the hand. We accumulate the frequency components from low to high, and get 12 different hand meshes, namely from M_1 to M_{12} . In our experiments, we divide low frequency and high frequency at the point \mathbf{K} in the figure (somewhere between M_9 and M_{10}).

Visualization of the normals, general shape, and mesh keypoints. We visualize the normal maps, general shape mesh, and 2D key points that are used in our method in Fig. 4.5 from column 1 to column 3, left to right. In the third column, the green dots are the keypoint generated from the off-the-shelf 2D keypoint estimator, while the red dots are from 3D hand joint projection. We can see that our generated Normal maps have good detailed shapes, and

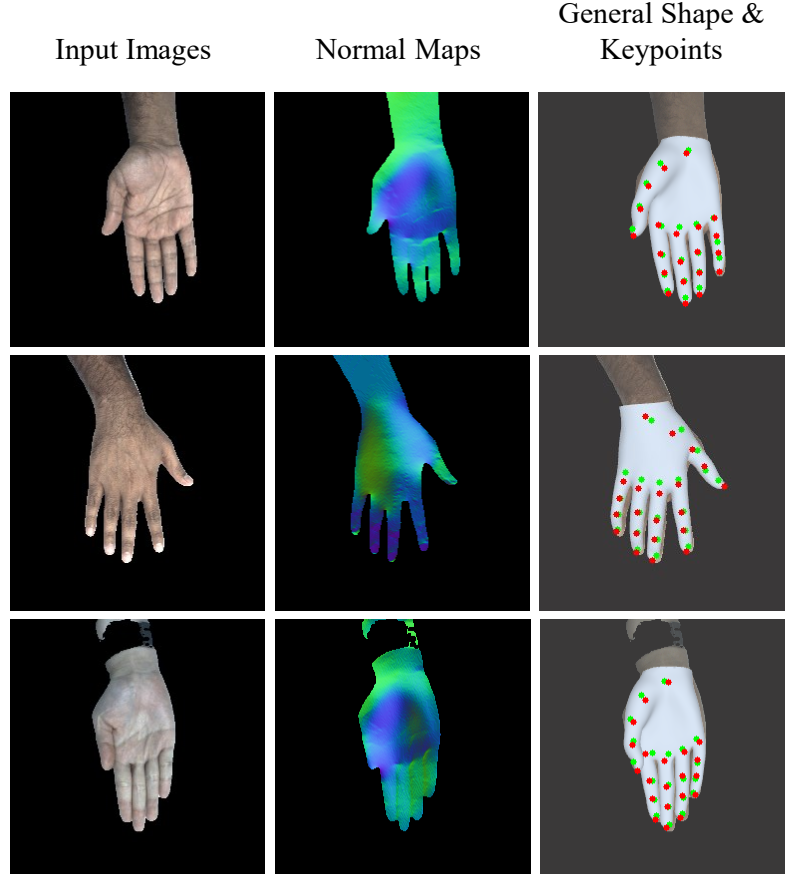


Figure 4.5: Visualization of the normal maps (2nd column from the left), general shape mesh, and 2D keypoints (3rd column from the left) used in our method. In the third column, green dots represent 2D keypoints from an off-the-shelf estimator, while red dots are the projections of 3D hand joints. We can observe the details and general shape alignment of the details provided by the normal map and general shape provided by conventional hand mesh reconstruction.

the general shape also highly aligns with the input image.

Visualized examples. We visualize our result in Fig. 4.6 for both (a) fast inference and (b) direct inference on *HandScan*. For direct inference, we also visualize our results on 11k hands (bottom row). As shown in the figure, our result has a better detail shape and fidelity than the baseline approach for both approaches.

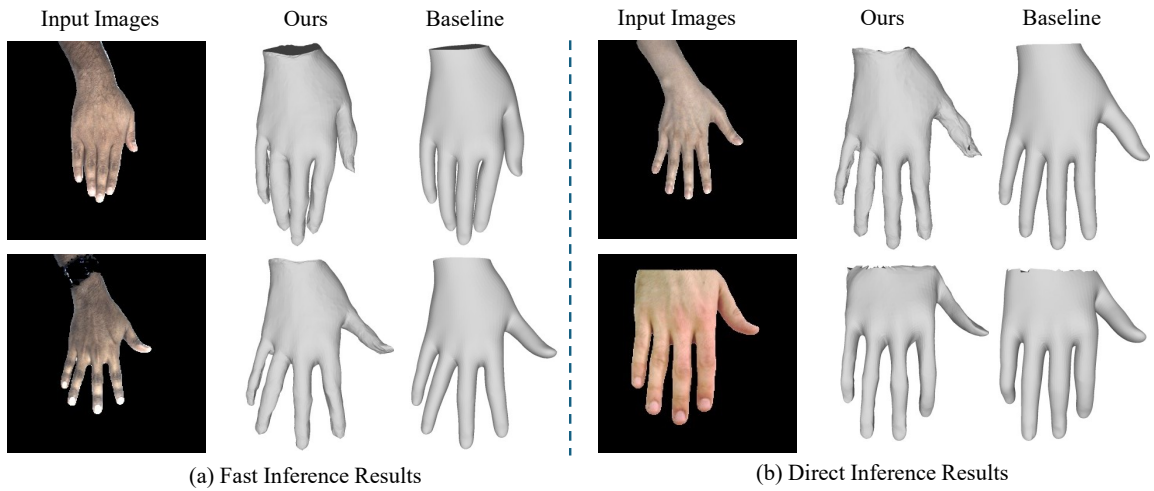


Figure 4.6: We visualize our results for both fast inference (a) and direct inference (b) on *HandScan*. For direct inference, we also visualize our results on 11k hands (bottom row). As shown in the figure, our result has a better detail shape and fidelity than the baseline approach for both fast inference approach and direct inference approach.

Chapter 5

Conclusion

This dissertation establishes a framework for assessing and creating high-fidelity 3D content, guided by human perception. We begin by introducing *Shape Grading*, a user-study benchmark that compiles quality scores for a wide array of distorted 3D meshes, spanning twelve ground-truth objects with seven distortion types at four severity levels. By comparing these human judgments against automated metrics, we illustrate how existing fidelity measures often fail to capture local details that play a critical role in perceived realism.

To address this shortcoming, we propose the Spectrum Area Under the Curve Difference (SAUCD) analytic metric, which leverages the discrete Laplace-Beltrami operator and Fourier transform to balance global structure with fine-grained surface details. By learning frequency-specific weights aligned with subjective evaluations, our method offers improved correlation with human perception. We then demonstrate the metric’s practical utility through two hand-reconstruction pipelines. The fully supervised approach employs a frequency-split network to preserve both coarse shape and detailed geometry, while the self-supervised FlipFlop system reconstructs textured, high-frequency hand meshes from just two RGB views.

Overall, this work closes crucial gaps in fidelity measurement and generation for AR/VR, combining large-scale perception studies with a spectrum-based metric and advanced recon-

struction techniques. The result is a robust pathway toward more immersive and realistic virtual experiences.

Appendix A

A Counterexample of the Original Cotan Formula not being Positive Semidefinite

We provide a simple mesh example to show that the original Cotan formula in Eq. (2.2) does not guarantee to be positive semidefinite. As shown in Fig. A.1a, we reconstruct a 4-vertex mesh that is not Delauney triangulated and the mixed Voronoi areas of the vertices are not all equal. We make the two faces on the bottom ($v_1v_2v_0$ and $v_3v_0v_2$) be two congruent obtuse isosceles triangles (shown in Fig. A.1b). The apex angles of the two isosceles triangles are $\frac{2\pi}{3}$, and the base angles are $\frac{\pi}{6}$. If we make the bottom two obtuse triangles form different angles to each other, the top two triangle faces ($v_0v_1v_3$ and $v_2v_3v_1$) are always congruent isosceles triangles (as in Fig. A.1c), and their apex angles vary continuously in the range of $(0, \frac{\pi}{3})$. Here, we make the bottom two obtuse triangles form a certain angle to each other so that the apex angles of the top two triangles are equal to $\frac{\pi}{6}$, which means their base angles are $\frac{5\pi}{12}$. For simplicity, we set the equal sides of the isosceles triangles to be 1 (shown in Fig. A.1a).

Now, we calculate the DLBO metric of this reconstructed mesh using the Cotan formula in Eq. (A.3). First, we calculate the mixed Voronoi area for each vertex. Because of the shape symmetry, we only need to calculate the mixed Voronoi areas for vertex v_0 and v_3 .

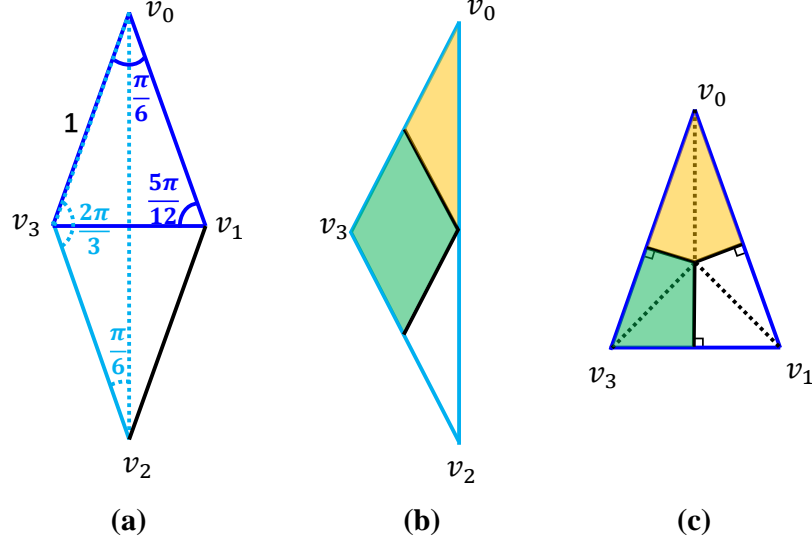


Figure A.1: A simple mesh example to show that the original Cotan formula does not guarantee to be positive semidefinite.

The mixed Voronoi areas for vertex v_2 and v_1 are equal to v_0 and v_3 , respectively. For vertex v_0 , its mixed Voronoi area A_0 can be calculated as the sum of 2 times of yellow area in Fig. A.1b and 1 time of yellow area in Fig. A.1c, which means

$$\begin{aligned}
 A_0 &= 2 \times \left(\frac{1}{4} \times \frac{1}{2} \cos \frac{\pi}{3} \right) + 1 \times \left(0.5 \tan \frac{\pi}{12} \times 0.5 \right) \\
 &= \frac{4 - \sqrt{3}}{8},
 \end{aligned} \tag{A.1}$$

where $\frac{1}{2} \cos \frac{\pi}{3}$ is the area of the outer triangle in Fig. A.1b and $0.5 \tan \frac{\pi}{12} \times 0.5$ is the area of the yellow part in Fig. A.1c. For vertex v_3 , its mixed Voronoi area A_3 can be calculated as the sum of 1 time of green area in Fig. A.1b and 2 times of green area in Fig. A.1c, which means

$$\begin{aligned}
 A_3 &= 1 \times \left(\frac{1}{2} \times \frac{1}{2} \cos \frac{\pi}{3} \right) \\
 &\quad + 2 \times \left(\frac{1}{2} \times \left(\sin \frac{\pi}{12} \cos \frac{\pi}{12} - 0.5 \tan \frac{\pi}{12} \times 0.5 \right) \right) \\
 &= \frac{3\sqrt{3} - 2}{8},
 \end{aligned} \tag{A.2}$$

where $\sin \frac{\pi}{12} \cos \frac{\pi}{12}$ is the area of the outer triangle in Fig. A.1c.

Second, we calculate the DLBO matrix according to Eq. (2.2). The DLBO matrix of the constructed mesh can be represented as

$$L = \begin{pmatrix} \frac{w_1}{2A_0} & \frac{w_0}{2A_0} & \frac{w_3}{2A_0} & \frac{w_0}{2A_0} \\ \frac{w_0}{2A_3} & \frac{w_2}{2A_3} & \frac{w_0}{2A_3} & \frac{w_4}{2A_3} \\ \frac{w_3}{2A_0} & \frac{w_0}{2A_0} & \frac{w_1}{2A_0} & \frac{w_0}{2A_0} \\ \frac{w_0}{2A_3} & \frac{w_4}{2A_3} & \frac{w_0}{2A_3} & \frac{w_2}{2A_3} \end{pmatrix}, \quad (\text{A.3})$$

where

$$\begin{aligned} w_0 &= -(\cot \frac{5\pi}{12} + \cot \frac{\pi}{6}) = -2, \\ w_1 &= 2(\cot \frac{5\pi}{12} + \cot \frac{\pi}{6} + \cot \frac{2\pi}{3}) = 4 - \frac{2\sqrt{3}}{3}, \\ w_2 &= 2(\cot \frac{5\pi}{12} + \cot \frac{\pi}{6} + \cot \frac{\pi}{6}) = 4 + 2\sqrt{3}, \\ w_3 &= -2 \cot \frac{2\pi}{3} = \frac{2\sqrt{3}}{3}, \\ w_4 &= -2 \cot \frac{\pi}{6} = -2\sqrt{3}. \end{aligned} \quad (\text{A.4})$$

Then, we can calculate the symmetric part of L as

$$L_{sym} = \frac{L + L^\top}{2}. \quad (\text{A.5})$$

We use Wolfram Mathematica [174] to calculate the eigenvalues of L_{sym} . The 4 eigenvalues are

$$\begin{aligned} \lambda_0 &= \frac{2 - \frac{2\sqrt{3}}{3}}{A_0}, \\ \lambda_1 &= \frac{2 + 2\sqrt{3}}{A_3}, \\ \lambda_2 &= \frac{A_0 + A_3 - \sqrt{2(A_0^2 + A_3^2)}}{A_0 A_3}, \\ \lambda_3 &= \frac{A_0 + A_3 + \sqrt{2(A_0^2 + A_3^2)}}{A_0 A_3}. \end{aligned} \quad (\text{A.6})$$

It is obvious that when A_0 and A_3 are both greater than 0, λ_0 , λ_1 , and λ_3 will be greater than 0. However, for λ_2 , we have

$$\begin{aligned}
\lambda_2 &= \frac{A_0 + A_3 - \sqrt{2(A_0^2 + A_3^2)}}{A_0 A_3} \\
&= \frac{\sqrt{A_0^2 + A_3^2 + 2A_0 A_3} - \sqrt{2(A_0^2 + A_3^2)}}{A_0 A_3} \\
&\leq \frac{\sqrt{A_0^2 + A_3^2 + (A_0^2 + A_3^2)} - \sqrt{2(A_0^2 + A_3^2)}}{A_0 A_3} \\
&= 0.
\end{aligned} \tag{A.7}$$

The equation holds if and only if $A_0 = A_3$. We know from Eq. (A.1) and Eq. (A.2) that $A_0 \neq A_3$. Thus, we have

$$\lambda_2 < 0, \tag{A.8}$$

which means in the given mesh example, the original Cotan formula is not positive semidefinite.

Appendix B

Proof of Positive-semidefiniteness of Revised Cotan Formula

In this section, we prove that our revised version of the Cotan formula in Eq. (2.4) is positive semidefinite. Here, the DLBO defined in Eq. (2.4) is

$$L_{ij} = \begin{cases} \frac{1}{2} \sum_{j \in N(i)} A_i^{-\frac{1}{2}} A_j^{-\frac{1}{2}} |\cot \alpha_{ij} + \cot \beta_{ij}|, & i = j \\ -\frac{1}{2} A_i^{-\frac{1}{2}} A_j^{-\frac{1}{2}} |\cot \alpha_{ij} + \cot \beta_{ij}|, & i \neq j \wedge j \in N(i) \\ 0, & i \neq j \wedge j \notin N(i). \end{cases} \quad (\text{B.1})$$

According to the Gershgorin circle theorem [175], for every eigenvalue λ_k of L ,

$$\lambda_k \in \bigcup_i S_i, \quad (\text{B.2})$$

where S_i is the i th Gershgorin disc. The Gershgorin disc is defined as

$$S_i = \{z \in \mathbb{C} : |z - L_{ii}| \leq R_i = \sum_{i \neq j} |L_{ij}|\}, \quad (\text{B.3})$$

where \mathbb{C} means the complex space. Since L is a real symmetric matrix, according to Eq. (B.1), the Gershgorin disc degenerates into a line segment in the real space as

$$S_i = \{s \in \mathbb{R} : |s - L_{ii}| \leq R_i = \sum_{i \neq j} |L_{ij}|\}. \quad (\text{B.4})$$

From Eq. (B.1), we can also have

$$\sum_{i \neq j} |L_{ij}| = \sum_{j \in N(i)} \frac{|\cot \alpha_{ij} + \cot \beta_{ij}|}{2\sqrt{A_i A_j}} = L_{ii}. \quad (\text{B.5})$$

Note that $L_{ii} \geq 0$, so having Eq. (B.5), from Eq. (B.4) we get

$$S_i = \{s \in \mathbb{R} : |s - L_{ii}| \leq R_i = L_{ii}\} \Leftrightarrow 0 \leq S_i \leq 2L_{ii}. \quad (\text{B.6})$$

Thus, according to Eq. (B.2), we have

$$0 \leq \lambda_k \leq 2 \max_i L_{ii}, \forall 0 \leq k \leq N, \quad (\text{B.7})$$

where N is the number of vertices. Then, L is positive semidefinite since L is a real symmetric matrix and all its eigenvalues are greater than or equal to zero.

Q.E.D.

Appendix C

Proof of SAUCD Satisfies Metric

Definition in Spectrum Domain

In this section, we prove that our SAUCD satisfies the metric definition in spectrum domain. In metric geometry, a metric is defined as $d(M_A, M_B)$ which satisfies the following four conditions [176]

1. $D(x_A, x_B) \geq 0$,
2. $D(x_A, x_B) = 0$ if and only if $x_A = x_B$,
3. $D(x_A, x_B) = D(x_B, x_A)$, and
4. $D(x_A, x_B) \leq D(x_A, x_C) + D(x_C, x_B)$

for any inputs x_A , x_B , and x_C in the space.

In our case, we prove that if $D(x_A, x_B)$ is defined as $\int_{\lambda} |x_A(\lambda) - x_B(\lambda)| d\lambda$ as in Eq. (2.7), $D(x_A, x_B)$ is a metric in spectrum domain.

a) We prove $D(x_A, x_B) \geq 0$. This is simply because

$$|x_A(\lambda) - x_B(\lambda)| \geq 0 \tag{C.1}$$

for every λ , then

$$\int_{\lambda} |x_A(\lambda) - x_B(\lambda)| d\lambda \geq 0. \quad (\text{C.2})$$

b) We prove $D(x_A, x_B) = 0$ if and only if $x_A = x_B$. First, if $x_A = x_B$, then $D(x_A, x_B) = 0$ by definition. Second, if $\exists D(x_A, x_B) = 0$ that makes $x_A \neq x_B$, then by definition of $D(x_A, x_B)$, we have

$$\int_{\lambda_1}^{\lambda_2} |x_A(\lambda) - x_B(\lambda)| d\lambda = 0. \quad (\text{C.3})$$

For Eq. (C.3), we get $\forall \lambda \in [\lambda_1, \lambda_2]$, $x_A(\lambda) = x_B(\lambda)$. This contradicts to the hypothesis $x_A \neq x_B$. So, if $D(x_A, x_B) = 0$, then $x_A = x_B$. In summary, $D(x_A, x_B) = 0$ if and only if $x_A = x_B$.

c) We prove $D(x_A, x_B) = D(x_B, x_A)$. Because $|x_A(\lambda) - x_B(\lambda)| = |x_B(\lambda) - x_A(\lambda)|$, then $\int_{\lambda} |x_A(\lambda) - x_B(\lambda)| d\lambda = \int_{\lambda} |x_B(\lambda) - x_A(\lambda)| d\lambda$, which means $D(x_A, x_B) = D(x_B, x_A)$.

d) We prove

$$D(x_A, x_B) \leq D(x_A, x_C) + D(x_C, x_B). \quad (\text{C.4})$$

We first reform the left-hand side of Eq. (C.4) by separating the integration range as

$$D(x_A, x_B) = \int_{A>B} (x_A - x_B) d\lambda + \int_{B>A} (x_B - x_A) d\lambda, \quad (\text{C.5})$$

where we use $\int_{A>B} (x_A - x_B) d\lambda$ to indicate

$$\int_{\{\lambda | x_A(\lambda) > x_B(\lambda)\}} (x_A - x_B) d\lambda \quad (\text{C.6})$$

as the integration of $x_A - x_B$ on the λ range where $x_A(\lambda) > x_B(\lambda)$. $\int_{A<B} (x_B - x_A) d\lambda$ represents a similar meaning.

Using similar representations as in Eq. (C.5), we reform the two terms on the right-hand

side of Eq. (C.4) as

$$D(x_A, x_C) = \int_{A>C} (x_A - x_C) d\lambda + \int_{C>A} (x_C - x_A) d\lambda \quad (\text{C.7})$$

and

$$D(x_C, x_B) = \int_{B>C} (x_B - x_C) d\lambda + \int_{C>B} (x_C - x_B) d\lambda. \quad (\text{C.8})$$

We further decompose the left-hand side of Eq. (C.4) as

$$\begin{aligned} D(x_A, x_B) &= \int_{A>B} (x_A - x_B) d\lambda + \int_{B>A} (x_B - x_A) d\lambda \\ &= \int_{C \geq A > B} (x_A - x_B) d\lambda + \int_{A > C > B} (x_A - x_B) d\lambda \\ &\quad + \int_{A > B \geq C} (x_A - x_B) d\lambda + \int_{C \geq B > A} (x_B - x_A) d\lambda \\ &\quad + \int_{B > C > A} (x_B - x_A) d\lambda + \int_{B > A \geq C} (x_B - x_A) d\lambda. \end{aligned} \quad (\text{C.9})$$

Similarly, we decompose the right-hand side of Eq. (C.4) as

$$\begin{aligned} D(x_A, x_C) + D(x_C, x_B) &= \int_{A>C} (x_A - x_C) d\lambda + \int_{C>A} (x_C - x_A) d\lambda \\ &\quad + \int_{B>C} (x_B - x_C) d\lambda + \int_{C>B} (x_C - x_B) d\lambda \\ &= \int_{C \geq A > B} (2x_C - x_A - x_B) d\lambda + \int_{A > C > B} (x_A - x_B) d\lambda \\ &\quad + \int_{A > B \geq C} (x_A + x_B - 2x_C) d\lambda + \int_{C \geq B > A} (2x_C - x_A - x_B) d\lambda \\ &\quad + \int_{B > C > A} (x_B - x_A) d\lambda + \int_{B > A \geq C} (x_A + x_B - 2x_C) d\lambda. \end{aligned} \quad (\text{C.10})$$

In the 1st and 4th terms of Eq. (C.10), when $x_C(\lambda) \geq x_A(\lambda) > x_B(\lambda)$ or $x_C(\lambda) \geq x_B(\lambda) > x_A(\lambda)$, we have $2x_C - x_A - x_B \geq x_B - x_A$ (replace x_C with x_B) and $2x_C - x_A - x_B \geq x_A - x_B$ (replace x_C with x_A). Similarly, in the 3rd and 6th terms of Eq. (C.10), when $x_A(\lambda) > x_B(\lambda) \geq x_C(\lambda)$ or $x_B(\lambda) > x_A(\lambda) \geq x_C(\lambda)$, we have $x_A + x_B - 2x_C \geq x_B - x_A$

(replace x_C with x_A) and $2x_A + x_B - 2x_C \geq x_A - x_B$ (replace x_C with x_B). Thus, we can simplify Eq. (C.10) into

$$\begin{aligned}
& D(x_A, x_C) + D(x_C, x_B) \\
& \geq \int_{C \geq A > B} (x_A - x_B) d\lambda + \int_{A > C > B} (x_A - x_B) d\lambda \\
& + \int_{A > B \geq C} (x_A - x_B) d\lambda + \int_{C \geq B > A} (x_B - x_A) d\lambda \\
& + \int_{B > C > A} (x_B - x_A) d\lambda + \int_{B > A \geq C} (x_B - x_A) d\lambda \\
& = D(x_A, x_B)
\end{aligned} \tag{C.11}$$

The equation holds if and only if

$$\int_{C > \{A, B\} \vee C < \{A, B\}} |x_A - x_B| d\lambda = 0. \tag{C.12}$$

in which $\int_{C > \{A, B\} \vee C < \{A, B\}}$ representation is defined similar to Eq. (C.6).

Q.E.D.

Reference

- [1] Bob G Witmer and Michael J Singer. “Measuring presence in virtual environments: A presence questionnaire”. In: *Presence* 7.3 (1998), pp. 225–240.
- [2] Robert B Welch et al. “The effects of pictorial realism, delay of visual feedback, and observer interactivity on the subjective sense of presence”. In: *Presence: Teleoperators & Virtual Environments* 5.3 (1996), pp. 263–273.
- [3] Hongyi Li et al. “Effects of immersion in a simulated natural environment on stress reduction and emotional arousal: A systematic review and meta-analysis”. In: *Frontiers in Psychology* 13 (2023), p. 1058177.
- [4] Ernest Bielinis et al. “Effect of viewing video representation of the urban environment and forest environment on mood and level of procrastination”. In: *International Journal of Environmental Research and Public Health* 17.14 (2020), p. 5109.
- [5] Hyun In Jo, Kounseok Lee, and Jin Yong Jeon. “Effect of noise sensitivity on psychophysiological response through monoscopic 360 video and stereoscopic sound environment experience: a randomized control trial”. In: *Scientific reports* 12.1 (2022), p. 4535.
- [6] Brid Sona, Erik Dietl, and Anna Steidle. “Recovery in sensory-enriched break environments: integrating vision, sound and scent into simulated indoor and outdoor environments”. In: *Ergonomics* 62.4 (2019), pp. 521–536.
- [7] Wenfei Yao, Xiaofeng Zhang, and Qi Gong. “The effect of exposure to the natural environment on stress reduction: A meta-analysis”. In: *Urban Forestry & Urban Greening* 57 (2021), p. 126932.
- [8] Nicola L Yeo et al. “What is the best way of delivering virtual nature for improving mood? An experimental comparison of high definition TV, 360 video, and computer generated virtual reality”. In: *Journal of environmental psychology* 72 (2020), p. 101500.

- [9] Jean-Christophe Servotte et al. “Virtual reality experience: Immersion, sense of presence, and cybersickness”. In: *Clinical Simulation in Nursing* 38 (2020), pp. 35–43.
- [10] Giuseppe Riva et al. “Affective interactions using virtual reality: the link between presence and emotions”. In: *Cyberpsychology & behavior* 10.1 (2007), pp. 45–56.
- [11] Karl Lenz. “Behavior in Public Places. Notes on the Social Organization of Gatherings”. In: *Goffman-Handbuch: Leben–Werk–Wirkung*. Springer, 2022, pp. 291–297.
- [12] Kristine Nowak. “Defining and differentiating copresence, social presence and presence as transportation”. In: *presence 2001 conference, Philadelphia, PA*. Vol. 2. Citeseer. 2001, pp. 686–710.
- [13] Gunilla Borgefors. “Distance transformations in arbitrary dimensions”. In: *Computer vision, graphics, and image processing* (1984), pp. 321–345.
- [14] Rundi Wu et al. “Multimodal shape completion via conditional generative adversarial networks”. In: *ECCV*. 2020, pp. 281–296.
- [15] Steven C Mills and Tillman J Ragan. “A tool for analyzing implementation fidelity of an integrated learning system”. In: *Educational Technology research and development* 48.4 (2000), pp. 21–41.
- [16] Tianyu Luan et al. “Spectrum AUC Difference (SAUCD): Human-aligned 3D Shape Evaluation”. In: *arXiv preprint arXiv:2403.01619* (2024).
- [17] Tianyu Luan et al. “High Fidelity 3D Hand Shape Reconstruction via Scalable Graph Frequency Decomposition”. In: *CVPR*. 2023, pp. 16795–16804.
- [18] Rundi Wu and Changxi Zheng. “Learning to Generate 3D Shapes from a Single Example”. In: *arXiv preprint arXiv:2208.02946* (2022).
- [19] Peizhen Lin et al. “3D mesh reconstruction of indoor scenes from a single image in-the-wild”. In: *International Conference on Graphics and Image Processing*. 2022, pp. 457–465.
- [20] Xingkui Wei et al. “Deep hybrid self-prior for full 3D mesh generation”. In: *ICCV*. 2021, pp. 5805–5814.
- [21] Xinxin Zuo et al. “Unsupervised 3d human mesh recovery from noisy point clouds”. In: *arXiv preprint arXiv:2107.07539* (2021).

- [22] Rongfei Zeng, Mai Su, and Xingwei Wang. “CD2: Fine-grained 3D Mesh Reconstruction with Twice Chamfer Distance”. In: *arXiv preprint arXiv:2206.00447* (2022).
- [23] Rakesh Shrestha et al. “Meshmvs: Multi-view stereo guided mesh reconstruction”. In: *3DV*. IEEE. 2021, pp. 1290–1300.
- [24] Marie-Julie Rakotosaona et al. “Learning delaunay surface elements for mesh reconstruction”. In: *CVPR*. 2021, pp. 22–31.
- [25] Zhihao Zhang, Xinyang Ren, and Xianqiang Yang. “Parametric chamfer alignment based on mesh deformation”. In: *Measurement and Control* (2023), pp. 192–201.
- [26] Audrius Kulikajevs et al. “Auto-Refining 3D Mesh Reconstruction Algorithm From Limited Angle Depth Data”. In: *IEEE Access* (2022), pp. 87083–87098.
- [27] Tao Hu et al. “Self-Supervised 3D Mesh Reconstruction from Single Images”. In: *CVPR*. 2021, pp. 6002–6011.
- [28] Zhiqin Chen et al. “Decor-gan: 3d shape detailization by conditional refinement”. In: *CVPR*. 2021, pp. 15740–15749.
- [29] Yinyu Nie et al. “Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image”. In: *CVPR*. 2020, pp. 55–64.
- [30] Paul Henderson and Vittorio Ferrari. “Learning to generate and reconstruct 3d meshes with only 2d supervision”. In: *arXiv preprint arXiv:1807.09259* (2018).
- [31] Jiaxiang Tang et al. “Point scene understanding via disentangled instance mesh reconstruction”. In: *ECCV*. 2022, pp. 684–701.
- [32] Hari Santhanam, Nehal Doiphode, and Jianbo Shi. “Automated Line Labelling: Dataset for Contour Detection and 3D Reconstruction”. In: *IEEE Winter Conference on Applications of Computer Vision*. 2023, pp. 3136–3145.
- [33] Nanyang Wang et al. “Pixel2mesh: Generating 3d mesh models from single rgb images”. In: *ECCV*. 2018, pp. 52–67.
- [34] Kyle Genova et al. “Local deep implicit functions for 3d shape”. In: *CVPR*. 2020, pp. 4857–4866.
- [35] Jan Bechtold et al. “Fostering generalization in single-view 3d reconstruction by learning a hierarchy of local and global shape priors”. In: *CVPR*. 2021, pp. 15880–15889.

- [36] Maxim Tatarchenko et al. “What do single-view 3d reconstruction networks learn?” In: *CVPR*. 2019, pp. 3405–3414.
- [37] Panos Achlioptas et al. “Learning representations and generative models for 3d point clouds”. In: *ICML*. 2018, pp. 40–49.
- [38] Solomon Kullback and Richard A Leibler. “On information and sufficiency”. In: *The annals of mathematical statistics* (1951), pp. 79–86.
- [39] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. “3d point cloud generative adversarial network based on tree structured graph convolutions”. In: *ICCV*. 2019, pp. 3859–3868.
- [40] Kai Wang et al. “A benchmark for 3D mesh watermarking”. In: *Shape Modeling International Conference*. IEEE. 2010, pp. 231–235.
- [41] Abdullah Bulbul et al. “Assessing visual quality of 3-D polygonal models”. In: *IEEE Signal Processing Magazine* (2011), pp. 80–90.
- [42] Guillaume Lavoué. “A local roughness measure for 3D meshes and its application to visual masking”. In: *ACM Transactions on Applied Perception* (2009), pp. 1–23.
- [43] Massimiliano Corsini et al. “Perceptual metrics for static and dynamic triangle meshes”. In: *Comput. Graph. Forum*. 2013, pp. 101–125.
- [44] Javier Duoandikoetxea and Javier Duoandikoetxea Zuazo. *Fourier analysis*. Vol. 29. American Mathematical Soc., 2001.
- [45] William L Burke, William L Burke, and William L Burke. *Applied differential geometry*. Cambridge University Press, 1985.
- [46] Alexander I Bobenko et al. *Discrete differential geometry*. Vol. 38. Springer, 2008.
- [47] Patrick Pérez, Michel Gangnet, and Andrew Blake. “Poisson image editing”. In: *SIGGRAPH*. 2003, pp. 313–318.
- [48] Jianbing Shen et al. “Gradient based image completion by solving the Poisson equation”. In: *Computers & Graphics* (2007), pp. 119–126.
- [49] Xin Wang. “Laplacian operator-based edge detectors”. In: *IEEE TPAMI* (2007), pp. 886–890.
- [50] Shen-Chuan Tai and Shih-Ming Yang. “A fast method for image noise estimation using Laplacian operator and adaptive edge detection”. In: *International Symposium on Communications, Control and Signal Processing*. 2008, pp. 1077–1081.

- [51] Dilip Krishnan and Rob Fergus. “Fast image deconvolution using hyper-Laplacian priors”. In: *NeurIPS 22* (2009).
- [52] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. “Local laplacian filters: edge-aware image processing with a laplacian pyramid.” In: *ACM TOG* (2011), p. 68.
- [53] Mark Meyer et al. “Discrete differential-geometry operators for triangulated 2-manifolds”. In: *Visualization and mathematics III*. Springer, 2003, pp. 35–57.
- [54] Fan RK Chung. *Spectral graph theory*. Vol. 92. American Mathematical Soc., 1997.
- [55] Karl Pearson. “Notes on the history of correlation”. In: *Biometrika* (1920), pp. 25–45.
- [56] Charles Spearman. “Correlation calculated from faulty data”. In: *British journal of psychology* (1910), p. 271.
- [57] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. “DeepHandMesh: A Weakly-supervised Deep Encoder-Decoder Framework for High-fidelity Hand Mesh Modeling”. In: *ECCV*. 2020.
- [58] Rasmus Jensen et al. “Large scale multi-view stereopsis evaluation”. In: *CVPR*. 2014, pp. 406–413.
- [59] Lizhen Wang et al. “FaceVerse: a Fine-grained and Detail-controllable 3D Face Morphable Model from a Hybrid Dataset”. In: *CVPR*. 2022, pp. 20333–20342.
- [60] *GoboTree - Photos, Cut-outs, 3D people*. <https://www.gobotree.com/>.
- [61] *SketchFab - The best 3D viewer on the web*. <https://sketchfab.com/>.
- [62] Nikolay Ponomarenko et al. “TID2008-a database for evaluation of full-reference visual quality assessment metrics”. In: *Advances of Modern Radioelectronics* (2009), pp. 30–45.
- [63] RECOMMENDATION ITU-R BT. “Methodology for the subjective assessment of the quality of television pictures”. In: *International Telecommunication Union* (2002).
- [64] Frederik Michel Dekking et al. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Vol. 488. Springer, 2005.
- [65] Maurice George Kendall et al. “The advanced theory of statistics.” In: *The advanced theory of statistics* (1946).

- [66] Mathieu Blondel et al. “Fast differentiable sorting and ranking”. In: *ICML*. 2020, pp. 950–959.
- [67] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *NeurIPS* 32 (2019).
- [68] Javier Romero, Dimitrios Tzionas, and Michael J. Black. “Embodied Hands: Modeling and Capturing Hands and Bodies Together”. In: *SIGGRAPH*. 245:1–245:17 36.6 (2017).
- [69] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.
- [70] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. “Convolutional mesh regression for single-image human shape reconstruction”. In: *CVPR*. 2019, pp. 4501–4510.
- [71] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. “Poisson surface reconstruction”. In: *Eurographics Symposium on Geometry Processing*. 2006.
- [72] Robert Bridson. “Fast Poisson disk sampling in arbitrary dimensions.” In: *SIGGRAPH sketches* (2007), p. 1.
- [73] Paolo Cignoni et al. “MeshLab: an Open-Source Mesh Processing Tool”. In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. 2008.
- [74] Gabriel Taubin. “Curve and surface smoothing without shrinkage”. In: *ICCV*. 1995, pp. 852–857.
- [75] Michael Garland and Paul S Heckbert. “Surface simplification using quadric error metrics”. In: *SIGGRAPH*. 1997, pp. 209–216.
- [76] Yana Hasson et al. “Learning joint reconstruction of hands and manipulated objects”. In: *CVPR*. 2019, pp. 11807–11816.
- [77] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. “Weakly-supervised domain adaptation via gan and mesh model for estimating 3d hand poses interacting objects”. In: *CVPR*. 2020, pp. 6121–6131.
- [78] John Yang et al. “Seqhand: Rgb-sequence-based 3d hand pose and shape estimation”. In: *ECCV*. Springer. 2020, pp. 122–139.

- [79] Xingyu Chen et al. “MobRecon: Mobile-Friendly Hand Mesh Reconstruction from Monocular Image”. In: *arXiv preprint arXiv:2112.02753* (2021).
- [80] Lixin Yang et al. “BiHand: Recovering hand mesh with multi-stage bisected hour-glass networks”. In: *arXiv preprint arXiv:2008.05079* (2020).
- [81] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. “Towards Accurate Alignment in Real-time 3D Hand-Mesh Reconstruction”. In: *ICCV*. 2021, pp. 11698–11707.
- [82] Kevin Lin, Lijuan Wang, and Zicheng Liu. “End-to-end human pose and mesh reconstruction with transformers”. In: *CVPR*. 2021, pp. 1954–1963.
- [83] Liangjian Chen et al. “Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos”. In: *IEEE Winter Conference on Applications of Computer Vision*. 2021, pp. 1050–1059.
- [84] Chengde Wan et al. “Dual grid net: Hand mesh vertex regression from single depth maps”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*. Springer. 2020, pp. 442–459.
- [85] Ziwei Yu et al. “Overcoming the trade-off between accuracy and plausibility in 3d hand shape reconstruction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 544–553.
- [86] Zhigang Tu et al. “Consistent 3d hand reconstruction in video via self-supervised learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.8 (2023), pp. 9469–9485.
- [87] Yujin Chen et al. “Joint hand-object 3d reconstruction from a single image with cross-branch feature fusion”. In: *IEEE Transactions on Image Processing* 30 (2021), pp. 4008–4021.
- [88] Yujin Chen et al. “Model-based 3d hand reconstruction via self-supervised learning”. In: *CVPR*. 2021, pp. 10451–10460.
- [89] Lihao Ge et al. “3d hand shape and pose estimation from a single rgb image”. In: *CVPR*. 2019, pp. 10833–10842.
- [90] Pengfei Xie et al. “MS-MANO: Enabling Hand Pose Tracking with Biomechanical Constraints”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 2382–2392.
- [91] Gyeongsik Moon et al. “InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image”. In: *ECCV*. 2020.

- [92] Xiong Zhang et al. “Hand Image Understanding via Deep Multi-Task Learning”. In: *ICCV*. 2021, pp. 11281–11292.
- [93] Sheng Liu et al. “Nech: neural clothed human model”. In: *2021 International Conference on Visual Communications and Image Processing (VCIP)*. IEEE. 2021, pp. 1–5.
- [94] Zhigang Tu et al. “Consistent 3D Hand Reconstruction in Video via self-supervised Learning”. In: *arXiv preprint arXiv:2201.09548* (2022).
- [95] Xiong Zhang et al. “End-to-end hand mesh recovery from a monocular rgb image”. In: *ICCV*. 2019, pp. 2354–2364.
- [96] Xinqian Zheng, Boyi Jiang, and Juyong Zhang. “Deformation representation based convolutional mesh autoencoder for 3D hand generation”. In: *Neurocomputing* 444 (2021), pp. 356–365.
- [97] Hao Peng, Chuhua Xian, and Yunbo Zhang. “3D hand mesh reconstruction from a monocular RGB image”. In: *The Visual Computer* 36.10 (2020), pp. 2227–2239.
- [98] Kunlun Xu, Yuxin Peng, and Jiahuan Zhou. “Uncover the Body: Occluded Person Re-identification via Masked Image Modeling”. In: *International Conference on Image and Graphics*. Springer. 2023, pp. 241–253.
- [99] Tianyu Luan et al. “Pc-hmr: Pose calibration for 3d human mesh recovery from 2d images/videos”. In: *AAAI*. 2021, pp. 2269–2276.
- [100] Tianyu Luan et al. “Divide and Fuse: Body Part Mesh Recovery from Partially Visible Human Images”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 350–367.
- [101] Yuanhao Zhai et al. “Language-guided human motion synthesis with atomic actions”. In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 5262–5271.
- [102] Xuan Gong et al. “Progressive Multi-View Human Mesh Recovery with Self-Supervision”. In: *AAAI*. 2023.
- [103] Xuan Gong et al. “Self-supervised Human Mesh Recovery with Cross-Representation Alignment”. In: *ECCV*. 2022.
- [104] Zhong Li et al. “3d human avatar digitization from a single image”. In: *Proceedings of the 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*. 2019, pp. 1–8.

- [105] Zhong Li et al. “Animated 3D human avatars from a single image with GAN-based texture inference”. In: *Computers & Graphics* 95 (2021), pp. 81–91.
- [106] Zhong Li et al. “Robust 3D human motion reconstruction via dynamic template construction”. In: *2017 International Conference on 3D Vision (3DV)*. IEEE. 2017, pp. 496–505.
- [107] Matthew Loper et al. “SMPL: A Skinned Multi-Person Linear Model”. In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.
- [108] Dominik Kulon et al. “Weakly-supervised mesh-convolutional hand reconstruction in the wild”. In: *CVPR*. 2020, pp. 4990–5000.
- [109] Gyeongsik Moon and Kyoung Mu Lee. “I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image”. In: *ECCV*. Springer. 2020, pp. 752–768.
- [110] Gyeongsik Moon, Takaaki Shiratori, and Kyoung Mu Lee. “DeepHandMesh: A Weakly-supervised Deep Encoder-Decoder Framework for High-fidelity Hand Mesh Modeling”. In: *ECCV*. 2020.
- [111] David I Shuman et al. “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains”. In: *IEEE signal processing magazine* 30.3 (2013), pp. 83–98.
- [112] Joan Bruna et al. “Spectral networks and locally connected networks on graphs”. In: *arXiv preprint arXiv:1312.6203* (2013).
- [113] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).
- [114] Michael Schlichtkrull et al. “Modeling relational data with graph convolutional networks”. In: *European semantic web conference*. Springer. 2018, pp. 593–607.
- [115] Xiaodan Xing et al. “Dynamic spectral graph convolution networks with assistant task training for early mci diagnosis”. In: *MICCAI*. Springer. 2019, pp. 639–646.
- [116] Chao Shang et al. “Multi-view spectral graph convolution with consistent edge attention for molecular modeling”. In: *Neurocomputing* 445 (2021), pp. 12–25.
- [117] Hao Zhu and Piotr Koniusz. “Simple spectral graph convolution”. In: *ICLR*. 2020.
- [118] Yi Ma et al. “Spectral-based graph convolutional network for directed graphs”. In: *arXiv preprint arXiv:1907.08990* (2019).

- [119] Jian Du et al. “Topology adaptive graph convolutional networks”. In: *arXiv preprint arXiv:1710.10370* (2017).
- [120] Zhengyang Wang et al. “Advanced graph and sequence neural networks for molecular property prediction and drug discovery”. In: *Bioinformatics* 38.9 (2022), pp. 2579–2586.
- [121] Shiyang Cheng et al. “Faster, better and more detailed: 3d face reconstruction with graph convolutional networks”. In: *ACCV*. 2020.
- [122] Jiangke Lin et al. “Towards high-fidelity 3d face reconstruction from in-the-wild images using graph convolutional networks”. In: *CVPR*. 2020, pp. 5891–5900.
- [123] Charles Loop. “Smooth subdivision surfaces based on triangles”. In: *Master’s thesis, University of Utah, Department of Mathematics* (1987).
- [124] Jameel Malik et al. “HandVoxNet++: 3D Hand Shape and Pose Estimation using Voxel-Based Neural Networks”. In: *arXiv preprint arXiv:2107.01205* (2021).
- [125] Deying Kong, Haoyu Ma, and Xiaohui Xie. “Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation”. In: *arXiv preprint arXiv:2009.12473* (2020).
- [126] Stefania Sardellitti, Sergio Barbarossa, and Paolo Di Lorenzo. “On the graph Fourier transform for directed graphs”. In: *IEEE Journal of Selected Topics in Signal Processing* 6 (2017), pp. 796–811.
- [127] Yana Hasson et al. “Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction”. In: *CVPR*. 2020, pp. 571–580.
- [128] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. “Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering”. In: *CVPR*. 2019, pp. 1067–1076.
- [129] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. “3d hand shape and pose from images in the wild”. In: *CVPR*. 2019, pp. 10843–10852.
- [130] Christian Zimmermann and Thomas Brox. “Learning to estimate 3d hand pose from single rgb images”. In: *ICCV*. 2017, pp. 4903–4911.
- [131] Hengkai Guo et al. “Towards good practices for deep 3d hand pose estimation”. In: *arXiv preprint arXiv:1707.07248* (2017).
- [132] Linlin Yang et al. “Aligning latent spaces for 3d hand pose estimation”. In: *ICCV*. 2019, pp. 2335–2343.

- [133] Viktor Rudnev et al. “EventHands: Real-Time Neural 3D Hand Pose Estimation from an Event Stream”. In: *ICCV*. 2021, pp. 12385–12395.
- [134] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. “Synthetic training for accurate 3d human pose and shape estimation in the wild”. In: *arXiv preprint arXiv:2009.10013* (2020).
- [135] Georgios Pavlakos et al. “Learning to estimate 3D human pose and shape from a single color image”. In: *CVPR*. 2018, pp. 459–468.
- [136] Jinlong Yang et al. “Estimation of human body shape in motion with wide clothing”. In: *ECCV*. Springer. 2016, pp. 439–454.
- [137] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. “Vibe: Video inference for human body pose and shape estimation”. In: *CVPR*. 2020, pp. 5253–5263.
- [138] Li Jiang et al. “Gal: Geometric adversarial loss for single-view 3d-object reconstruction”. In: *ECCV*. 2018, pp. 802–816.
- [139] Tong Wu et al. “Density-aware chamfer distance as a comprehensive metric for point cloud completion”. In: *arXiv preprint arXiv:2111.12702* (2021).
- [140] Vassilis Athitsos and Stan Sclaroff. “Estimating 3D hand pose from a cluttered image”. In: *CVPR*. Vol. 2. IEEE. 2003, pp. II–432.
- [141] Lars Mescheder et al. “Occupancy networks: Learning 3d reconstruction in function space”. In: *CVPR*. 2019, pp. 4460–4470.
- [142] Yujin Chen et al. “Model-based 3D Hand Reconstruction via Self-Supervised Learning”. In: *CVPR*. 2021.
- [143] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. “Massively Parallel Multi-view Stereopsis by Surface Normal Diffusion”. In: 2015.
- [144] Yizhou Yu et al. “Mesh editing with poisson-based gradient field manipulation”. In: *SIGGRAPH*. 2004, pp. 644–651.
- [145] Zicong Fan et al. “Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation”. In: *International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 1–10.
- [146] Javier Romero, Dimitrios Tzionas, and Michael J. Black. “Embodied Hands: Modeling and Capturing Hands and Bodies Together”. In: *SIGGRAPH* (2017).

- [147] Gyeongsik Moon et al. “Authentic Hand Avatar from a Phone Scan via Universal Hand Model”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 2029–2038.
- [148] Xiao Fu et al. “GeoWizard: Unleashing the Diffusion Priors for 3D Geometry Estimation from a Single Image”. In: *ECCV*. 2024.
- [149] Yvain Quéau et al. “A variational approach to shape-from-shading under natural illumination”. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition: 11th International Conference, EMMCVPR 2017, Venice, Italy, October 30–November 1, 2017, Revised Selected Papers 11*. Springer. 2018, pp. 342–357.
- [150] Christian Zimmermann et al. “Freihand: A dataset for markerless capture of hand pose and shape from single rgb images”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 813–822.
- [151] Georgios Pavlakos et al. “Reconstructing Hands in 3D with Transformers”. In: *CVPR*. 2024.
- [152] Xingyu Chen et al. “HandOS: 3D Hand Reconstruction in One Stage”. In: *arXiv preprint arXiv:2412.01537* (2024).
- [153] Zhishan Zhou et al. “A simple baseline for efficient hand mesh reconstruction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 1367–1376.
- [154] Michael Seeber et al. “Realistichands: A hybrid model for 3d hand reconstruction”. In: *2021 International conference on 3D vision (3DV)*. IEEE. 2021, pp. 22–31.
- [155] Qijun Gan et al. “Fine-grained multi-view hand reconstruction using inverse rendering”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 38. 3. 2024, pp. 1779–1787.
- [156] Mengcheng Li et al. “Interacting attention graph for single image two-hand reconstruction”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 2761–2770.
- [157] Wenjing Pan and Xinrong Chen. “Hand-object interaction reconstruction method based on diffusion model and SDFs”. In: *IET Conference Proceedings CP989*. Vol. 2024. 21. IET. 2024, pp. 466–475.
- [158] Zhenjun Yu et al. “Dynamic Reconstruction of Hand-Object Interaction with Distributed Force-aware Contact Representation”. In: *arXiv preprint arXiv:2411.09572* (2024).

- [159] Mihai Zanfir et al. “Thundr: Transformer-based 3d human reconstruction with markers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12971–12980.
- [160] Lixin Yang et al. “Cpf: Learning a contact potential field to model the hand-object interaction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11097–11106.
- [161] Enric Corona et al. “LISA: Learning Implicit Shape and Appearance of Hands”. In: *CVPR*. 2022.
- [162] Chen Guo et al. “Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 12858–12868.
- [163] Yajing Chen et al. “Self-supervised learning of detailed 3d face reconstruction”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 8696–8705.
- [164] Luo Jiang et al. “3D face reconstruction with geometry details from a single image”. In: *IEEE Transactions on Image Processing* 27.10 (2018), pp. 4756–4770.
- [165] Manyi Li and Hao Zhang. “D2im-net: Learning detail disentangled implicit fields from single images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10246–10255.
- [166] Justus Thies, Michael Zollhöfer, and Matthias Nießner. “Deferred neural rendering: Image synthesis using neural textures”. In: *Acm Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–12.
- [167] Jan Kautz, John Snyder, and Peter-Pike J Sloan. “Fast arbitrary brdf shading for low-frequency lighting using spherical harmonics.” In: *Rendering Techniques* 2.291-296 (2002), p. 1.
- [168] Jean-Denis Durou et al. “A comprehensive introduction to photometric 3d-reconstruction”. In: *Advances in Photometric 3D-Reconstruction* (2020), pp. 1–29.
- [169] Mahmoud Afifi. “11K Hands: Gender recognition and biometric identification using a large dataset of hand images”. In: *Multimedia Tools and Applications* 78 (2019), pp. 20835–20854.
- [170] Camillo Lugaresi et al. “Mediapipe: A framework for perceiving and processing reality”. In: *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*. Vol. 2019. 2019.

- [171] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.
- [172] Zhongpai Gao et al. “Semi-supervised 3d face representation learning from unconstrained photo collections”. In: *CVPRW*. 2020, pp. 348–349.
- [173] Alexander Kirillov et al. “Segment Anything”. In: *arXiv:2304.02643* (2023).
- [174] Stephen Wolfram. *Mathematica: a system for doing mathematics by computer*. Addison Wesley Longman Publishing Co., Inc., 1991.
- [175] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge University Press, 2012.
- [176] Dmitri Burago, Yuri Burago, and Sergei Ivanov. *A course in metric geometry*. Vol. 33. American Mathematical Society, 2022.